CARL VON OSSIETZKY universität OLDENBURG

Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

— Abteilung Systemanalyse und -optimierung —

Abschlussbericht der Projektgruppe

Marine Observation Platform for Surfaces IV

---

vorgelegt von

Michael Beering
Konstantin Gebel
Eike Hagena
Maximilian Hipp
Björn Koopmann
Henning Lawatsch
Zahra Paya
Dennis Pilny
Peter Tank

Gutachter:

Prof. Dr.-Ing. Axel Hahn
M.Sc. Marius Brinkmann

Oldenburg, 28. September 2016

Zusammenfassung

Der vorliegende Bericht fasst die Ergebnisse des Mops iv-Projektvorhabens zusammen. Die an dem System Mops beteiligten Projektgruppen entwickeln ein autonomes Wasserfahrzeug, um dem ICBM ein effizientes System zur Aufnahme von Messdaten in regionalen Binnengewässern bereitzustellen.

Der thematische Schwerpunkt der Projektgruppe Mops iv ist die vollautomatische Erkennung von Kollisionsrisiken sowie die Einleitung geeigneter Gegenmaßnahmen zur Vermeidung von Zusammenstößen. Notwendige Rahmenbedingungen wie Hardwareerweiterungen und Softwareschnittstellen wurden dafür definiert. Der Abschlussbericht beinhaltet weiterhin grundlegende Projektentscheidungen, Schwerpunktdefinitionen, die Beschreibung genutzter Kollaborationstools sowie gruppeninterner Arbeitstechniken.

Die Meilensteine werden innerhalb der fünf Teilbereiche *Hardware*, *Software*, *Kollisionserkennung*, *Kollisionsverhütung* und *Regelung* bearbeitet. Im Anschluss an eine umfassende Anforderungsanalyse werden mit dem systematischen Entwurf der Teilkomponenten die Grundlagen für die Realisierung des Gesamtsystems geschaffen. Schwerpunkt der Implementierung ist neben einer sensorbasierten Kollisionserkennung die simulative Kollisionsverhütung.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einführung

Die Projektgruppe Mops iv ist eine jährlich stattfindende Veranstaltung der Carl von Ossietzky Universität Oldenburg, die von der Abteilung Systemanalyse und -optimierung angeboten wird. Unterstützt wird diese durch das Institut für Chemie und Biologie des Meeres (ICBM) in Oldenburg. Die genaue Bezeichnung des Projektes lautet *Marine Observation Plattform for Surfaces* und beschäftigt sich mit der Weiterentwicklung eines Forschungsbootes, welches autonom Daten über Gewässer sammeln soll.

Die Gruppe befasst sich in vierter Iteration mit der Entwicklung des Bootes und besteht aus neun Teilnehmern der Studiengänge Informatik und Wirtschaftsinformatik. Betreut wird die Gruppe von der Abteilung Systemanalyse und -optimierung mit den Verantwortlichen Prof. Dr.-Ing. Axel Hahn, Sascha Hornauer, Mohamed Abdelaal und Marius Brinkmann.

Viele Gewässer werden aufgrund der Bewirtschaftung oder Verschmutzung durch Menschen geschädigt. Diese Veränderungen an der Umwelt sollen mithilfe der Plattform untersucht werden.

1.1 Aufgabenstellung und Zielsetzung

Primäres Ziel ist es, die Betriebssicherheit des Wasserfahrzeuges zu erhöhen. Hierzu zählen die Überarbeitung der Nutzbarkeit, der Hardwarekomponenten sowie der Betriebsfähigkeit des Wasserfahrzeugs. Unter Betriebsfähigkeit fällt die Systemstabilität bei alltäglicher Nutzung sowie die Widerstandsfähigkeit gegen mögliche Umwelteinflüsse.

Auch die Software zur Kontrolle des Wasserfahrzeugs steht zur Überarbeitung bzgl. Stabilität und Benutzerfreundlichkeit an. Des Weiteren wird der Informationsgehalt des Graphical-User-Interfaces angepasst. Durch die Implementierung einer effizienten Kollisionserkennung sollen Hindernisse und Kollisionsrisiken automatisch erkannt werden. Hierzu werden Sensoren eingesetzt, um Objekte zu identifizieren und zu verfolgen.

Mithilfe der gesammelten Daten können mögliche Kollisionen vorhergesagt und durch eine Kollisionsverhütung Gegenmaßnahmen eingeleitet werden. Diese berechnet einen Ausweichkurs und stellt sicher, dass die Position des Wasserfahrzeuges korrekt ist. Ohne dieses kann nicht sichergestellt werden, dass eine Kollision vermieden wird. Die dazu nötigen Reglerstrukturen und Algorithmen müssen überarbeitet und validiert werden.

1.2 Inhaltlicher Aufbau dieses Dokuments

Der Abschlussbericht gliedert sich in neun Kapitel. Der erste Abschnitt beginnt mit einer Einführung in das Projekt Mops iv und einer Übersicht über die Aufgabenstellung und die Ziele des Projektes.

Das zweite Kapitel beschreibt die Projektorganisation und schildert das Vorgehensmodell, welches zur Umsetzung verwendet wird. Die Meilensteinplanung gibt einen Überblick über den zeitlichen Verlauf des Projektes und die in einem Zeitabschnitt geplanten Arbeitspakete. Neben der Rollenverteilung beinhaltet das Kapitel einen Abschnitt über die Infrastruktur, in der wichtige Kommunikationsmittel und Kollaborationswerkzeuge vorgestellt werden.

Die Anforderungsdefinition wird im dritten Kapitel des Zwischenberichts detailliert beschrieben. Neben der Projektvision und den Stakeholdern werden die Rahmenbedinungen erläutert. Darüber hinaus werden funktionale und nicht-funktionalen Anforderungen für die Teilgebiete *Software*, *Kollisionserkennung*, *Kollisionsverhütung*, *Bootsaufbau und Seetauglichkeit* erhoben.

Das vierte Kapitel enthält einen Überblick über das aktuell eingesetzte System. Der Ist-Zustand der Software wurde durch Experimente und analysieren des Quellcodes ermittelt und dokumentiert. Außerdem wurde der Ist-Zustand des Bootes beschrieben und das Zusammenspiel von Hard- und Software in einem Trockentest untersucht.

Die theoretischen Grundlagen des Regelungssystems werden im fünften Kapitel beschrieben. Es wurden verschiedene Reglertypen betrachtet, um festzustellen, welcher für die PositionsRegelung des Wasserfahrzeugs geeignet ist. Außerdem werden die für die kamerabasierte Kollisionserkennung nötigen Bildverarbeitungsgrundlagen gelegt.

Das sechste Kapitel enthält die funktionale und technische Systemarchitektur und vollzieht einen systematischen Entwurf der Teilkomponenten. Das Konzept der Kollisionserkennung beinhaltet neben einer Abgrenzung der Systemumgebung ein Architekturkonzept, das die Grundlage für die Realisierung bildet. Das Konzept zur Kollisionsverhütung beschränkt sich bisher auf einen Abschnitt zur Reglerstruktur und Reglerauswahl.

Das Kapitel Realisierung umfasst fünf zentrale Themen. Der erste Abschnitt befasst sich mit der Hardware des Gesamtsystems. Das elektrische System wurde vollständig überarbeitet, um die Stabilität der Stromversorgung zu erhöhen. Des Weiteren wurden Komponenten wie der Laserscan-

ner und die Kamera am Wasserfahrzeug montiert. Der zweite Abschnitt beschreibt die Änderungen an Onshore- und Onboard-Software. Außerdem wird ein Viewer für die Auswertung der Laserdaten vorgestellt. Für den Datenaustausch der verschiedenen Softwarekomponenten wird ein Access-Point eingesetzt. Die für die Kommunikation nötige Architektur wird daher ebenfalls in diesem Abschnitt erläutert. Abschnitt drei und vier beschreiben detailliert die Kollisionserkennung- und vermeidung. Hierbei wird zunächst die kamerabasierte und im Anschluss die laserscannerbasierte Kollisionserkennung erklärt. Der letzte Abschnitt befasst sich mit der konzeptionellen Erstellung eines neuen Regelungssystems und den dafür entwickelten Simulationen.

Das achte Kapitel geht auf die Evaluierung des Projektes ein. Die kamera- und laserbasierte Kollisionserkennung werden bzgl. ihrer Qualität vergleichen. Darauffolgend werden die Ergebnisse einer Laufzeitanalyse für zentrale Softwarekomponenten vorgestellt. Während der Durchführung des Projektes konnte einige technische Probleme festgestellt, die ebenfalls in diesem Kapitel beschrieben werden. Die letzten zwei Abschnitte des Kapitel gehen auf die Revision der Anforderungen und den Abgleich des Meilensteinplans ein.

Den Abschluss des Berichts bildet das Kapitel Projektabschluss. Hierin wird ein Fazit für die durchgeführte Projektgruppe gezogen und ein Ausblick auf eine mögliche Fortsetzung des Projekts gegeben. Außerdem enthält das Dokument verschiedene Anhänge, in denen die Seminarausarbeitungen, die Sitzungs-, Interview- und Testprotokolle, ausgewählte Handbücher sowie einige ergänzende Informationen über den Projektverlauf angebracht werden.

Kapitel 2

Projektorganisation

Der erste Schritt zur Weiterentwicklung des Projektes ist es, eine geeignete Projektstruktur festzulegen. Im Folgenden wird das gewählte Vorgehensmodell Scrum beschrieben und wie es in dieser Projektgruppe realisiert wird.

Des Weiteren geht dieses Kapitel auf die Meilensteinplanung ein. Dazu werden die einzelnen Meilensteine, die zugehörigen Zeiträume und die Aufgabenbereiche angeführt. Dieser Meilensteinplan dient dazu einen Überblick über den Projektverlauf zu gewinnen.

Im Übrigen wird die Rollenverteilung im Projekt aufgezeigt. Anhand dieses Abschnittes lässt sich nachvollziehen, welches Gruppenmitglied für ein bestimmtes Aufgabengebiet zuständig ist.

2.1 Vorgehensmodell

Im Rahmen der gemeinsamen Sitzungen wurden verschiedene Vorgehensmodelle zur Projektorganisation vorgestellt. Auch eine Seminararbeit behandelte dieses Thema umfassend. Die Gruppe hat sich für ein generelles Vorgehen nach dem Scrum-Modell entschieden.

Scrum ist ein aus der Softwareentwicklung bekanntes agiles Vorgehensmodell, bei dem iterativ in kurzen Intervallen von zwei bis fünf Wochen ein Teilrelease des Produktes entwickelt wird. Diese Abschnitte nennt man Sprint. Hierbei wird darauf geachtet, dass die Aufgaben für den neuen Zeitraum umfassend geplant und von der Gruppe im Aufwand bewertet werden. Der generelle Ablauf von Scrum ist in vier Abschnitte aufgeteilt:

Sprint Planning Im Sprint Planning wird die nächste Iteration des zu erstellenden Produkts geplant. Die Planung sieht vor, User Stories auszuwählen, die sich entweder bereits im Backlog befinden oder im Rahmen eines Brainstormings oder mit Hilfe anderer Methoden ermittelt werden. Hierbei ist immer die Meilensteinplanung zu bedenken, da sich die User Stories natürlich an der Zielsetzung orientieren sollen. Nach der

Bestimmung der vollständigen User Stories für den kommenden Sprint werden Tasks erstellt, die von den einzelnen Projektmitgliedern zugewiesen werden können. Entweder ist eine Zuweisung an andere Mitglieder oder an sich selbst möglich. Der Arbeitsaufwand für die Tasks wird von den Gruppenmitgliedern während der Sitzung geschätzt. Ein Vorgehensmodell zum Schätzen nennt sich beispielsweise Planning Poker.

Sprint Während des Sprints werden die dem Sprint hinzugefügten Tasks abgearbeitet. Mögliche Anpassungen an User Stories werden während des Sprints im Backlog abgelegt.

Sprint Review Am Ende eines Sprints wird im Sprint Review der beendete Sprint analysiert. Die gewonnenen Erkenntnisse fließen in die Planung des Folgesprints ein. Ergebnis des Reviews ist dabei der Teilrelease des Systems.

Sprint Retrospektive Nach dem Review werden alle positiven und negativen Aspekte aus Sicht der Entwickler aufgenommen. Dies dient der Verbesserung des Entwicklungsklimas und der Organisation der Gruppe.

Erste Tests des gewählten Modells haben ergeben, dass für die weitere Verwendung einige Anpassungen vorgenommen werden. Die Gründe dafür sind der hohe anfängliche Dokumentationsteil und die Tatsache, dass das Projekt einen hohen Hardwareanteil aufweist. Auch die Tatsache, dass es sich um ein studentisches Projekt handelt, erfordert diverse Anpassungen des Scrum-Modells, um den Anforderungen gerecht zu werden.

1. Es werden keine Daily Scrums durchgeführt, da die Projektgruppe nicht in einem typischen Arbeitsumfeld agiert. Aus diesem Grund ist ein regelmäßiges, tägliches Treffen in Zusammenarbeit mit allen Gruppenmitgliedern nicht möglich. Es werden daher wöchentliche Scrum-Meetings abgehalten, sogenannte Weekly Scrums. Im Projektverlauf sind Phasen mit täglichen Treffen durchaus vorgesehen. In diesen Phasen wird die Gruppe sich jedoch auf die Erledigung der Tasks konzentrieren und auch weiterhin nur Weekly Scrums durchführen.

2. Vor dem Beginn der Produktivsprints wird eine umfangreiche, zusätzliche Anforderungserhebung durchgeführt. Die Anforderungserhebung zielt darauf ab, einen generellen Überblick über das zu erstellende Produkt zu erhalten und eine weitläufigere Projektplanung zu ermöglichen, da die gegebenen Umstände der Projektgruppe dies erfordern. Andere Module der Teilnehmer, Klausuren und Arbeit erschweren die Projektplanung teilweise, sodass eine umfangreichere Planung im Vorfeld den erschwerten Umständen entgegenwirken kann.

3. Auf das Schätzen von Tasks wird in der Dokumentationsphase verzichtet. Die Aufgaben sind während der ersten Monate zu sehr abhängig von Einflussgrößen wie die spontane Erweiterung von Teilbereichen. Problematisch hierbei ist die Unterbrechung der Arbeit im Falle einer solchen spontanen Anpassung. Auch wechseln die Aufgabenbereiche häufig untereinander und einige Mitglieder nehmen kontinuierliche Querschnittsfunktionen ein, wie die Korrektur der Texte oder die Anpassungen am LaTeX-Dokument. Außerdem werden die Tasks meilensteinorientiert geplant, anstatt mit User Stories.

4. Vor Ende eines Sprints trifft sich die Projektplanungsgruppe und erstellt anhand der aktuellen Erkenntnisse und den geplanten Meilensteine Tasks für das Backlog. Diese werden in einer Gruppensitzung beim Sprint Planning vorgestellt, angepasst und für den kommenden Sprint ausgewählt.

5. Die Dauer eines Sprints beträgt drei Wochen.

2.2 Meilensteinplanung

Die vorläufige Planung der Produktivphase hat eine Unterteilung in zwölf Sprints von jeweils dreiwöchiger Dauer ergeben. Der erste Sprint beginnt am 26. Januar 2016 und der letzte Sprint endet am 30. September 2016. Eine graphische Darstellung des geplanten Verlaufs ist in Abb. D.1 zu finden.

Die Meilensteinplanung sieht vor, alle drei Sprints einen Meilenstein zu erreichen. Die Produktivphase wird von fünf Hauptgruppen bearbeitet:

G1 Software

G2 Hardware

G3 Kollisionserkennung

G4 Regelung

G5 Kollisionsverhütung

2.2.1 Meilenstein I

Bei der Erreichung des ersten Meilensteins werden die Gruppen G1 bis G4 involviert sein. Die Gruppe *G5 - Kollisionsverhütung* wird erst im weiteren Projektverlauf zusammengestellt, da zur Bearbeitung zunächst die Gruppen G1-G4 Vorarbeiten leisten müssen. Im folgenden Abschnitt werden die verschiedenen Gruppenaufgaben erläutert und zusammengefasst.

Die Gruppe *G1 - Software I* befasst sich im ersten Meilenstein mit der Diagnose hinsichtlich der Wartbarkeit der Software. Hierfür wird der Ist-Zustand

analysiert und dokumentiert. Eine erste Analyse hat zudem ergeben, dass das genutzte Build-System Ant auf Maven umgestellt werden soll. Die vorhandenen Kartendaten sollen umfassend ausgewertet werden, damit eine automatische Gefahrenzonen-Detektion möglich ist. Auch Umweltparameter wie zum Beispiel Windrichtung, Windstärke oder Strömung soll simuliert werden können.

Software I
(1) Validierung des Ist-Zustands der Software - Trockenaufbau - Auslesen und Dokumentieren - Überprüfen verschiedener Testfälle - Software-Debugging
(2) Umstellung des Build-Systems (Ant auf Maven)
(3) Vorbereiten der Versionierung
(4) Parsen von Kartendaten zur Gefahrenzonen-Detektion
(5) Parametrisierung der Hardware-Simulation (Wind, Strömung)

Die *G2 - Hardware* Gruppe führt erste Trockentests durch und führt einen Abgleich mit der vorliegenden Dokumentation der Vorgängergruppe durch, um Schwachstellen und Ansatzpunkte für Verbesserungen zu evaluieren. Notwendige bauliche Maßnahmen zur Absenkung eines Sensorkorbs, wie beispielsweise ein Moonpool in der Mitte des Bootes, sollen evaluiert werden.

Hardware I
(1) Validierung des Ist-Zustands der Hardware - Trockenaufbau - Messen und Dokumentieren - Überprüfen verschiedener Testfälle
(2) Überarbeiten und Vervollständigen der existierenden Handbücher
(3) Erarbeiten eines Konzepts zur Sensoranbringung (Moonpool o.ä.)

Der erste Meilenstein sieht zudem vor, erste Ergebnisse für die *Kollisionserkennung (G3)* zu erlangen. Diese sollen beinhalten, welche Hardware angeschafft und wie die Erkennung realisiert werden soll. Es soll betrachtet werden, welche Schnittstellen und Rahmenbedingungen für die weitere Realisierung definiert werden müssen. Weitere zentrale Designentscheidungen, wie

zum Beispiel ein Konzept zur Anbringung der Sensoren oder erste Vorüberlegungen zu technischen Architekturen, sollen ebenfalls behandelt werden.

Kollisionserkennung I
(1) Abgrenzung von der Systemumgebung
(2) Herleiten der funktionalen Architektur
(3) Definition von Schnittstellen
(4) Festlegung von Rahmenbedingungen
(5) Treffen zentraler Designentscheidungen
- Sensoren/Kameras (inkl. Auflösung, Schnittstellen, Preis)
- Konzept zur Anbringung der Sensoren
- Formulieren von Safety Goals und Sicherheitsanforderungen
- Vorüberlegungen zur technischen Architektur
- Ausblick auf weitere Realisierung

Die Gruppe *G4 - Regelung* evaluiert den Reglertyp und arbeitet ein geeignetes mathematisches Modell zur Steuerung des Systems aus. Dieses soll bereits in Ansätzen umgesetzt werden.

Regelung I
(1) Evaluierung des Reglertyps
(2) Definition eines geeigneten mathematischen Modells
(3) Aufbau eines ersten Regelkreises (ohne Übertragungsfunktionen)

2.2.2 Meilenstein II

Auf Basis der Ergebnisse des ersten Meilensteins konzentrieren die Gruppen sich für die Phase des zweiten Meilensteins auf die erste prototypische Realisierung am System.

Der Bereich *Software(G1)* ist ab dem zweiten Meilenstein als eine bereichsübergreifende Querschnittsfunktion zu betrachten. Ihre Aufgabe wird es sein, das mathematische Modell in die Onshore- und Onboard-Software zu implementieren. Darüber hinaus unterstützt der Bereich die Tätigkeitsfelder der *Kollisionserkennung (G3)*. Der Bereich *Kollisionserkennung (G3)* wird während der Erreichung von Meilenstein II stärker mit dem Software-Bereich verknüpft werden. In enger Zusammenarbeit werden die getroffenen Designentscheidungen hinsichtlich des Aufbaus des Kollisionserkennungssystems

implementiert und die entsprechend notwendigen Schnittstellen geschaffen. *Software (G1)* wird weiterhin die GUI pflegen und Anpassungen vornehmen, um weitere Statusinformationen über die GUI darstellen zu können (Beispielsweise Regler-, und Umweltinformationen). Auch werden Kommunikationskonzepte bedacht und eventuell implementiert, die eine größere Distanz des Bootes zum Anwender ermöglicht.

Software II
(1) Implementierung des mathematischen Modells in Onshore- und Onboard-Software
(2) Überarbeitung des Kommunikationskonzeptes
(3) Erweiterung von Kommunikationsmöglichkeiten von Onshore- und Onboard-Software
(4) Realisierung des Sicherheitssystems (Heartbeat-Monitoring)

Kollisionserkennung II
(1) Funktionale Architektur implementieren
(2) Implementierung von Schnittstellen für Kommunikation im System
(3) Evaluation von Hardwarekomponenten
(4) Ergebnisse von Tests analysieren
(5) Vorbereiten der Anbindung der Kollisionsverhütung

Weiterhin ist vorgesehen, das Boot auf die Sensoranbringung vorzubereiten (*G2 - Hardware*) und gegen eindringendes Wasser zu schützen. Ein Konzept dafür liegt nach dem Erreichen von Meilenstein I vor. Auch soll ein Sicherheitssystem implementiert werden. Dieses überwacht die Komponenten und stellt eine unabhängige Komponente im System dar, die von außen jederzeit überprüft werden kann.

Hardware II
(1) Vorbereiten und Realisieren der Sensoranbringung
(2) Umsetzung des Konzepts zur Wasserdichtigkeit
(3) Vorbereiten der Hardware-Anpassungen für Kollisionserkennung
(4) Integrieren des Sicherheitssystems (Heartbeat-Monitoring)

Für die Erreichung des zweiten Meilensteins sollen bereits erste Ansätze für die Kollisionsverhütung entstehen, mit denen in Meilenstein III gearbeitet werden kann. Der Bereich *Regelung (G4)* wird abgeschlossen und in den Bereich *Kollisionsverhütung (G5)* migriert.

Kollisionsverhütung und Regelung I
(1) Zusammenführen von Regelung und Kollisionsverhütung (2) Optimieren der Algorithmen zur Kollisionsverhütung

2.2.3 Meilenstein III

Der dritte Meilenstein fokussiert die Kollisionsverhütung. Basierend auf den Ergebnissen von Meilenstein II sollen die Ansätze weiter verfolgt werden und konkrete Möglichkeiten getestet werden. Hier arbeiten die Bereiche *Hardware (G2)*, *Software (G1)*, *Kollisionsverhütung (G5)* und *Kollisionserkennung (G3)* eng miteinander zusammen.

Software III
(1) Anbindung der neuen Hardware an die Software (2) Erweitern relevanter Schnittstellen für Hardware/Software Integration (3) Vornehmen von GUI-Anpassungen und -Erweiterungen (4) Zusammenarbeit mit anderen Bereichen als Bereich mit Querschnittsfunktion

Für die Kollisionsverhütung werden zunächst im *Hardware (G2)* Bereich ausgearbeitete Design-Entscheidungen praktisch umgesetzt. Falls weitere Hardware benötigt wird, wird diese angeschafft und in das Gesamtsystem integriert.

Hardware III
(1) Anschaffen von benötigter Hardware (2) Integration der neuen Hardware in das System

Die *Kollisionserkennung (G3)* adaptiert das Erkennungssystem gemäß der Architektur in das Gesamtsystem und erweitert die implementieren Schnittstellen zur Kommunikation.

Kollisionserkennung III
(1) Erkennungssystem umfassend in Gesamtsystem integrieren
(2) Fortlaufende Systemprüfungen vornehmen
(3) Schnittstellen anpassen und erweitern

Die Kollisionsverhütung beinhaltet weiterhin die Komponenten der *Regelung (G4)*, um das Schiffsmodell entsprechend steuern zu können, sofern Maßnahmen getroffen werden müssen. Diese beiden Bereiche wurden in Meilenstein II vereint und bilden auch weiterhin eine Arbeit.

Kollisionsverhütung und Regelung II
(1) Ansätze der Kollisionsverhütung erweitern
(2) Kollisionsverhütung implementieren
(3) Kollisionsverhütung mit Regelung abstimmen

2.2.4 Meilenstein IV

In Meilenstein IV sollen umfangreiche Systemtests, der Projektabschluss und das Anfertigen der Dokumentation erfolgen. Begleitend zu den Phasen des Projektes wird eine umfangreiche Dokumentation angefertigt. Da die zur Verfügung stehenden Ressourcen während der vorherigen Phasen auf der Erreichung der Projektziele und Erfüllung der Anforderungen liegt, wird eine umfassende Vervollständigung der Dokumentation erforderlich sein, damit nachfolgende Gruppen ein solides Nachschlagewerk erhalten, in dem alle wichtigen Informationen enthalten sind.

Systemtests sind nötig, um Schwachstellen zu identifizieren und für die nachfolgende Projektgruppe zu dokumentieren. Diese bilden erste Ansätze für weitere Arbeiten und können gegebenenfalls sogar noch während des Projekts behoben werden.

2.3 Rollenverteilung

Um für bestimmte Themen einen über das gesamte Projekt gleichbleibenden Ansprechpartner zu haben, hat es sich als zweckmäßig erwiesen, Rollen einzuführen. Wird eine Rolle von einem oder mehreren Verantwortlichen übernommen, so ist derjenige erster Ansprechpartner über das entsprechende Themengebiet. Die Rollenverteilung hat innerhalb der Gruppe den Vorteil, dass immer ein Ansprechpartner zur Verfügung steht. Außerdem können sich die Betreuer bei Fragen an die entsprechenden Personen wenden.

Administrator

Die Aufgabe des Administrators besteht primär in der Wartung des Servers der Projektgruppe. Er sorgt dafür, dass Services wie das Git, Etherpad und ein FTP-Server zur Verfügung stehen. Kommt es zu Störungen, ist er für das Beheben des Problems zuständig (Verantwortlich: *Maximilian Hipp*).

Projektleiter

Der Projektleiter trägt die Verantwortung für das gesamte Projekt. Seine Aufgabe besteht darin, einen Überblick über den aktuellen Verlauf des Projektes zu haben und das Projekt zur Erbringung der geplanten Ziele zu führen. Bei Problemen tritt er als Vermittler auf und ist für die Motivation des Teams zuständig (Verantwortlich: *Dennis Pilny*).

Wiki-Beauftragter

Das Wiki wird genutzt, um verschiedene Informationen für alle Gruppenmitglieder über eine Webseite zur Verfügung zu stellen. Die Wartung des genutzten Doku-Wikis fällt in den Zuständigkeitsbereich der Wiki-Beauftragten (Verantwortlich: *Zhara Paya*).

Doku-Beauftragter

Die Dokumentation ist ein zentraler Baustein des Projekts und Grundlage für die spätere Bewertung. Die Pflege und das Vorantreiben der Dokumentation ist Aufgabe des Doku-Beauftragten. Außerdem ist er für das Einhalten der LaTeX-Konventionen verantwortlich (Verantwortlich: *Björn Koopmann*).

Git-Beauftragter

Der Git-Beauftragte ist Ansprechpartner bei Problemen, die das Git betreffen. Er steht bei Fragen zu kryptischen Fehlermeldungen zur Verfügung und sorgt für einen reibungslosen Betrieb des Versionierungssystems (Verantwortlich: *Björn Koopmann*).

Jira-Beauftragter

Der Jira-Beauftragte verwaltet das durch das OFFIS zur Verfügung gestellte Jira-System. Er ist an der Sprint-Planung beteiligt und nur ihm ist es erlaubt, einen neuen Sprint zu starten oder zu beenden (Verantwortlich: *Konstantin Gebel*).

Webauftritt-Beauftragter

Der Webauftritt präsentiert die Projektgruppe nach Außen. Die dazugehörige Homepage wird von dem Verantwortlichen erstellt und gepflegt. Außerdem wird diese regelmäßig mit aktuellen Informationen versehen (Verantwortlich: *Konstantin Gebel*).

ICBM-Beauftragter

Der ICBM-Beauftragte ist die zentrale Kontaktperson zu den Mitarbeitern des Instituts. Er wendet sich bei Fragen an entsprechende Personen und ist an dem projektbegleitenden Interviews mit Prof. Dr. Zielinski beteiligt (Verantwortlich: *Peter Tank*).

Finanzbeauftragter

Für die Anschaffung verschiedener Bauteile stehen begrenzte finanzielle Ressourcen zur Verfügung. Die Verwaltung des entsprechenden Budgets fällt in den Aufgabenbereich der Finanzbeauftragten (Verantwortlich: *Hennig Lawatsch, Eike Hagena*).

Inventur-Beauftragter

Die Inventur dient der Feststellung, welches Material und welche Bauteile im Gruppenraum verfügbar sind. Das Führen der Inventurliste ist Aufgabe des Inventurbeauftragten (Verantwortlich: *Peter Tank*).

Raum-Beauftragter

Der Raum-Beauftragte verwaltet die Schränke und das Material, das im Gruppenraum gelagert wird. Bei Fragen, die den Raum betreffen, wie beispielsweise der Organisation mit anderen Gruppen, die den Gruppenraum benutzen, oder das Einrichten eines Gruppen-WLANs, ist er zuständig (Verantwortlich: *Michael Beering*).

2.4 Infrastruktur

Zu der Projektorganisation gehören effiziente Kommunikationstechniken und Kollaborationstools, um die Zusammenarbeit im Team praktikabel realisieren

zu können. Dazu zählt eine umfassende Versionsverwaltung, eine Applikation für das temporäre festhalten von Gedanken und Notizen, ein Projektmanagement Tool und ein Online-Nachschlagewerk. Der nachfolgende Teil stellt die eingesetzten Applikationen und Tools vor, die während der Projektarbeit von den Mitgliedern eingeführt und genutzt wurden.

Git

Für die Auswahl einer Software zur Versionsverwaltung wurden innerhalb der Gruppe Git und SVN vorgeschlagen. Die anschließende Analyse und Abstimmung fiel zugunsten von Git aus, da bereits ein Großteil der Gruppenmitglieder Erfahrungen mit dieser Software aufweisen konnte. Auch die Verfügbarkeit gängiger Git-Anwendungen sowie die umfangreicheren Merge-Möglichkeiten waren Argumente für die Auswahl von Git.

Etherpad

Mithilfe von einer eigenen Etherpad-Instanz kann das Team gleichzeitig und in Echtzeit an einem gemeinsamen Dokument arbeiten. Hierfür ist es lediglich nötig, den Link zu dem Dokument zu kennen. Die Einsatzgebiete von Etherpad sind beispielsweise das Verfassen von Sitzungsprotokollen, Brainstorming oder die Arbeit in Kleingruppen.

Jira

Jira ist eine webbasierte Projektmanagementplattform, die die Zusammenarbeit innerhalb eines Teams ermöglicht. Hierzu zählt die strukturierte Aufteilung von Aufgaben, die Möglichkeit Fehler zu kennzeichnen und mittels Fehlerbehebungsprozessen auszubessern, sowie die übersichtliche Darstellung des aktuellen Projektstatus.

Zur Bearbeitung der Aufgaben werden sogenannte Issues erstellt. Diese betiteln eine Aufgabe oder eine Teilaufgabe einer zusammengefassten Aufgabe. Teammitglieder können sich selbst oder gegenseitig Aufgaben zuweisen und diese bearbeiten. Auch eine umfangreiche Zeiterfassung für die Bearbeitung der Aufgaben ist integriert.

In der Gruppe wird zudem ein Agile-Board genutzt, welches die einfache Modellierung des genutzten Scrum-Prozesses ermöglicht. Sprints können detailliert abgebildet werden und die Aufgaben zwischen verschiedenen Phasen verschoben werden. Diese können auch angepasst werden. So konnte beispielsweise die „Review“-Phase dargestellt werden, in der erledigte Aufgaben von einem anderen Teammitglied geprüft werden, bevor sie als erledigt gekennzeichnet werden können.

DokuWiki

DokuWiki bezeichnet eine freie, auf PHP basierende Wiki-Software. Zur Nutzung der Software wird keine eigene Datenbank vorausgesetzt. Die Speicherung der Informationen erfolgt mittels einfacher Textdateien. Der Funktionsumfang ist in der Basis-Version ausreichend und kann mittels verschiedener Plugins erweitert werden.

Vorteilhaft an DokuWiki ist primär die geringe Komplexität sowie die hohe Robustheit. Versionsverwaltung, Zugriffskontrolle mittels individuellen Accounts und Zugriffsrechten, Volltextsuche und ein umfangreicher Editor zählen zu den Funktionen der Software.

Slack

Slack ist der Name einer Software, die für die Kommunikation und Zusammenarbeit zwischen Mitglieder genutzt wird. Für die Nutzung der Software kann zwischen einem webbasierten Client oder nativen Anwendungen für alle gängigen Betriebssysteme gewählt werden.

Zu Beginn kann man einem Team beitreten. In diesem organisieren sich die Mitglieder untereinander selbst. Die zur Verfügung stehenden Organisationsstrukturen von Slack ermöglichen die Erstellung von Kanälen, öffentlicher oder privater Natur, als auch den Austausch zwischen lediglich zwei Mitgliedern untereinander.

Der Vorteil dieser Plattform ist, dass für Kleingruppen eigene Gruppen erstellt werden können oder die Informationen in den jeweiligen Kanälen der Priorität nach verteilt werden. Wichtige Informationen werden beispielsweise in einem Haupt-Kanal verbreitet, wohingegen eher Unwichtiges in einem separaten Kanal ausgetauscht wird. Weiterhin werden sogenannte Funktionsintegrationen angeboten. Mit diesen kann beispielsweise die Integration von Jira oder Jenkins in Slack realisiert werden. Slack wird in der kostenlosen Version genutzt.

Kapitel 3

Anforderungsdefinition

Während der Anforderungsanalyse in den ersten Wochen der Bearbeitungszeit konnte der Funktionsumfang des zu realisierenden Produkts durch die Formulierung einer Projektvision, Rahmenbedingungen sowie funktionalen und nicht-funktionalen Anforderungen definiert werden. Das nachfolgende Kapitel präsentiert die Ergebnisse dieser Arbeitsphase und liefert wichtige Erkenntnisse für die Konzeption des Systems.

3.1 Projektvision

Das Projekt Mops iv ist ein von neun Studenten der Universität Oldenburg durchgeführtes Projekt, welches über einen Zeitraum von einem Jahr bearbeitet wird. Das Akronym Mops steht für *Marine Observation Platform for Surfaces* und bezeichnet eine Projektidee, die inzwischen in der vierten Generation fortgeführt wird.

In den vorherigen Projektgruppen Mops i-iii wurde ein Wasserfahrzeug entwickelt, welches selbstständig in einer vordefinierten, gesicherten Testumgebung einen Kurs abfahren kann. Letztlich ist es das Ziel des Projektes, mithilfe des Fahrzeugs vereinfachte Seetests und Messungen durchführen zu können. Das Projekt strebt außerdem an, menschliche Kapazitäten zu entlasten und somit eine Kostenersparnis zu erwirken.

Die Tests mit dem Boot werden auf dem Tweelbäker See, mit der Unterstützung des Oldenburger Yachtclubs, durchgeführt. Die Betreuung findet durch die Abteilung Systemanalyse und -optimierung unter der Leitung von Prof. Axel Hahn und Sascha Hornauer an der Universität Oldenburg statt. Zusätzlich erhält das Projekt Unterstützung durch das ICBM mit Prof. Dr. Oliver Zielinksi als Ansprechpartner. Die Lernziele des Projekts sind Projektmanagement und -planung, sowie die Teamarbeit bei dem Bewältigen von komplexen Problemen und der ingenieurmäßigen Entwicklung von Lösungen.

3.1.1 Motivation

In Flüssen, Seen oder Hafenbecken werden immer wieder Vertiefungsarbeiten durchgeführt, um Streckenabschnitte für Schiffe befahrbar zu machen. Um eine hinreichend genaue Einschätzung der biologischen und chemischen Vorgänge in dem zu untersuchenden Gewässer zu gewährleisten, ist es notwendig, ausgewählte Umweltparameter an verschiedenen Messpunkten zu erfassen und die erhaltenen Daten in die nachgeordneten Analysen und Bewertungsprozesse einzubeziehen.

Aufgrund der großen Ausdehnung der betrachteten Gewässer und der zunehmenden Intensivierung der Schifffahrt, wird durch diese Aufgabe in Zukunft eine große Menge an Personal gebunden. Um den Aufwand zu minimieren, gibt es Überlegungen, diese Messungen automatisiert und autonom durchzuführen. Das Projekt MOPS hat daher das Ziel, eine autonome Plattform zu entwickeln, mit der automatisiert Messungen in Gewässern durchgeführt werden können. So können abgelegene oder gesperrte Gebiete erforscht werden, die für den Mensch möglicherweise nur schwer erreichbar sind.

Neben der Überwachung von Wasserqualitätskriterien bei Baggerarbeiten, eignet sich die Plattform zudem auch zur Aufzeichnung von Messwerten über längere Zeiträume und lässt somit Aussagen über die langfristige Entwicklung eines aquatischen Ökosystems zu. Hier könnten dann zum Beispiel neben der Wasserqualität auch die dort lebenden Tiere kontrolliert und vor eventuellen Schadstoffen geschützt werden.

3.1.2 Problembeschreibung

Um die vorher genannten Kriterien zu erfüllen, ist es nötig ein autonomes Boot zu entwickeln, welches unseren Anforderungen entspricht. Hierfür müssen gewisse Voraussetzungen erfüllt werden. So müssen Hardware, Software und der physische Aufbau auf die Gegebenheiten der Umwelt angepasst werden. Dabei spielt insbesondere das Wetter eine entscheidende Rolle. Des Weiteren entstehen Gefahren im Schiffsverkehr. Diesen muss durch die Erkennung und Vermeidung von Kollisionen vorgebeugt werden. Als Ergebnis soll ein autonomes System entstehen, welches dynamisch auf die äußeren Einflüsse reagiert und dessen verschiedene Komponenten zuverlässig kollaborieren müssen.

Bestandteile des bestehenden Systems

Die bisher bestehenden Systemkomponenten werden hier in zwei Gruppen kategorisiert: Hardware und Software. Tabelle 3.1 bietet einen Überblick darüber. Seitens der Software gibt es die Komponenten für die Kontrolle an Land und für die Kommunikation. Für die Hardware gibt es Komponenten die zum Betrieb des Wasserfahrzeugs nötig sind, also beispielsweise die Stromversorgung, den Antrieb und die Regelung. Des Weiteren gibt es die Sensorik für

die Navigation und zur Aufnahme der Umweltdaten. Abschließend sind noch die Kommunikationskomponenten vorhanden.

Hardware	*Software*
Antrieb	Onboard
Batterie / Energieversorgung	Onshore
Steuerung / Regelung	Kommunikation
Sensoren Navigation	
Sensorkorb für Umweltsensoren	
Kommunikation	

Tabelle 3.1: Übersicht über die eingesetzten Komponenten

Mithilfe der Onshore-Software können Wegpunkte und Sperrzonen definiert werden, die das Boot abfährt beziehungsweise meidet. Hierfür ist es allerdings notwendig, dass eventuelle statische Hindernisse oder Gefahrengebiete vorher festgelegt worden sind.

Für die Fahrt setzt die Plattform ein GPS- und Kompass-System ein. Sollte es zu Fehlern während eines Einsatzes kommen, kann das Boot weiterhin über eine Funkfernbedienung gesteuert werden. Die Steuerung des Bootes kann also jederzeit manuell übernommen oder die Mission einfach abgebrochen werden.

Nachteile der aktuellen Lösung

Im Folgenden werden die Nachteile des Gesamtsystems beschrieben.

Benutzerfreundlichkeit Die Handhabung der Stromversorgung ist wenig benutzerfreundlich und fehleranfällig. In der vorherigen Projektphase kam es durch Verpolungen der Batterien wiederholt zu Hardwareausfällen und einem erhöhten Wartungsbedarf.

Die manuelle Kalibrierung des internen Kompasses ist aufwändig und erfordert mindestens zwei Personen. Um die bisherige Plattform in Betrieb zu nehmen, müssen die Sensoren vor dem Beginn der Mission kalibriert werden.

Der Aufbau des Systems ist zeitintensiv und bindet eine zu große Menge an Personal. Zusätzlich ist der Betrieb der Software zeitintensiv, da die Definition der Sperrzonen eine genaue Auskundschaftung des Einsatzgebietes erfordert.

Kollisionserkennung und -vermeidung Trotz Konfiguration reagiert die Plattform nur auf statische Hindernisse, die im Vorfeld definiert werden

müssen. Eine dynamische Erkennung und das Ausweichen von Hindernissen ist nicht möglich.

Regelung Laut des Abschlussberichts der vorherigen Gruppe gibt es keinen vollständigen, das heißt für alle Wetterlagen stabilen, Parametersatz. Hierdurch kann nicht für jede Situation eine korrekte Positionsregelung garantiert werden.

Seetauglichkeit Das Boot ist nicht gegen raue Wetterbedingungen und eintretendes Wasser geschützt.

Sensormessung Bislang ist die Messung von Sensordaten in einem spezifischen Messbereich nicht implementiert. Es wurden zwar bereits testweise die Sensoren C4E-Conductivity zur Prüfung des Salzgehaltes und Nephelometric Turbidity zur Messung der Trübheit des Wasser eingebaut, diese wurden allerdings nicht ins System integriert. Die Daten können demnach noch nicht ausgewertet werden.

Bezüglich des Sensorkorbs stellt sich die Problematik, dass dieser nicht zu Wasser gelassen werden kann, da hierfür eine geeignete Vorrichtung fehlt. Das Ausbringen der Sensoren ist damit zurzeit unmöglich.

Stealth-Problematik Das Boot läuft Gefahr, durch seine geringe Größe und fehlende Informationssysteme oder Signalanlagen, von anderen Verkehrsteilnehmern auf dem Wasser übersehen zu werden.

3.1.3 Lösungsidee

Wir nehmen uns zur Aufgabe, bei der Weiterentwicklung des bestehenden Systems die vorher genannten Nachteile möglichst aufzuheben. Dabei werden zunächst die erwarteten Vorteile genannt und anschließend eine Priorisierung dieser vorgenommen. Zudem wird betrachtet, worauf bei der Realisierung geachtet werden muss.

Erwartete Vorteile

Sollten die vorher genannten Probleme von uns beseitigt werden, sind für das Forschungsboot folgende Vorteile zu erwarten.

Benutzerfreundlichkeit Die Verbesserung der Benutzerfreundlichkeit ist auf zwei Ebenen durchzuführen. Zum einem die hardwareseitige, die den Aufbau und die Inbetriebnahme des Bootes regelt. Dazu könnte der Einbau eines Verpolungsschutzes gehören und eine vereinfachte Kalibrierung des Bootes.

Weiterhin soll die Benutzerfreundlichkeit der Onshore-Software erhöht werden. Dieser fehlt bisher noch eine Schnittstelle zur Auswertung der Sensordaten und die Darstellung von dynamischen Hindernissen.

Kollisionserkennung und -vermeidung Für den Einsatz auf neuen Gewässern ist vorgesehen, Kollisionen mit anderen Schiffen und Schwimmern durch eine zuverlässige Kollisionserkennung und -vermeidung zu verhindern. Durch den Einbezug von Sensoren zur Detektion von Kollisionsrisiken, können statische und dynamische Hindernisse zur Laufzeit selbstständig erkannt und die Aufprallwahrscheinlichkeit durch das Einleiten geeigneter Gegenmaßnahmen reduziert werden.

Die durch unsere Projektgruppe erweiterte Plattform wird somit über eine effektive Kollisionserkennung im Nah- und Fernbereich verfügen. Somit kann ein notwendiger Schutz für die Gesundheit und das Leben aller beteiligten Menschen sowie der Unversehrtheit der Umwelt sichergestellt werden.

Regelung Die Verbesserung der Positionsregelung garantiert eine feingranularere Situationsanalyse, erhöht die Genauigkeit der Navigation und steigert das Vertrauen in die ermittelten Sensorwerte.

Seetauglichkeit Durch eine Optimierung des Antriebskonzepts und dem Erhöhen der Leistung der beiden Motoren wird die Seetauglichkeit des Bootes erhöht. Weiterhin kann durch eine korrekte Einstellung der verwendeten Regler die Genauigkeit der Bahnregelung erhöht werden. Ein an das Einsatzgebiet angepasster Schiffsaufbau schützt die verwendete Elektronik vor Spritzwasser und gewährleistet so einen wirksamen Schutz vor Niederschlägen und Wellenschlag.

Um es mit rasch wechselnden Wetterbedingungen aufnehmen zu können, kann eine Modifikation des bestehenden Bootes hinsichtlich des Schutzes vor eindringendem Wasser eine Lösung sein.

Sensormessung Die durch die Gruppe erarbeiteten Lösungsansätze sehen die Einbindung von weiteren Umweltsensoren vor. Hierzu zählt die genaue Anforderungserhebung, sowie welche Daten zu welchem Zeitpunkt an welcher Stelle und in welcher Form erhoben werden sollen.

Stealth-Problematik Damit andere Verkehrsteilnehmer das Boot sehen, ist das Anbringen einer geeigneten Signalanlage erforderlich.

Bewertungskriterien

Die Qualität der erarbeiteten Lösungen soll über die folgenden Bewertungskriterien überprüfbar gemacht werden.

Benutzerfreundlichkeit Das geplante System muss möglichst einfach kalibriert und in einen betriebsbereiten Zustand versetzt werden können. Sämtliche Arbeitsschritte zum Aufbau des Systems sollten in einem übersichtlichen und leicht verständlichen Benutzerhandbuch aufgeführt sein.

Beschaffenheit der Daten Die Menge an unterschiedlichen Parametern, die mithilfe der Sensoren erfasst und durch die Software ausgewertet werden können, muss ausreichend sein, um eine hinreichend genaue Bewertung der Umwelteinflüsse vornehmen zu können.

Erweiterbarkeit Die eingesetzte Sensorik muss leicht erweiterbar sein. Hierzu zählen sowohl weitere Sensoren als auch Systeme zur Kollisionsvermeidung oder Objekterkennung. Die Einbindung von neuen Systemen muss über bestehende Strukturen möglich sein.

Flexibilität Das System muss sich leicht auf andere Bootsformen und Einsatzfahrzeuge übertragen lassen.

Robustheit Die Bestandteile des Systems, insbesondere der Rumpf, die Antriebseinheit und die Vorrichtung zum Ausbringen der Sensoren, sollten besonders gut an die Umgebung angepasst sein, in der sie zum Einsatz kommen. Außerdem sollten sie möglichst robust gegen äußere Einflüsse sein, um die Häufigkeit von Wartungsarbeiten zu minimieren.

Auch wenn es die Möglichkeit gibt, die Fernsteuerung des Bootes zu verwenden, sollte diese nur im Notfall benutzt werden. Das System muss insgesamt stabil laufen und gut funktionieren. Ebenfalls sollten die physischen Parameter einen hohen Grad an Robustheit aufweisen. Schwierigere Wetterverhältnisse dürfen kein Problem für das Boot darstellen und Gefahren müssen präventiv erkannt werden.

Sicherheit Die Stabilität des Systems darf durch äußere Einflüsse nicht gefährdet oder eingeschränkt werden. Kollisionsrisiken im Nahbereich müssen zuverlässig erkannt werden, um Zusammenstöße mit Schwimmern oder Tieren zu vermeiden. Mögliche Hindernisse und detektierbare Gefahrenbereiche sollen stets mit einem ausreichenden Sicherheitsabstand umfahren werden.

Wartbarkeit Alle verwendeten Bauteile, Schaltpläne und Software-Module sollten in der schriftlichen Ausarbeitung ausführlich und nachvollziehbar dokumentiert werden. Der Aufbau der im System verbauten Komponenten sollte sich an den gängigen Richtlinien und Standards orientieren. Insbesondere bei elektronischen Schaltungen ist auf die korrekte Kennzeichnung der einzelnen Leitungen zu achten. Auch der Programmcode ist zu dokumentieren.

Zuverlässigkeit Zu der Vollständigkeit des Systems gehört ebenfalls die Zuverlässigkeit der gesammelten Daten. Sensoren müssen für die Einsatzgebiete geeignet sein. Außerdem müssen Vorrichtungen geschaffen werden, um den korrekten Einsatz der Sensoren zu ermöglichen.

3.2 Stakeholder

Bei der Durchführung der Projektgruppe MOPS IV haben unterschiedliche Personen Interesse an dem Fortschreiten des Projektes. Diese sogenannten Stakeholder werden im Folgenden aufgeführt und kurz erläutert. Abbildung 3.1 fasst die unterschiedlichen Stakeholder in einer Grafik zusammen.

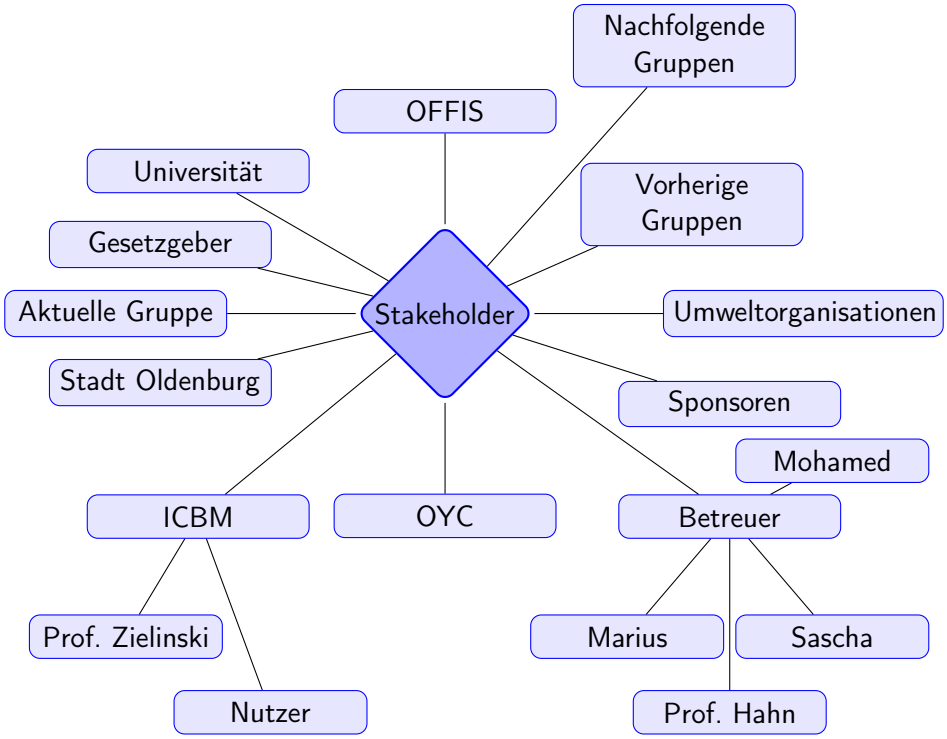


Abbildung 3.1: Relevante Stakeholder

OFFIS Das OFFIS steht unter verschiedenen Aspekten in Zusammenhang mit den Projekten MOPS I-IV. Unter anderem stellt es die Räumlichkeit für die Projektgruppe zur Verfügung. Hier werden sowohl Gruppentreffen als auch Montagearbeiten an der Plattform durchgeführt.

Universität Die Universität ist im weiteren Sinne Organisator der Lehrveranstaltung zur durchgeführten Projektgruppe. Die Teilnehmer durchlaufen im Rahmen ihres Studiums die Projektgruppe als Teil ihres Pflichtstudienplans. Darüber hinaus haben möglicherweise einige Abteilungen der Universität Interesse an den Ergebnissen, die die Projektgruppe erzielt, um diese in ihre eigenen Forschungen einzubeziehen.

Gesetzgeber Der Gesetzgeber spielt für das Durchführen sämtlicher Praxisversuche mit dem Wasserfahrzeug eine wichtige Rolle. Im Rahmen der Experimente auf dem Tweelbäker See müssen gesetzliche Richtlinien eingehalten werden. Sollte die Plattform langfristig auf Flüssen oder im Meer eingesetzt werden, so ist auch dies unter Berücksichtigung der geltenden Gesetze zu realisieren.

Aktuelle Gruppe Die Mitglieder der Projektgruppe führen das Projekt durch. Von den Teilnehmern werden die Ziele des Projektes festgelegt und im Anschluss bearbeitet. Die Mitglieder sind für das erfolgreiche Durchführen und Planen sämtlicher Aufgaben eigenverantwortlich.

Stadt Oldenburg Die Stadt Oldenburg tritt im Rahmen der Projektgruppe primär als Betreiber des Tweelbäker Sees und des Oldenburger Hafens auf. Mit der Stadt ist Rücksprache für reibungslose Feldtests zu halten, um so kein Gefährdungspotential für Experimente zu erzeugen.

ICBM Das Institut für Chemie und Biologie des Meeres ist der Auftraggeber der vergangenen Projektgruppen (MOPS I-III) gewesen. Hier ist besonders Prof. Dr. Oliver Zielinksi zu nennen, der Ansprechpartner für die Projektgruppe ist. Des Weiteren haben die Mitarbeit des ICBM Interesse daran, das Forschungsboot für ihre eigenen Forschungsvorhaben zu verwenden. Seitens der Projektgruppe kann von den Mitarbeitern und ihrem Fachwissen im Bereich der maritimen Sensorik profitiert werden.

OYC Der Oldenburger Yacht-Club betreibt einen Hafen an der Hunte. Für die Projektgruppe ist dies besonders interessant, da die Praxistests auf dieses Gebiet erweitert werden könnten. Dies würde dem Ziel, das Wasserfahrzeug langfristig im Meer einzusetzen, ein deutliches Stück näher bringen. In diesem Gebiet könnten beispielsweise Seezeichen ausgewertet werden. Außerdem stellt der Club am Tweelbäler See Ausrüstung zur Verfügung.

Betreuer Die Betreuer Prof. Dr. Axel Hahn, Sascha Hornauer, Mohamed Abdelaal und Marius Brinkmann sind Ansprechpartner für die Mitglieder der Projektgruppe. Sie sind an dem Fortschreiten des Projektes interessiert und geben Impulse für die Definition der erwarteten Ziele der Projektgruppe. Außerdem wird die Bewertung des Projektes am Ende der Bearbeitungszeit von den Betreuern vorgenommen. Des Weiteren sind Sascha Hornauer, Mohamed Abdelaal und Marius Brinkmann beratend bei den Gruppensitzungen anwesend.

Sponsoren Sponsoren erleichtern die Finanzierung zukünftiger Bauteile oder Sensoren. Da das Projekt nur über begrenzte finanzielle Ressourcen

verfügt, ist ein Sponsoring für die Projektgruppe ein willkommenes Mittel finanzielle Unterstützung zu erhalten.

Umweltorganisationen Da die Aufgabe des Wasserfahrzeugs darin besteht Gewässer zu untersuchen, könnten Umweltorganisationen an den Ergebnissen, die die Plattform liefert, interessiert sein.

Vorherige Projektgruppen Die Mitglieder der vergangenen Projektgruppen sind Ansprechpartner für grundlegende Fragen, die das bisher entwickelte Boot betreffen. Darüber hinaus haben die früheren Teilnehmer die Dokumentation, Software und Hardware für die Projektgruppe zur Verfügung gestellt. Damit ist die bisher geleistete Arbeit eine wichtige Informationsquelle.

Nachfolgende Projektgruppen Den Teilnehmern der nachfolgenden Projektgruppen wird das montierte Wasserfahrzeug mit sämtlicher Dokumentation übergeben. Auf Grundlage der entwickelten Komponenten können sie das Projekt fortsetzen.

3.3 Rahmenbedingungen

Bei dem Erheben der funktionalen und nicht-funktionalen Anforderungen sowie der Exploration des Entwurfsraumes unterliegen sowohl das zu realisierende Gesamtsystem als auch der Entwurfsprozess selbst zahlreichen Rahmenbedingungen, die im Verlauf dieses Kapitels vorgestellt werden.

R-1 Das System wird in deutschen Binnengewässern eingesetzt.

R-2 Das System ist für eine Einsatzdauer von acht bis zwölf Stunden konzipiert.

R-3 Das System muss Einsatztemperaturen von -5 bis $40\,^{\circ}\mathrm{C}$ standhalten.

R-4 Das System kann nur bei guten Sichtverhältnissen, d.h. bei einer Mindestsichtweite von 10 m, eingesetzt werden.

3.3.1 Software

Die bereits vorhandene Software zur Steuerung des Gesamtsystems ermöglicht einen stabilen Betrieb der Plattform. Insgesamt jedoch ist diese in Hinsicht auf einige Funktionalitäten und die Benutzerfreundlichkeit nicht ausgereift. Deswegen werden im Folgenden zuerst die nötigen Anwendungsfälle an die

Software und deren Verwendung gestellt, aus welchen dann die Erhebung der funktionalen und nicht funktionalen Anforderungen folgt.

Anwendungsfälle

Generell gibt es bei der Verwendung der Plattform zwei verschiedene Akteure. Zum einen Nutzer, deren Ziel es ist, diese zur autonomen Überprüfung der Wasserqualität in Gewässern zu verwenden. Dementsprechend beschränken sich die Anwendungsfälle auf die reine Bedienung des Bootes.

Zum anderen Entwickler, deren Anwendungsfälle sich zusätzlich um die Konfiguration, die manuelle Steuerung sowie die Simulation der Plattform in einer Entwicklungsumgebung erweitern. Abbildung 3.2 stellt diese in in einem Anwendungsfalldiagramm dar.

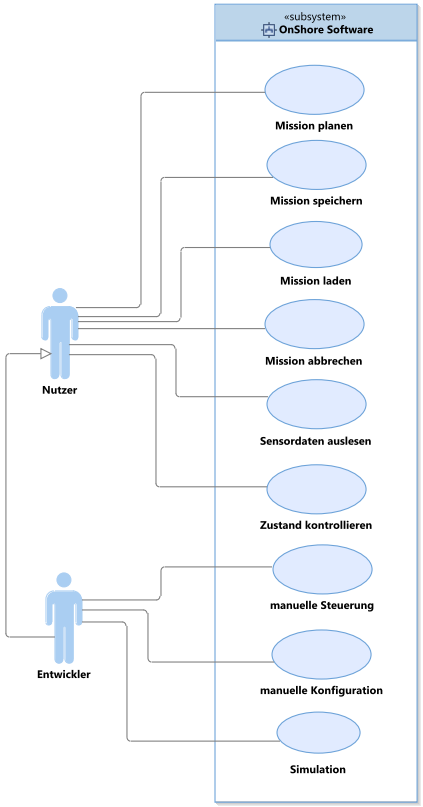


Abbildung 3.2: Anwendungsfalldiagramm „Onshore-Software“

Mission planen Das Planen einer Mission erfolgt innerhalb der Onshore-Software. Dabei muss der Nutzer Wegpunkte auf einer Karte festlegen, welche

das Boot anfahren und Messdaten messen soll. Die Planung darf dabei keine invaliden Missionen zulassen. Um dies zu ermöglichen sollten Gefahrenbereiche sowie das mathematische Modell des Bootes in der Onshore-Software dargestellt werden. Eine Mission ist dann valide, wenn die Wegpunkte und die Route nicht innerhalb von Gefahrenzonen liegen.

Kurzbeschreibung	Mithilfe der Onshore-Software wird eine Mission zur Überwachung der Wasserqualität eines Gewässers geplant.
Akteure	Nutzer
Auslöser	Die Wasserqualität eines Gewässers soll überprüft werden.
Vorbedingung	Die Onshore-Software steht zur Verfügung.
Ergebnis	Ein valider Missionsplan steht zur Verfügung, mit welchem die Wasserqualität eines spezifischen Gewässers überprüft wird.
Genereller Ablauf	(1) Nutzer startet die Onshore-Software (2) Nutzer wählt Punkte aus, an denen Sensordaten zur Wasserqualität aufgenommen werden sollen

Mission speichern Während der Planung einer Mission in der Onshore-Software soll der Nutzer die Möglichkeit besitzen den aktuellen Zustand der Planung zu speichern.

Kurzbeschreibung	Eine bereits erstellte Mission wird extern gespeichert.
Akteure	Nutzer
Auslöser	Ein Nutzer will eine geplante Mission speichern, um sie zu einem späteren Zeitpunkt weiter zu bearbeiten oder zu starten.
Vorbedingung	Teile einer Mission müssen bereits geplant sein.
Ergebnis	Ein valider Missionsplan steht zur Verfügung, mit welchem die Wasserqualität eines spezifischen Gewässers überprüft wird.

Genereller Ablauf	(1) Nutzer startet die Onshore-Software (2) Nutzer arbeitet an einer Mission (3) Nutzer speichert die Mission auf einen externen Datenträger

Mission laden Eine bereits gespeicherte Version einer Mission soll von einem Nutzer wieder in die Onshore-Software geladen können. Das Speichern und Laden von Missionen erhöht die Benutzerfreundlichkeit der Software und kann vor Datenverlust schützen, falls die Onshore-Software abstürzt.

Kurzbeschreibung	Eine extern gespeicherte Mission wird in die Onshore-Software geladen.
Akteure	Nutzer
Auslöser	Eine bereits gespeicherte Mission soll wiederverwendet werden.
Vorbedingung	Der Nutzer besitzt eine zuvor gespeicherte Mission.
Ergebnis	Die zuvor gespeicherte Mission wurde in die Onshore-Software geladen und kann nun weiter bearbeitet werden.
Genereller Ablauf	(1) Nutzer startet die Onshore-Software (2) Nutzer lädt eine zuvor gespeicherte Mission in die Onshore-Software

Mission starten Eine Valide Mission, die auf eine Bootsinstanz geladen wurde, soll vom Nutzer gestartet werden können.

Kurzbeschreibung	Eine geplante Mission wird gestartet, wenn sie valide ist.
Akteure	Nutzer
Auslöser	Ein Boot soll eine Valide Mission zur Überwachung der Wasserqualität starten.

Vorbedingung	(1) Eine Valide Mission muss in der Onshore-Software vorliegen (2) Das Boot muss zu Wasser gelassen sein und betriebsbereit sein (3) Die Mission muss auf das Boot übertragen worden sein
Ergebnis	Das Boot beginnt mit der Durchführung der Mission.
Genereller Ablauf	(1) Boot ist zu Wasser gelassen und betriebsbereit (2) Nutzer hat eine Valide Mission in der Onshore-Software erstellt oder geladen (3) Nutzer überträgt Mission auf das Boot (4) Nutzer startet Mission

Mission abbrechen Eine Nutzer soll die Möglichkeit besitzen, eine bereits gestartet Mission in der Onshore-Software abzubrechen.

Kurzbeschreibung	Eine bereits gestartete Mission wird abgebrochen, dabei fährt das Boot zurück zur Basis.
Akteure	Nutzer
Auslöser	Das Boot befindet sich in einer Gefahrensituation oder die Mission soll aus anderen Gründen abgebrochen werden.
Vorbedingung	(1) Das Boot befindet sich auf einer Mission (2) Es besteht eine aktive Verbindung zum Boot
Ergebnis	Das Boot bricht seine aktuelle Mission ab und befindet sich auf dem Weg zur Basis.
Genereller Ablauf	(1) Nutzer ruft über de Onshore-Software den Abbruch der Mission hervor (2) Das Boot bewegt sich zurück zur Basis

Sensordaten auslesen Die Sensordaten, die während eines Missionsdurchlaufs aufgenommen wurden, sollen von einem Nutzer nach der Mission exportiert werden können.

Kurzbeschreibung	Nach einer abgeschlossenen Mission exportiert der Nutzer die aufgenommenen Sensordaten.
Akteure	Nutzer
Auslöser	Die Sensordaten einer Mission sollen exportiert werden.
Vorbedingung	Eine Mission wurde erfolgreich abgeschlossen.
Ergebnis	Die Sensordaten liegen exportiert bereit zur Weiterverarbeitung vor.
Genereller Ablauf	(1) Boot beendet Mission (2) Nutzer exportiert Ergebnisse der Sensoren auf einen externen Datenträger

Zustand kontrollieren Der aktuelle Zustand des Bootes soll während des Betriebs kontrolliert werden können. Dazu gehören die aktuelle Position, Geschwindigkeit, Ausrichtung, aktuelles Ziel und weitere Systeminformationen.

Ein skizzierter Anwendungsfall sieht vor, dass das Boot nach dem Start einer Mission alleine gelassen wird und beim Abschluss abgeholt wird. Während der Durchführung ist es vorstellbar, dass der Nutzer sich räumlich entfernt.

Kurzbeschreibung	Überprüfung des Zustands des Bootes.
Akteure	Nutzer
Auslöser	Der Nutzer will sich einen Überblick über den Zustand des Bootes verschaffen.
Vorbedingung	Es besteht eine aktive Verbindung zum Boot.
Ergebnis	Der Nutzer hat die Möglichkeit, den aktuellen Zustand des Bootes zu überprüfen.
Genereller Ablauf	(1) Nutzer fragt über eine Schnittstelle den Zustand des Bootes an (2) Nutzer erhält Informationen zum Zustand des Bootes

Manuelle Konfiguration Um die Benutzerfreundlichkeit für Entwickler zu erhöhen, sollen diese die Möglichkeit besitzen, während des laufenden Betriebs diverse Systemparameter wie zum Beispiel die Auswahl von Reglertypen anzupassen.

Kurzbeschreibung	Ein Entwickler passt diverse Systemparameter während des laufenden Betriebs an.
Akteure	Entwickler
Auslöser	Ein anpassbarer Systemparameter soll einen neuen Wert erhalten.
Vorbedingung	Es besteht eine aktive Verbindung der Onshore-Software zum betriebsbereiten Boot.
Ergebnis	Das Boot erhält einen neue Konfiguration.
Genereller Ablauf	(1) Der aktuelle Parametersatz wird vom Entwickler über die Onshore-Software vom Boot angefordert (2) Der Entwickler kann die konfigurierbaren Parameter anpassen (3) Die angepassten Parameter werden zurück an das Boot übertragen

Manuelle Steuerung Nutzer sollen die Möglichkeit besitzen, jederzeit die manuelle Steuerung des Bootes zu übernehmen.

Kurzbeschreibung	Manuelle Steuerung des Bootes.
Akteure	Entwickler
Auslöser	Der Entwickler sieht einen Grund die manuelle Steuerung des Bootes zu übernehmen
Vorbedingung	Das Boot befindet sich betriebsbereit in einem Gewässer und ist in Funkreichweite der Fernbedienung.
Ergebnis	Der Entwickler besitzt die volle Kontrolle über die Steuerung des Bootes.
Genereller Ablauf	(1) Entwickler startet die Fernbedienung (2) Entwickler steuert das Boot manuell über die Fernbedienung

Simulation Entwickler sollen die Möglichkeit besitzen, die Hardwarekomponenten des Bootes zu simulieren. Dies vereinfacht die Entwicklung des Gesamtsystems und ermöglicht die frühe Identifikation möglicher Fehlerquellen.

Kurzbeschreibung	Ein Entwickler simuliert zur Entwicklung die Hardwarekomponenten des Bootes, ohne dass das Boot aufgebaut werden muss.
Akteure	Entwickler
Auslöser	Die Softwarekomponenten des Bootes sollen weiterentwickelt werden.
Vorbedingung	Der Sourcecode sowie eine geeignete Entwicklungsumgebung stehen zur Verfügung.
Ergebnis	Der Entwickler kann an den Softwarekomponenten des Bootes weiterentwickeln, während die Hardware simuliert wird.
Genereller Ablauf	(1) Entwickler aktiviert die Simulation der Hardwarekomponenten (2) Entwickler arbeitet an der Software und kann diese Testen

3.3.2 Kollisionserkennung

Das aktuell eingesetzte System verfügt über keine Möglichkeit, Hindernisse zur Laufzeit zu erkennen. Lediglich im Vorfeld durch den Nutzer festgelegte Sperrzonen werden vom Wasserfahrzeug in der Kursplanung berücksichtigt. Unter realen Bedingungen kommt es jedoch vor, dass unvorhergesehene Hindernisse auf der Missionsroute auftauchen können.

Um Kollisionen mit Lebewesen oder anderen Objekten zu vermeiden, ist es notwendig, diese rechtzeitig zu erkennen. Während der Realisierung gilt es also, effiziente Methoden für eine effektive Objekterkennung, eine rechtzeitige Identifikation der detektierten Objekte sowie eine eigenständige Bewertung der vorherrschenden Kollisionsrisiken durchzuführen.

Die in Abschnitt 3.4.1 erhobenen Anforderungen lassen dieses Bestreben in den systematischen Entwurfsprozess einfließen und bilden eine wichtige Grundlage für die spätere Realisierung.

3.3.3 Kollisionsverhütung

Analog zu den im vorherigen Abschnitt beschriebenen Herausforderungen bei der Erkennung von Kollisionsrisiken verfügt das bestehende System über keine geeignete Algorithmik, die eine Berücksichtigung dynamischer Hindernisse ermöglicht. Dies führt dazu, dass bei spontan auftretenden Hindernissen keine Möglichkeit besteht, kollisionsverhütende Maßnahmen einzuleiten. Der Algo-

rithmus zur Kursplanung soll daher erweitert werden, um geeignete Methoden für eine effektive Kollisionsvermeidung bereitzustellen.

Um einen Kurs präzise zu befahren, ist weiterhin eine möglichst exakte Regelung erforderlich. Im Rahmen der Anforderungsanalyse für die Kollisionsverhütung werden daher auch die Erwartungen an die Regelung des Wasserfahrzeugs erfasst. Um ein strukturiertes Vorgehen bei der Bearbeitung zu gewährleisten, erfolgen die nachfolgenden Betrachtungen in den drei übergeordneten Kategorien Kursberechnung/Bahnplanung, Erkennen von physikalischen Grenzfällen und Regelung/Mathematisches Modell.

Kursberechnung/Bahnplanung

Wenn ein Hindernis die vorgesehene Route des Wasserfahrzeuges kreuzt, ist eine Kollisionsvermeidung zwingend durchzuführen. Hierfür muss der zuvor berechnete Kurs korrigiert werden. Da die aktuelle Implementierung der Kursplanung keine dynamischen Hindernisse berücksichtigt, ist dies ein primäres Ziel der Projektgruppe. Neben dem automatischen Korrigieren des Kurses beim Auftritt eines dynamischen Hindernisses soll der Bahnplanungsalgorithmus auch simuliert werden können. Dies dient dem Zweck auch bei Trockentests die Funktionalität überprüfen zu können.

Um den berechneten Kurs auch anderen Systemkomponenten zur Verfügung zu stellen, ist es notwendig, dass die Kursinformationen gespeichert werden und auch nach einem Systemausfall zur Verfügung stehen. Hierfür ist ein geeignetes Speicherkonzept von Bedeutung.

Berechnet der Bahnplanungsalgorithmus einen Kurs, so sollen ausgewählte Umgebungsparameter, wie beispielsweise der Abstand zum Hindernis oder die vorherrschenden Umwelteinflüsse berücksichtigt werden. Der Kurs darf darüber hinaus nicht durch eine Sperrzone verlaufen. Wenn der Algorithmus eine befahrbare Route ermittelt, muss dies den übrigen Systemkomponenten mitgeteilt werden. Ist dies nicht der Fall und eine Kollision ist nicht zu vermeiden, soll ein akustisches Signal zur Warnung ausgegeben und die Motoren abgeschaltet werden, um größere Schäden zu verhindern.

Erkennen von physikalischen Grenzfällen

Um die Verfahren zur Kollisionsvermeidung nutzen zu können, muss die Plattform in einem definierten Einsatzgebiet agieren. Um die Position des Wasserfahrzeugs möglichst exakt zu bestimmen, sollte ein GPS-Signal ausgewertet werden. Zur Überprüfung der Plausibilität des GPS-Signals können dabei auch andere Medien herangezogen werden.

Um einen Überblick über den Einsatzraum zu gewinnen, sollten die Grenzen in der Onshore-Software visualisiert werden. Die Grenzen des Einsatzgebiets sollten automatisch als Sperrzone definiert werden. Hierdurch wird sichergestellt, dass die Grenzen nicht überschritten werden.

Damit das Wasserfahrzeug nicht in eine Sperrzone getrieben wird, müssen auch die Richtung und die Geschwindigkeit der Strömung in die Berechnungen des Bahnplanungsalgorithmus einbezogen werden. Analog hierzu ist eine Berücksichtigung der Windrichtung und -geschwindigkeit erforderlich.

Regelung/Mathematisches Modell

Um ein stabiles Systemverhalten zu garantieren, muss als Grundlage die Auswahl geeigneter Regler stattfinden. Hierzu muss jeweils ein passender Parametersatz identifiziert werden.

Zur Implementierung einer optimalen Regelung müssen weiterhin mögliche Störgrößen bestimmt oder abgeschätzt werden. Hierdurch soll das System möglichst schnell in einen stabilen, ausgeregelten Zustand überführt werden.

Um das Systemverhalten möglichst genau vorhersagen zu können, ist es notwendig, ein passendes mathematisches Schiffsmodell zu erstellen. Um überall konsistente Daten zu haben, sollen die Daten aus Regelung und mathematischem Modell allen anderen Systemkomponenten zur Verfügung gestellt werden. Die an den Ein- und Ausgängen des Reglers anliegenden Werte sollten zudem in der Onshore-Software visualisiert werden, um das Debugging durch die Entwickler zu vereinfachen.

3.3.4 Bootsaufbau und Seetauglichkeit

Im Rahmen des Projektvorhabens MOPS IV möchten wir zudem eine nachhaltige Verbesserung der Seetauglichkeit vornehmen. Das Boot soll also in der Zukunft der Anforderungen der Seetauglichkeit gerecht werden.

In einem ersten Schritt möchten wir hierfür die Stabilität gegenüber auftretenden Wellen verbessern. Außerdem ist eine Optimierung des Bootskörpers sowie eine ausführliche Betrachtung der Stealthproblematik vorgesehen. Das heißt, dass beispielsweise Signallichter an das Boot angebracht werden müssen. Auch die Verbesserung der Usability ist ein wichtiges Thema, sodass im Verlauf des Projekts Regeln für die Vor- und Nachbereitung eines möglichen Einsatzes sowie zur Wartung des Systems aufgestellt werden müssen.

Als nächstes möchten wir die Bestandteile des Bootes erläutern. Danach soll eine Anforderungserhebung aufgestellt werden und eine Analyse stattfinden, in wie weit das Boot verbessert werden kann.

Bestandteile Sobald Änderungen am Boot vorgenommen werden, müssen die Maße beachtet werden, um die Fahreigenschaften nicht negativ zu beeinflussen. Das Boot misst eine Länge von 240 cm, besitzt eine Breite von 126 cm und hat eine Höhe von 37 cm. Der Tiefgang des Bootes hängt von dem Gewicht der Zuladung ab.

Das Boot kann eine Höchstgeschwindigkeit von sechs Knoten erreichen, was ungefähr einer Geschwindigkeit von 11 km/h entspricht. Als Überdachung

dient bisher eine einfache Plane, die allerdings nur vor leichten Regen schützen kann. Eine Konstruktion die gegen hohen Wellengang oder starken Regen schützen kann fehlt. Das bedeutet auch, dass es bisher keinen richtigen Schutz für die auf dem Boot befindlichen Hardware gibt.

Einsatzgebiet Bisher wurde die Plattform hauptsächlich auf dem Tweelbäker See eingesetzt. Es ist jedoch mittelfristig geplant, diese auch auf einigen Kanälen des Emsgebietes einzusetzen. Sollte das Boot später auch auf anderen Binnengewässern, wie beispielsweise der Hunte, getestet werden, sind die nachfolgenden Besonderheiten zu beachten:

Verkehrsteilnehmer Auf der Hunte sind Europaschiffe und Küstenmotorschiffe bis 1000 BRT zulässig [121].

Europaschiffe Europaschiffe sind Binnenschiffstypen, die eine maximale Länge von 85 m besitzen, 9.5 m breit sind, eine Tauchtiefe von 2.5 m haben und 1350 t Transportkapazität besitzen [98].

Küstenmotorschiffe Küstenmotorschiffe sind motorisierte Frachtschiffe. Sie transportieren Container, Stück- und Schüttgut und kommen auf Flüssen und nahe der Küste zum Einsatz [124].

Wellengang Auf der Hunte ist mit kleinen Bugwellen der Europa- und Küstenmotorschiffen zu rechnen.

Strömung Laut verschiedenen Foren, sind auf der Hunte Strömungen zwischen 2 bis 5 km/h zu erwarten.

Zugang zum Wasser Der Oldenburger Yacht-Club e.V. besitzt am Oldenburger Hafen Slipanlagen zur Wasserung des Bootes [30].

Wassertiefe und -breite Die Hunte besitzt eine Kanalbreite von 10 bis 15 m und ist je nach Abschnitt zwischen 2 und 3.5 m tief [121] [123].

Beachtung der Gezeiten Die Gezeiten der Hunte werden kostenlos vom Bundesamt für Schifffahrt und Hydrographie bereitgestellt. Es lassen sich die Gezeiten für eine Woche im Voraus ablesen [40].

Anwendungsfälle

Anhand von konkreten Anwendungsfaellen sollen hier die funktionalen Anforderungen näher erläutert werden.

Kurzbeschreibung	Das Boot ins Wasser lassen.
Akteure	Nutzer

Auslöser	Der Nutzer möchte das Boot starten und lässt es zu Wasser.
Vorbedingung	Das Boot wurde zum Wasser transportiert und ist im betriebsbereiten Zustand.
Ergebnis	Das Boot befindet sich funktionsfähig im Wasser.
Genereller Ablauf	(1) Der Nutzer befördert das Boot zum Ufer (2) Der Nutzer arbeitet die Checkliste ab (3) Das Boot wird zu Wasser gelassen

Kurzbeschreibung	Das Boot aus dem Wasser befördern.
Akteure	Nutzer
Auslöser	Der Nutzer möchte das Boot aus dem Wasser holen.
Vorbedingung	Das Boot ist funktionsfähig zum Ufer zurückgekehrt.
Ergebnis	Das Boot ist zurück an Land.
Genereller Ablauf	(1) Der Nutzer holt das Boot aus dem Wasser (2) Der Nutzer arbeitet die Checkliste ab

Bewertungskriterien

Bei der Überarbeitung und Erweiterung der Plattform ist die Berücksichtigung der nachfolgenden Kostenfaktoren und Bewertungskriterien sinnvoll:

Kosten Das Projekt sollte nicht zu viele Kosten auf sich ziehen. Jeglicher Zukauf neuer Ausrüstung muss in der Gruppe und mit den verantwortlichen Betreuern abgesprochen und dokumentiert werden. Die Funktionalität der Ausrüstung steht im Vordergrund.

Beachtung anderer Funktionen Die Aufrüstungen sollen im Einklang mit anderen Projektzielen sein. Durch ein breiteres Boot könnte beispielsweise die Kollisionsverhütung negativ beeinflusst werden. Ebenso könnte es die Abstimmung mit der Software und damit die effektive Wegfindung stören.

Berücksichtigung der Umwelt Das System muss die geltenden Umweltgesetze beachten. Auch andere Verkehrsteilnehmer müssen berücksich-

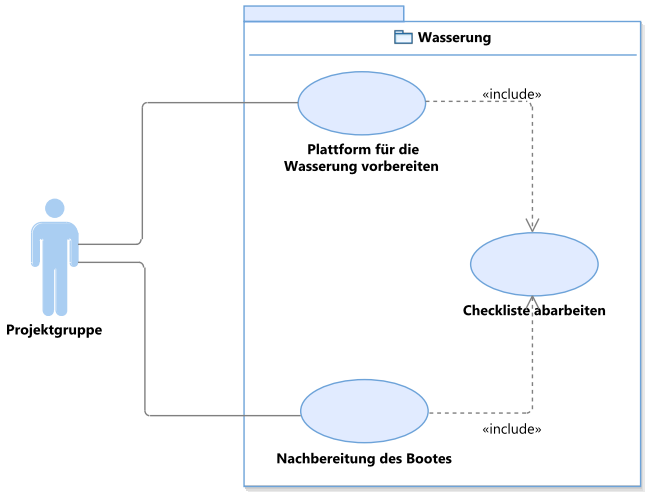


Abbildung 3.3: Anwendungsfalldiagramm „Wasserung“

tigt werden. Vermieden werden sollen beispielsweise Signale, die andere Verkehrsteilnehmer stören oder jegliche Beschädigungen der Natur.

Zuverlässigkeit und Nachhaltigkeit Insgesamt sollten Nachrüstungen auf Dauer zuverlässig arbeiten und nicht kurzlebig sein. Demnach muss es auch von nachfolgenden Projektgruppen verwendbar sein. Zudem muss das Boot bei der Teilnahme am Schiffsverkehr und auf der See den entsprechenden Bedingungen standhalten.

Sicherheit Das Boot darf zu keiner Zeit eine Gefahr für andere Verkehrsteilnehmer sein. Auch die Plattform selbst muss vor kentern oder Ähnliches geschützt sein. Ebenso gilt es, in der Umgebung agierende Tiere und andere Personen zu schützen. Eine korrekte Einschätzung des Einsatzgebietes ist hier dringend erforderlich.

Ergonomie Durch eine Optimierung der Bootsform soll ein geringerer Widerstand entstehen. Dadurch soll das Boot besser im Wasser gleiten und stabiler ausgerichtet sein. Der Einfluss auf die Software muss beachtet werden.

Verfügbarkeit Gesetzlich vorgeschriebene Funktionen, wie das Rundumlicht, müssen zu jeder Zeit voll funktionsfähig sein.

Wartung Neue Hardware und Technik muss sich durch eine einfache und effektive Wartbarkeit auszeichnen. Das heißt, dass System muss leicht zu reparieren und austauschbar sein.

Mögliche Verbesserungen

In diesem Abschnitt werden die gewünschten Verbesserungen vorgestellt. Diese sollen das Boot in der Zukunft sicherer und gleichzeitig auch effizienter machen. Es soll also sichergestellt werden, dass das Boot beispielsweise besser wahrgenommen wird.

Dachkonstruktion Um starkem Regen und eintretendes Wasser durch hohe Wellen entgegenzusetzen, muss eine Dachkonstruktion entwickelt werden, die kein Wasser durchlässt. Bei einer einfachen Plane besteht die Möglichkeit, dass Pfützen entstehen, die das Boot schwerer machen und es eventuell zum kentern bringt. Das Wasser muss also irgendwie abfließen können. Eine Spitzdachkonstruktion wäre gut denkbar.

Stabilitätsverbesserung Die Verbesserung der Stabilität ist ein wichtiger Punkt, damit das Boot besser gegen raues Wetter und auftretenden Wellen geschützt ist. Wie in Abschnitt Einsatzgebiet beschrieben, kann es auf den Kanälen zu Bugwellen von anderen Schiffen kommen. Um die Stabilität des Bootes zu verbessern, bieten sich mehrere Möglichkeiten an. Genauer betrachten möchten wir die Schlingerkiele.

Schlingerkiel Schlingerkiele versuchen die rollenden Bewegungen des Bootes um die Längsachse zu verringern. Sie sind flache Stahlprofile und werden an beiden Seiten des Bootes angebracht, genauer gesagt, an der Kimm. Das ist der Übergang vom Schiffboden in den Seitenwänden und verlaufen auf der dort, wo der Schiffrumpf die größte Breite besitzt [127]. Schlingerkiele sind so bemessen, dass sie etwa die Hälfte der Schiffslänge ausmachen und werden meist an die mittlere Partie angebracht. Hier kann der Schlingerkiel die größte Widerstandsarbeit leisten [78]. Schlingerkiele sind kostengünstig anzubringen und außerdem wartungsfrei [126].

Abbildung 3.4: Abbildung eines Schlingerkiels [126]

Schlingertank Bei einem Schlingertank befinden sich an den Längsseiten des Schiffes, wassergefüllte Tanks, die so hoch wie möglich liegen und durch Rohre miteinander verbunden sind. Dieser Tank versucht Schwankungen um die Längsachse zu dämpfen und funktioniert wie ein Wasserpendel. Ein Schlingertank ist also auf die Eigenresonanz des Schiffes abgestimmt. Fängt das Schiff durch seitliche Wellen zum Rollen an, fließt das Wasser zwischen den Tanks vor und wieder zurück. Durch die gegenphasige Bewegung des Wasser, reduziert es das Schlingen des Schiffes [128].

Stealth-Problematik

Anstatt von Topp- und Hecklichtern, wird für das Boot lediglich ein weißes Rundumlicht benötigt. Das liegt daran, dass es die Länge von 7 Metern nicht überschreitet und auch die Höchstgeschwindigkeit 7 Knoten nicht Übersteigt. Außerdem müssen, sofern es möglich ist, Seitenlichter angebracht werden [5].

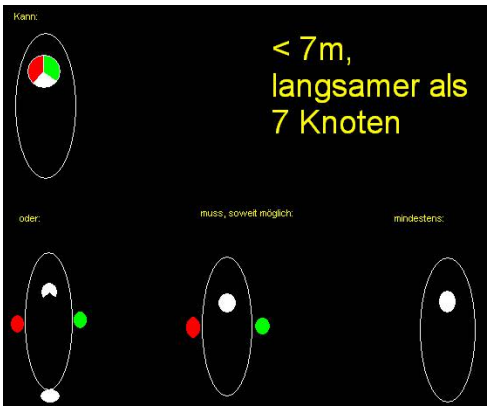


Abbildung 3.5: Anbringungsmöglichkeiten von Signallichtern (vgl. [94])

Ein solches Rundumlicht ist schon für relativ wenig Geld, ca. 10 Euro - 25 Euro, zu bekommen [94].

Damit auf der Binnenschifffahrt Gefahren- und Warnsignale abgegeben werden können, etwa bei schlechter Sicht oder wenn das Schiff manövrierunfähig ist, braucht die Plattform ein Signal- bzw. Nebelhorn.

Verbesserung der Usability und Wartung

Sollte das Boot später über eine sogenannte Sliprampe oder auch Slipanlage zu Wasser gelassen werden, bedarf dies einer gewissen Vorbereitung. Hierfür möchten wir eine Checkliste aufstellen, die dann abgearbeitet werden muss. Eine vorläufige Checkliste soll nun dargestellt werden:

- Vorbereitung des Bootes

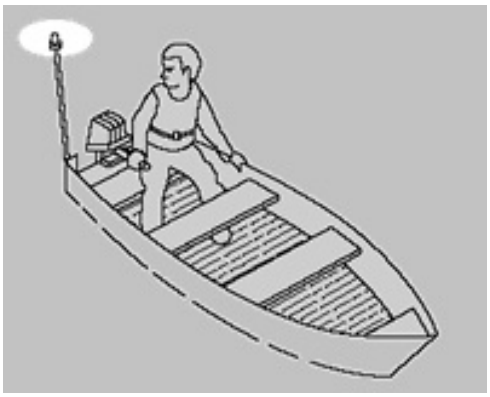


Abbildung 3.6: Rundumlicht am Heck des Bootes (vgl. [5])

  - Abnehmen der Plane, Persenning
  - Aufrichten der Funkantenne
  - Aufrichten des Signallichts
  - Abnehmen verschiedener Sicherungen, etwa für den Antrieb
  - Batterieladung überprüfen
- Überprüfung der Funkverbindung zum Rechner
- Technische Anlagen überprüfen
- Slipanlage überprüfen
  - Ausrichten
  - Sicherungen lösen
- Umgebung überprüfen
- Auf die Rampe fahren und das Boot zu Wasser lassen

Außerdem sollte folgender Punkt beachtet werden: Ist das Boot zurück, sollte dies nicht auf der Slipanlage für den abtransport vorbereitet werden. Denn wenn in der Zeit ein anderes Boot die Rampe beansprucht, kann dies zu einer Stresssituation führen. Das bedeutet, dass eventuell wichtige Dinge in der Hektik vergessen werden. Gleiches gilt natürlich auch beim Wassern des Bootes. Deswegen wird das Boot immer neben der Rampe vorbereitet [110].

3.4 Anforderungen

Das folgende Kapitel fasst die Ergebnisse der Anforderungsanalyse zu Beginn der Bearbeitungszeit zusammen. Im Rahmen dieser ersten Arbeitsphase wurden sowohl funktionale als auch nicht-funktionale Anforderungen erhoben.

3.4.1 Funktionale Anforderungen

Unter funktionalen Anforderungen werden Anforderungen verstanden, die die erwartete Funktionalität des zu entwickelnden Systems beschreiben. Für eine bessere Strukturierung wurde eine Einteilung in die Bereiche *Software*, *Kollisionserkennung* und *Kollisionsverhütung* vorgenommen.

Software

Die nachfolgenden Anforderungen wurden aus den Rahmenbedingungen der Software abgeleitet und bilden eine vollständige Beschreibung der Erwartungen an die entsprechenden Komponenten.

F-SW-1	Die Software muss es dem Nutzer ermöglichen, eine Mission zu planen.
F-SW-1.1	Das Planen einer Mission soll durch die Darstellung des mathematischen Modells vereinfacht werden.
F-SW-1.2	Das Planen einer Mission soll durch die automatische Kennzeichnung von Gefahrenzonen vereinfacht werden.
F-SW-2	Eine erstellte Mission muss vom Nutzer gespeichert werden können.
F-SW-3	Eine Mission muss auf einen externen Datenträger exportiert werden können.
F-SW-4	Eine gespeicherte Mission muss geladen werden können.
F-SW-5	Eine exportierte Mission muss importiert werden können.
F-SW-6	Eine zu startende Mission muss valide sein.
F-SW-7	Zum Starten einer Mission muss eine Verbindung zum Boot bestehen.
F-SW-8	Eine gestartete Mission muss abgebrochen werden können.

F-SW-8.1	Nach dem Abbruch einer Mission muss das Boot zur Basis zurückkehren.
F-SW-8.2	Der Pfad zur Basis muss valide sein.
F-SW-9	Nach dem Abschluss einer Mission sollen die aufgenommenen Sensordaten dem Nutzer zum Export bereitstehen.
F-SW-10	Der aktuelle Zustand des Bootes muss durch den Nutzer überwacht werden können.
F-SW-10.1	Der Soll-Kurs muss überwacht werden können.
F-SW-10.2	Der Ist-Kurs muss überwacht werden können.
F-SW-10.3	Der Missionsfortschritt muss überwacht werden können.
F-SW-10.4	Die Geschwindigkeit muss überwacht werden können.
F-SW-10.5	Der verbleibende Kapazität der Batterie muss überwacht werden können.
F-SW-10.6	Die verbleibende Laufzeit der Batterie muss überwacht werden können.
F-SW-10.7	Die abgeschätzte Zeit für den Weg zurück zur Basis muss überwacht werden können.
F-SW-10.8	Die Aktivität des Bootes muss überwacht werden können.
F-SW-10.9	Die Wetterdaten am Ort des Bootes müssen überwacht werden können.
F-SW-10.10	Mögliche Fehlerzustände müssen überwacht werden können.
F-SW-10.11	Die Signalstärke der XBee-Verbindung muss überwacht werden können.
F-SW-11	Die Software soll die Positionen erkannter Hindernisse approximiert darstellen.
F-SW-12	Nach einer Kollision soll der Nutzer benachrichtigt werden.
F-SW-13	Das System soll eine Schnittstelle bereitstellen, die es dem Entwickler ermöglicht, ausgewählte Systemparameter zu verändern.

F-SW-14	Zu Testzwecken muss die Software über eine Möglichkeit verfügen, die Hardware des Bootes auf einem Rechner zu simulieren.

Kollisionserkennung

Die nachfolgenden Anforderungen wurden anhand der Seminarausarbeitungen, weiterer Recherchearbeit sowie aus den zuvor definierten Rahmenbedingungen abgeleitet. Sie geben Aufschluss darüber, welche Erwartungen an das Subsystem der Kollisionserkennung gestellt werden.

F-KE-1	Das System muss die Umgebung kontinuierlich beobachten.
F-KE-1.1	Das System muss den Bereich vor dem Boot beobachten.
F-KE-1.2	Das System muss die Bereiche neben dem Boot beobachten.
F-KE-1.3	Das System muss den Bereich hinter dem Boot beobachten.
F-KE-2	Das System muss Objekte erkennen können.
F-KE-3	Das System muss Objekte identifizieren können.
F-KE-3.1	Das System muss Seezeichen identifizieren können.
F-KE-3.2	Das System muss Hindernisse identifizieren können.
F-KE-3.2.1	Das System muss statische Hindernisse identifizieren können.
F-KE-3.2.2	Das System muss dynamische Hindernisse identifizieren können.
F-KE-4	Das System soll domänen-unspezifisch agieren.
F-KE-5	Das System muss objektspezifische Parameter erkennen.
F-KE-5.1	Das System muss die Größe von Objekten erkennen.
F-KE-5.2	Das System muss die Entfernung zu Objekten erkennen.
F-KE-5.3	Das System soll die Geschwindigkeit eines Objekts bestimmen.
F-KE-5.4	Das System soll den Kurs eines Objekts abschätzen.

F-KE-6	Das System muss abwägen, ob eine Kollision wahrscheinlich ist.
F-KE-7	Bei der Erkennung eines Seezeichens soll das System einen Hinweis an die Bahnplanung weiterleiten.
F-KE-8	Bei dem Aktivieren der Fernsteuerung muss die Kollisionserkennung deaktiviert werden.

Kollisionsverhütung

Der folgende Abschnitt enthält die Anforderungen an die Kollisionsvermeidung und die Regelung des Wasserfahrzeugs. Hierbei sind die Anforderungen in die Kategorien Kursberechnung/Bahnplanung, Erkennen von physikalischen Grenzfällen und Regelung/Mathematisches Modell gegliedert.

Kursberechnung/Bahnplanung Im Verlauf der Anforderungsanalyse wurden zehn funktionale Anforderungen hinsichtlich der Kursberechnung und Bahnplanung erhoben.

F-KV-1	Der berechnete Pfad muss valide sein.
F-KV-2	Der berechnete Pfad muss gespeichert werden.
F-KV-3	Nach einem Neustart des Systems muss der gespeicherte Pfad erneut aufgerufen werden können.
F-KV-4	Der Algorithmus für die Bahnplanung muss Hindernisse einbeziehen.
F-KV-4.1	Der Algorithmus muss statische Hindernisse einbeziehen.
F-KV-4.2	Der Algorithmus muss dynamische Hindernisse einbeziehen.
F-KV-5	Das Boot soll die Position halten, wenn durch den Bahnplanungsalgorithmus kein valider Pfad gefunden werden kann.
F-KV-6	Der Algorithmus für die Bahnplanung muss vollständig simuliert werden können.
F-KV-7	Das System muss den Kurs beibehalten, wenn es zu keiner Kollision kommen wird.
F-KV-8	Das System muss selbstständig kollisionsverhütende Maßnahmen einleiten können, wenn es zu einer Kollision kommen würde.

Erkennen von physikalischen Grenzfällen Die Erkennung physikalischer Grenzfälle beschreibt funktionale Anforderungen, die in erster Linie die Einflüsse von Umweltparametern auf das System und den Bahnplanungsalgorithmus abdeckt.

F-KV-9	Die Plausibilität des GPS-Signals soll anhand bereits vorhandener Sensoren überprüft werden.
F-KV-10	Der Bahnplanungsalgorithmus soll die Richtung und die Geschwindigkeit der Strömung berücksichtigen können.
F-KV-11	Der Bahnplanungsalgorithmus soll die Richtung und die Geschwindigkeit des Windes berücksichtigen können.

Regelung/Mathematisches Modell Das mathematische Modell bildet die funktionalen Eigenschaften des Systems ab. Die Regelung ist dafür zuständig, aus gegebenen Parametern einen Kurs zu erzeugen und die Stabilität des Systems zu gewährleisten.

F-KV-12	Das System muss über geeignete Regler (tbd) verfügen.
F-KV-13	Die jeweiligen Regler müssen über einen Parametersatz verfügen, der unter Berücksichtigung möglicher Störgrößen ein stabiles System garantiert.
F-KV-14	Das mathematische Schiffsmodell muss die physikalischen Eigenschaften des Bootes modellieren.

3.4.2 Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen beschreiben Erwartungen an die Leistung des Systems. Der folgende Abschnitt führt die für diese Klasse relevanten Anforderungssätze auf. Gegliedert werden sie – analog zu dem vorherigen Kapitel – in den Kategorien *Software*, *Kollisionserkennung* und *Kollisionsverhütung*. Weiterhin werden auch nicht-funktionale Anforderungen für den Bereich *Bootsaufbau und Seetauglichkeit* erhoben.

Software

Die nicht-funktionale Anforderungen an die Software wurden anhand der Rahmenbedingungen abgeleitet. Sie beschreiben, in welcher Programmiersprache gearbeitet und welches Buildsystem genutzt werden soll.

N-SW-1	Die Onboard-Software muss in der Programmiersprache Java geschrieben werden.
N-SW-2	Die Onshore-Software muss in der Programmiersprache Java geschrieben werden.
N-SW-3	Die Onboard- und Onshore-Software sollen mithilfe des Buildsystems Maven erzeugt werden.

Kollisionserkennung

Aus Recherchearbeit ähnlicher Projekte und den im Vorfeld definierten Rahmenbedingungen konnten weiterhin zahlreiche nicht-funktionale Anforderungen an die Kollisionserkennung erfasst werden.

N-KE-1	Das System muss in ausreichender Höhe angebracht werden.
N-KE-2	Das System soll auch bei schlechter Witterung funktionieren.
N-KE-3	Die Objekterkennung muss zuverlässig sein.
N-KE-4	Die Hindernisidentifizierung muss zuverlässig sein.
N-KE-5	Das System muss mit wechselnden Umweltparametern umgehen können.
N-KE-5.1	Das System muss mit leichtem Wellengang umgehen können.
N-KE-5.2	Das System muss vor bildverändernden Wassertropfen geschützt sein.
N-KE-5.3	Das System muss vor Wasser geschützt sein.
N-KE-6	Das System soll lokale Parameter verarbeiten können.
N-KE-6.1	Das System kann Eigenbewegungen ausgleichen können.
N-KE-6.2	Das System kann einen Bildstabilisator verwenden.

N-KE-6.3	Das System kann einen Vertikalausgleich vornehmen können.
N-KE-6.4	Das System kann einen Horizontalausgleich vornehmen können.
N-KE-7	Das System soll kollisionsgefährdete Objekte approximiert darstellen können.
N-KE-7.1	Das System soll Formen erkennen und darstellen können.
N-KE-7.2	Das System soll Farben erkennen und darstellen können.
N-KE-8	Der Algorithmus muss rechtzeitig (tbd) Ergebnisse liefern.
N-KE-8.1	Der Prozessor muss für die Auswertung der Daten geeignet dimensioniert sein.
N-KE-8.2	Die Grafikeinheit muss für die Auswertung der Daten ausreichend performant sein.
N-KE-9	Der Algorithmus soll die Horizontlinie erkennen können.
N-KE-10	Der Algorithmus soll die relevante Seefläche erkennen können.
N-KE-11	Der Algorithmus kann das Ufer von der Seefläche abgrenzen.
N-KE-12	Der Algorithmus soll keine Kalibrierung erfordern.
N-KE-13	Der Algorithmus muss mit Wellenbewegungen (hohe Frequenz sich ändernder Pixel) umgehen können.

Kollisionsverhütung

Auch die vier nicht-funktionalen Anforderungen an die Kollisionsverhütung lassen sich in die bereits bekannten Kategorien Kursberechnung/Bahnplanung und Regelung/Mathematisches Modell einteilen.

Kursberechnung/Bahnplanung Die Anforderungen an die Kursberechnung/Bahnplanung beziehen sich auf die gewünschten Eigenschaften eines für ein gegebenes Hindernisfeld berechneten, kollisionsfreien Pfades.

N-KV-1	Das Boot muss einen ausreichenden Mindestabstand zu einem Hindernis einhalten.

N-KV-2	Der vom System berechnete Weg soll möglichst effizient (tbd) sein.

Regelung/Mathematisches Modell Zur Abbildung der Erwartungen an die Qualität der Regelung und das mathematische Modell wurden zwei weitere nicht-funktionale Anforderungen abgeleitet.

N-KV-3	Der ausgeregelte Zustand soll schnell (tbd) erreicht werden.
N-KV-4	Das mathematische Schiffsmodell muss die physikalischen Eigenschaften des Bootes mit einer hohen Genauigkeit (tbd) abbilden.

Bootsaufbau und Seetauglichkeit

Weiterhin wurden neun nicht-funktionale Anforderungen an den Aufbau des Bootes sowie die Seetauglichkeit erhoben. Sie umfassen vor allem Umweltbedingungen, denen das zu realisierende System standhalten muss.

N-BS-1	Das Boot muss vor eindringendem Wasser geschützt sein.
N-BS-2	Das Boot muss kleinere Wellen (tbd) unbeschadet überstehen.
N-BS-3	Das Boot soll leichte Strömungen (tbd) bewältigen können.
N-BS-4	Das Boot soll vor Regen geschützt sein.
N-BS-5	Das Boot soll für andere Verkehrsteilnehmer deutlich wahrzunehmen sein.
N-BS-6	Das Boot muss einfach ins Wasser gelassen werden können.
N-BS-7	Das Boot muss einfach aus dem Wasser geholt werden können.
N-BS-8	Der Kompass soll automatisch kalibriert werden.
N-BS-9	Ein Entwickler muss jederzeit die Möglichkeit besitzen manuell über eine Fernbedienung die Steuerung des Bootes zu übernehmen.

Kapitel 4

Aktuell eingesetztes System

In diesem Kapitel wird aufgezeigt, wie das aktuelle System aufgebaut ist und funktioniert, um im späteren Verlauf dieser Arbeit alle von uns getätigten Veränderungen zu verdeutlichen. Dazu gehen wir zunächst auf die aktuelle Software ein. Für die Onshore- und Onboard-Software weisen wir im gleichen Zug auch auf die auftretenden Probleme hin. Im nächsten Schritt wird der allgemeine Bootsaufbau beschrieben. Dazu haben wir die Handbücher der Vorgängergruppe herangezogen und Trockentests durchgeführt. Aus den Ergebnissen machten sich weitere Probleme und Eigenheiten erkennbar, welche ebenfalls beschrieben werden.

4.1 Ist-Zustand der Software

Der grundlegende Aufbau und die Funktionsweise der Softwarekomponenten wurden bereits in der Seminararbeit zur Software (siehe Anhang A.1) erläutert, deswegen wird auf eine erneute Erklärung an dieser Stelle verzichtet. Stattdessen folgen weitere Erkenntnisse, die während weiterer Tests am System festgestellt wurden.

Die initiale Inbetriebnahme der Softwarekomponenten ist mit einem erhöhten Arbeitsaufwand verbunden, da die vorhandenen nativen Bibliotheken nachträglich eingebunden werden müssen.

Onshore-Software

Die Onshore-Software weist Probleme bei der Darstellung des aktuellen Zustands des Bootes auf. So ist nicht ersichtlich, ob eine aktive Kommunikationsverbindung besteht, da unabhängig vom Verbindungszustand ein Boot auf der Karte dargestellt wird. Bei der Informationsübersicht der Bootsparameter fehlt der aktuelle Zustand der Komponenten. Es gibt keinen Überblick darüber, ob die GPS-Antenne oder das IMU-Board angeschlossen sind. Für

nicht angeschlossenen Komponenten werden Standard-Werte angezeigt, dies suggeriert dem Nutzer die Funktionsfähigkeit der Komponenten.

Der Verbindungsaufbau zur Plattform ist ohne Änderungen am Quelltext nur auf Windows Rechnern ausführbar, da der Verbindungsaufbau ausschließlich möglich ist, wenn die XBee-Antenne am COM-Port 5 angeschlossen ist. Dadurch ist eine Inbetriebnahme auf anderen Betriebssystemen nicht möglich.

Das Speichern und Laden von Mission ist voll funktionsfähig. Jedoch wird kein unübersichtliches Dateiformat verwendet, das eine schnelle Identifikation der Missionen ermöglicht. Der Dialog zum Speichern von Missionen als ein Dialog zum Öffnen von Dateien implementiert und umgekehrt.

Es hat sich gezeigt, dass das parallele Ausführen der Threads unter bestimmten Umständen zu Deadlocks führen kann. Dies führt zu einem Ausfall der Kommunikation zur Plattform.

Onboard-Software

Die Onboard-Software wird auf dem Raspberry Pi des Wasserfahrzeugs ausgeführt. Neben dem Bereitstellen einer Kommunikationsverbindung zur Onshore-Software ist die Onboard-Software für die Steuerung und das Auslesen sämtlicher Sensoren der Plattform verantwortlich.

Die Software setzt eine installierte Version des Betriebssystems des Linuxderivats Raspbian voraus. Derzeit kann auf den Raspberry Pi per SSH zugegriffen werden, indem per Laptop eine Verbindung zum Netzwerk „MOPS“ hergestellt wird. Alternativ kann eine Verbindung per Kabel gewählt werden. Die verwendete IP des Rechners ist `192.168.1.1`. Mit dem Benutzernamen „pi“ und dem Kennwort „raspberry“ kann eine Verbindung hergestellt. Aus Sicherheitsgründen empfiehlt es sich die Zugangsdaten zu ändern. Darüber hinaus sind von den vorangegangen Projektgruppen Einstellungen am Betriebssystem, die beispielsweise das Netzwerk betreffen, vorgenommen worden, sodass schnellstmöglich eine Image-Datei der SD-Karte des Raspberry Pi erstellt werden sollte. Dies reduziert den Konfigurationsarbeiten nach einem Ausfall der SD-Karte. Die Dateistruktur, die alle für die Benutzung des Wasserfahrzeugs nötigen Scripte und Programme enthält, ist in Abb. 4.1 dargestellt.

Das Verzeichnis `install` enthält im Wesentlichen Informationen zum Einrichten des Netzwerk und des WLAN-Sticks.

Der Ordner `bin` enthält Skripte zum Starten und Stoppen der Onboard-Software. Diese können zum Beispiel nach einem Absturz der Software oder einem Einspielen eines Updates hilfreich sein. Da die Onboard-Software diverse Informationen protokolliert, wurde das Script `showlastLog.sh` geschrieben. Dieses gibt in der Kommandozeile einen Ausschnitt der letzten Log-Nachrichten aus. Dies ist vor allem zum Überprüfen, ob eine XBee-Verbindung hergestellt wurde oder ob ein GPS-Signal vorhanden ist, sinnvoll. Außerdem sind die erzeugten Protokolle die einzige Information über das laufenden Pro-

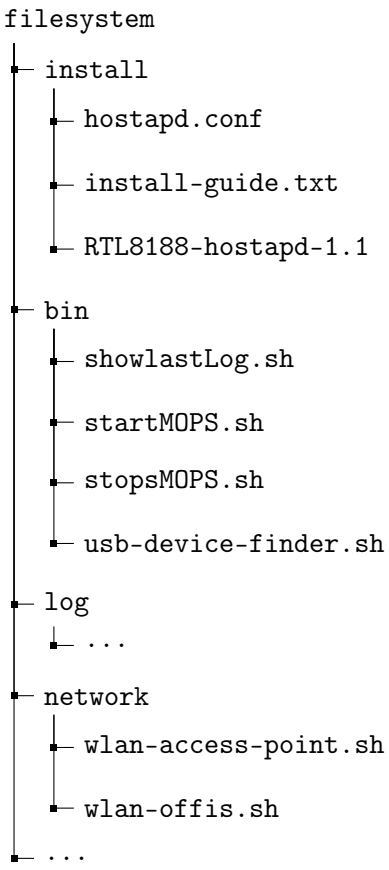


Abbildung 4.1: Struktur des *Raspberry Pi*-Dateisystems

gramm. Ein Nachteil zeigte sich darin, dass nicht alle Protokolle per XBee an die Onshore-Software übertragen werden. Beispielsweise wird nicht der letzte GPS-Kontakt mitgesendet. Das Script `usb-device-finder` gibt in der Kommandozeile eine Übersicht über alle angeschlossenen USB-Geräte aus. Da ein Großteil der Geräte per USB verbunden sind, kann das Skript für einen schnellen Überblick nützlich sein.

Der Verzeichnis `log` enthält alle aufgezeichneten Protokolldateien. Hierbei wird nach jedem Neustart der Onboard-Software ein neues Log-File angelegt. Die Namensgebung erfolgt hierbei mit einem Zeitstempel und einer dreistelligen Nummer.

Der Ordner `network` enthält zwei Skripte. Das Erste dient zum Erstellen eines WLAN-Access-Points und das Zweite zum Verbinden mit dem OFFIS-WLAN im Projektgruppenraum.

4.2 Ist-Zustand des Bootsaufbaus

Dieser Ist-Zustand des Bootes soll einen Überblick darüber geben, wie das Boot aufgebaut ist, aus welchen Komponenten das Boot besteht und wie der aktuelle Stand ist. Hierfür wurden Daten aus den Dokumentationen der vorherigen Gruppen Mops ii und Mops iii entnommen.

Hardware-Komponenten

- Raspberry Pi
- Leistungselektronik: Sabertooth
- Arduino Motor
- GPS: GM-65-Empfänger mit Adopt SkytraQ Venus8 Chipsatz
- Kompass: 9DoF Razor IMU-Board

4.2.1 Betrachtung des Nutzlastmanagements

In diesem Abschnitt wird der Aufbau des Bootes betrachtet. Außerdem wird geprüft, welche Hardware vorhanden ist und wo diese vorzufinden ist.

Heckkonstruktion

Die folgenden Daten sind aus der Dokumentation von Mops ii entnommen und wurden von uns mit den aktuellen Stand abgeglichen. Die Heckkonstruktion besteht aus Aluminium-Stangen die über Winkeln miteinander verbunden sind, die Maße betragen $965 \times 1000 \times 300$ ($L \times B \times H$ in mm). Sie wurde so konstruiert, dass in der Mitte drei wasserdichte Otterboxen der Größe $420 \times 245 \times 270$ ($L \times B \times H$ in mm) Platz finden. Eine Stange, die quer über den Otterboxen angebracht ist, sorgt dafür, dass die Boxen bei starken Wellengang oder Ähnlichem nicht über Bord gehen können. Die Stange selbst lässt sich über eine Schraube lösen, wodurch ein Erreichen der Boxen problemlos möglich ist. Innerhalb des Rahmens werden die Boxen von vier Aluminiumstangen getrennt. Zwischen den Trennstangen befindet sich ein Freiraum, indem die Schwimmkörper mittels Gürtelschnallen angebracht werden können.

Bugkonstruktion

Auch die Bugkonstruktion besteht aus einem Alumiumprofil mit Winkeln und Gelenken. Die Maße des Rahmens betragen $495 \times 715 \times 255$ ($L \times B \times H$ in mm). Hier sind die Akkumulatoren untergebracht. Auch die Batterieboxen werden durch eine Querstange gesichert. Das Gerüst ist oben und unten, mittels Schrauben, mit der Heckkonstruktion befestigt.

Da der Motorcontroller, die Spule der Not-Aus-Schaltung und die große Anzahl an Kabeln starke Magnetfelder erzeugen, sind diese in separaten Otterboxen untergebracht. Dies ist notwendig, um die Genauigkeit des Sensorboards und des GPS-Moduls nicht zu beeinträchtigen.

Weitere Komponenten

Für das Abdecken des Bootes wird eine Abdeckplane verwendet. Sie wird über ein Seil und ein Spanngurt am Boot befestigt. Die Montage der Abdeckplane ist zeitintensiv, beachtet außerdem nicht den Platzbedarf des Antennenkabels und schützt nicht vor plötzlich eintretenden Regenschauern. Die Otterboxen sind farblich markiert und enthalten folgende Komponenten:

- **Box 1 (rot):** Mikrocontroller und Relais stellen die Leistungselektronik dar. Die Stromversorgung ist über den Akku sicher gestellt und kann über den Zug-Not-Aus-Schalter unterbrochen werden, welcher aber momentan noch falsch implementiert ist. Die Komponenten sind mit einfachem Klebeband in der Box verklebt.
- **Box 2 (gelb):** Hier befinden sich Arduino Uno, Arduino Cape, ein Sensorboard, das via USB an das Raspberry Pi angeschlossen ist, sowie der RC-Empfänger für die Fernsteuerung. Die Komponenten sind ebenfalls mit einfachem Klebeband in der Box verklebt.
- **Box 3 (grün):** In der grünen Box befindet sich das entnahmefähige Raspberry Pi, an dem das XBee-Modul, Das GPS-Modul und ein USB-Hub angebracht sind. Die GPS-Antenne befindet sich direkt am Boot.

Validierungsliste Die Validierungsliste zeigt auf, welche Komponenten für die Inbetriebnahme des Bootes vorhanden sein sollten.

- **Bugsegment:** 1× obere Abdeckung, 1× unteres Gehäuse, 4× Gummihalterungen, 1× Batteriesteckverbindung mit zwei Klemmen, 1× Batterie 12 V.
- **Hecksegment:** 3× Otterboxen (rot, gelb, grün) mit den vorher aufgeführten Komponenten
- **Motoranbringung:** 2× Rhino-VX-34, 1× Stabilisierungsprofil, 2× Schiffsschrauben, 1× Schlüssel für die Schiffsschrauben
- **Verkabelung für die Boxen:** 2× M6 Sechskantschraube, 2× Nutenstein N6 mit Federkugel, stahlverzinkt, × Stromkabel
- **Auftriebskörper:** 4× Befestigungsgurte, 2× Auftriebskörper)
- **Plane:** 1× Plane, 1× Seil, 1× Spannhaken

4.2.2 Validierung der Handbücher

Folgende Schritte müssen bei der Inbetriebnahme durchgeführt werden. Falls notwendig, sollte für den grundlegenden Aufbau der Plattform die Anleitung in der Dokumentation der Gruppe Mops ii in Kapitel C.2 verwendet werden.

Batterie Der Ladezustand der Batterie kann nur durch ein Multimeter ermittelt werden, da die aktuelle Batterie über keine Statusanzeige verfügt.

Betriebsbereitschaft herstellen Eine Anleitung für korrekte Einstellungen und Steckverbindungen ist in der Dokumentation der Gruppe Mops iii zu finden in Kapitel C.3.

4.2.3 Ergebnisse des Trockentests

Ein Trockentest konnte erfolgreich durchgeführt werden. Wichtige Erkenntnisse sind:

- Die Motoren funktionieren nur mit gezogenem Kill-Switch! Laut Anleitung von Mops iii sollte die Kappe des Not-Aus allerdings wieder aufgesteckt werden.
- Die Schalter der Fernbedienung müssen beim Start alle nach hinten gestellt sein, andernfalls erscheint auf der Fernbedienung die Meldung „Switch-Error“
- Die Kontroll-LED leuchtet grün bei Bereitschaft und blinkt grün-blau, wenn eine Mission eintrifft
- Es muss ebenfalls darauf geachtet werden, dass die LEDs der einzelnen Hardware-Komponenten leuchten, um die Funktionsfähigkeit sicherzustellen.
- Ansonsten kann nach Anleitung von Mops iii verfahren werden

Kapitel 5

Theoretische Grundlagen

Der Schwerpunkt der theoretischen Grundlagen liegt auf den Themengebieten Regelung und Bildverarbeitung. Der Abschnitt Regelung umfasst eine Einführung in den PID- und Fuzzy-Regler. Das Kapitel Bildverarbeitung gibt eine Einführung in verschiedene Themen, die für die kamerabasierte Kollisionserkennung notwendig sind. Hierzu zählen beispielsweise die Kamerakalibrierung oder der Schwellwertfilter.

5.1 Regelungssystem

Ein Regelungssystem besteht aus miteinander verbundenen Komponenten. Diese Komponenten bilden eine Systemkonfiguration, die ein erwünschtes Verhalten, die Systemantwort, liefern soll. Auf Basis der linearen Systeme besteht für die Komponenten eines Systems eine Ursache-Wirkung-Beziehung.

Eine zu regelnde Komponente oder ein Prozess kann als ein Block dargestellt werden (siehe Abb. 5.1). Die Ursache-Wirkung-Beziehung des Prozesses wurde bei der Beziehung zwischen Eingang und Ausgang dieses Block repräsentiert. Diese Beziehung wird auch Übertragungsfunktion genannt. Gibt es keine Rückkopplung der Ausgangsgröße, ist es eine Steuerung [48, S. 23].

Abbildung 5.1: Kenngrößen eines Prozesses

Im Unterschied zu einem Steuerungssystem benutzt die Regelung immer das Ist-Ausgangssignals, das mit dem Soll-Ausgangssignals durch Rückfüh-

rung im Vergleich steht (siehe Abbildung 5.2). Dieses zurückgeführte Ausgangssignal wird als Rückkopplungssignal bezeichnet. In einer Regelung sollte ein vorgeschriebenes Verhältnis zwischen beiden Systemgrößen bestehen, damit ihre Funktionen miteinander verglichen werden können [48, S. 24].

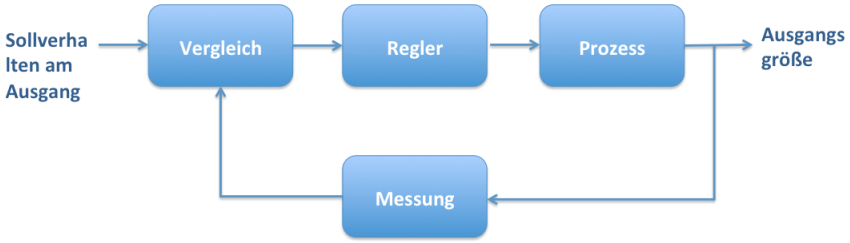


Abbildung 5.2: Schematische Darstellung eines Regelkreises

5.1.1 PID-Regelung

Der PID-Regler ist die häufigste Form von Feedback und dadurch werden viele industrielle Prozesse kontrolliert.

Der PID-Regler ist schon seit den 1940er Jahren ein wesentliches Element der Prozesskontrolle. Bei der Regelung von Prozessen sind heute mehr als 95 Prozent der Regelkreise vom PID-Typ. PID-Regler befinden sich heute in allen Bereichen wo Steuerelemente verwendet werden und werden oft mit Logik, sequenziellen Funktionen, Selektoren und einfachen Funktionsblöcken kombiniert, um komplizierte Automatisierungssysteme für Energieerzeugung, Transport und Fertigung zu konstruieren [34].

Der PID-Regler liefert einen proportionalen Term (P), einen integrierenden Term (I) und einen differenzierenden Term (D). Die Übertragungsfunktion im Laplace-Bereich lautet wie folgt (vgl. [48]):

$$G_c(s) = K_P + \frac{K_I}{s} + K_D\ s \tag{5.1}$$

Im Zeitbereich sind die Ausgangsgrößen gleich:

$$u(t) = K_P\ e(t) + K_I \int e(t)\ dt + K_D\ \frac{d}{dt}\ e(t) \tag{5.2}$$

Das Regelverhalten des PID-Reglers ist geeignet für Strecken mit großen Energiespeichern, die so weit wie möglich schnell und ohne bleibende Regelabweichung geregelt werden müssen. Sprich der Sollwert der Ausgangsgröße soll möglichst schnell mit möglichst keiner Abweichung erreicht werden.

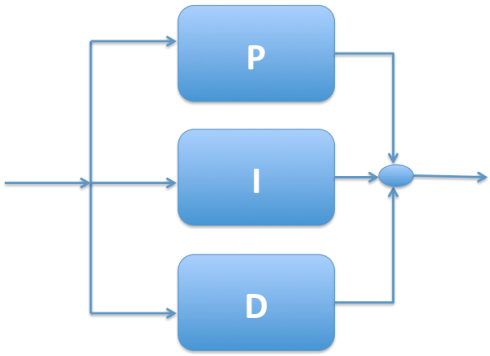


Abbildung 5.3: Grundprinzip eines PID-Reglers

Der PID-Regler ist der vollständigste Controller und am häufigsten genutzt, da er eine schnelle Antwort und ein Steuersignal bietet, um die tendenzielle Stabilität des Systems und eine minimale Steady-state Fehler zu erreichen [48, S. 526].

5.1.2 Fuzzy-Regelung

Die Fuzzy-Regelung stellt eine formale Methodik zur Darstellung, zur Bearbeitung und zur Umsetzung eines menschlichen heuristischen Wissens dar, um ein System zu steuern [109, S. 10]. Die Fuzzy-Regelung kommt aus der Fuzzy-Logik, die im Gegensatz zur klassischen Booleschen Logik mit nur zwei möglichen Wahrheitswerten realisiert wird. Dabei enthält sie nicht nur 0 und 1 als Extremfall der Wahrheit, sondern umfasst sie auch verschiedene Zustände der Wahrheit dazwischen.

Bei der Verwendung der Fuzzy-Logik in der Regelungstechnik wurde die Fuzzy-Regelung gebildet, die manche Probleme der Regelungstechnik lösen kann [113, S. 329].

Der Fuzzy-Regler besitzt vier Hauptkomponenten:

1. Die „Regelbasis“ enthält die Kenntnisse in Form von einer Reihe von Regeln, um festzulegen wie man das System am besten regelt.

2. Der Interfacemechanismus bewertet welche Steuerregeln zum aktuellen Zeitpunkt relevant sind und entscheidet was die Eingabe in der Anlage sein sollte.

3. Die Fuzzifizierung-Schnittstelle ändert die Eingänge, sodass sie interpretiert und mit den Regeln in der Regelbasis verglichen werden können.

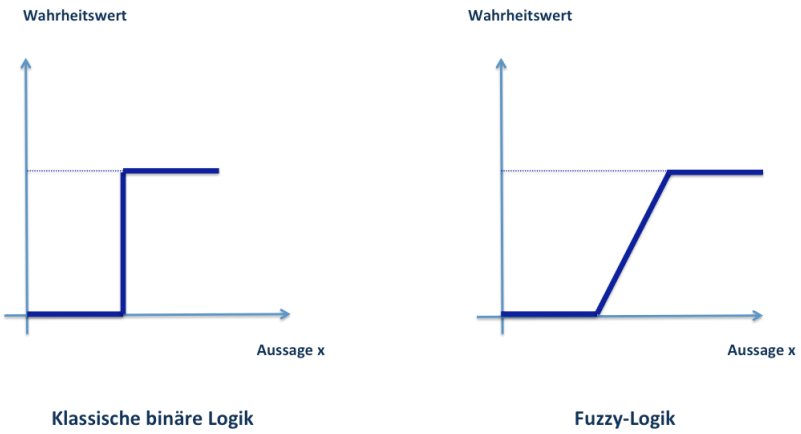


Abbildung 5.4: Vergleich von binärer und Fuzzy-Logik

4. Die Defuzzifizierung-Schnittstelle wandelt die Schlussfolgerungen, die durch den Interfacemechanismus erreicht werden, in die Eingänge der Anlage.

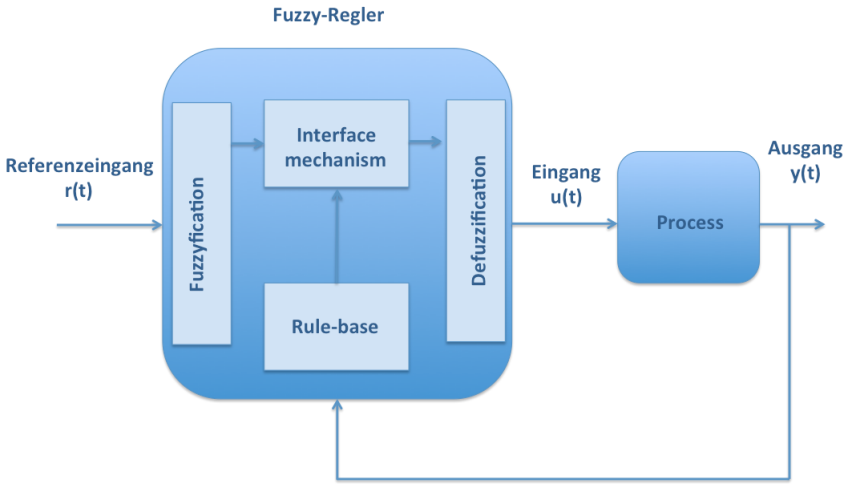


Abbildung 5.5: Grundprinzip eines Fuzzy-Reglers

Grundsätzlich sollte der Fuzzy-Regler als künstlicher Entscheider gesehen werden, der in einem geschlossenen System in Echtzeit arbeitet. Er sammelt die Ausgangsdaten $y(t)$, vergleicht sie mit dem Referenzeingang $r(t)$ und entscheidet, was für die Eingabe in der Anlage als $u(t)$ dargestellt werden sollte, um sicherzustellen, dass die Performance-Ziele erreicht werden [109, S. 11].

5.2 Bildverarbeitung

Die kamerabasierte Kollisionserkennung basiert auf dem Anwenden verschiedener Algorithmen auf Bilddaten. Hierfür werden im folgenden einige notwendige Algorithmen und Filter vorgestellt, die innerhalb der Kollisionserkennung verwendet werden.

5.2.1 Kamerakalibrierung

Für verschiedene Zwecke ist die Auswertung von Kamerabildern von besonderem Interesse. Im Rahmen der Projektgruppe werden beispielsweise Kamerabilder verwendet, um auf diesen Objekte, die sich vor dem Wasserfahrzeug befinden, zu detektieren. Unabhängig von der eingesetzten Kamera werden zur Generierung der Bilddaten Linsensysteme in Verbindung mit einem Bildsensor eingesetzt. Durch Variation von Linse und Sensor lassen sich Weitwinkel-, Makro-, Mikro oder Teleaufnahmen erzeugen. Durch die Verwendeten Linsen kommt es beim resultierenden Bild häufig zu Verzerrungseffekten. Diese sind beim Verwenden von Weitwinkelobjektiven besonders ausgeprägt. Um die Aufnahmen für die Objekterkennung oder eine Horizonterkennung nutzbar zu machen, ist es empfehlenswert die Verzerrung mit Hilfe einer Kamerakalibrierung zu reduzieren. Der nachfolgende Abschnitt beschreibt die von der Projektgruppe durchgeführte Kalibrierung und zeigt einige Vorher-/Nachheraufnahmen zur Veranschaulichung.

Das in Abbildung 5.6(a) dargestellt Bild zeigt, dass der Horizont der Aufnahme eine deutliche Krümmung aufweist. Dies ist darauf zurückzuführen, dass für die Aufnahmen eine Kamera mit einem Weitwinkelobjektiv eingesetzt wurde. Das Weitwinkelobjektiv ermöglicht zwar einen deutlichen größeren Sichtwinkel als ein Standardobjektiv, sorgt aber gleichzeit dafür, dass die Bilder eine gewissen Verzerrung aufweisen. Die Verzerrung kann mit Hilfe einer Kamerakalibrierung korrigiert werden, sodass das in Abb. 5.6(b) dargestellte Bild berechnet werden kann. Hierauf ist jetzt ein exakt gerade Horizont zu erkennen.

Durchgeführt wird die Kamerakalibrierung mit dem in [82] beschriebenen Verfahren. Das Verfahren nutzt verschiedene in OpenCV implementierte Funktionen, die ein schnelles Kalibrieren der Kamera ermöglichen. Die Grundidee hierbei ist es ein Kalibrierungsobjekt einzusetzen. Dieses zeigt häufig ein Schachbrettmuster, dessen Kachelgröße bekannt ist. Abbildung 5.7 zeigt

(a) Originalbild der GoPro

(b) Entzerrtes Bild

Abbildung 5.6: Beispielhafte Korrektur eines verzerrten GoPro-Bildes

eine Beispielaufnahme. In dieser ist Verzerrung sehr deutlich innerhalb des Musters oder auch an den Schränken und Wänden des Raumes zu erkennen.

Mit Hilfe von OpenCV werden in einem nächsten Schritt die Eckpunkte der Kacheln aus dem Bild extrahiert. Da die Maße des Kalibrierungsobjektes bekannt sind, kann hieraus berechnet werden, wie die Pixel des Bildes verschoben werden müssen, um die Verzerrung zu korrigieren. Auch hierfür bietet OpenCV Funktionen, die einem die Berechnung und das Remapping der Pixel abnehmen. Eine tiefere Beschreibung des Kalibrierungsprozesses ist an dieser Stelle nicht nötig, diese kann in [82] nachgelesen werden. Für die Projektgruppe war es ausreichend die zur Verfügung gestellten Funktionen zu verwenden und den sehr theoretischen Berechnungsprozess nicht im Detail zu betrachten.

Gespeichert werden die mit Hilfe von OpenCV berechneten Kalibrierungsparameter automatisch in einer XML-Datei, sodass diese in späteren Programmen eingelesen und benutzt werden können.

Abbildung 5.7: Beispielaufnahme eines Kalibrierungsobjektes

5.2.2 Morphologische Filter

Morphologische Filteroperationen sind eng mit der mathematischen Morphologie verknüpft. Da die mathematischen Aspekte für die Anwendung im Rahmen des Projektes uninteressant sind, wird auf eine detaillierte Beschreibung an dieser Stelle verzichtet. Einen Einblick geben beispielsweise [108, S. 176 ff.] und [43, S. 171 ff.]. Der folgende Abschnitt erläutert daher anhand von Beispielen eine Filter zur horizontalen Linienerkennung und die Dilatation.

Horizontale Linienerkennung

Morphologische Filter betrachten in der Regel Nachbarschaftsbeziehungen von ausgewählten Pixeln, um dann nach festgelegte Regeln den Wert für das Pixel im Ausgangsbild zu bestimmen. Um horizontale Linien in einem Bild zu erkennen, wird daher Schritt für Schritt jedes Pixel des Bilder durchlaufen und analysiert. Abbildung 5.8 zeigt die ausgewählt Nachbarschaft, die für die Erkennung von horizontalen Linien interessant ist. Das mit „Origin“ gekennzeichnete Pixel entspricht dem aktuell betrachtete Pixel, für welches ein Ausgangswert bestimmt werden soll. Das Beispiel bezieht sich auf das in Abb. 5.9(a) dargestellte Binärbild, sodass die Pixel nur die Werte 0 oder 1 annehmen können.

Die Entscheidung, welchen Ausgangswert das entsprechende Pixel annimmt, ist wie folgt definiert: Wir betrachten sechs Nachbarn um das Pixel und wählen als Ausgangswert den in der Nachbarschaft am häufigsten auftretenden Wert. Der dargestellt Vektor von Nachbarpixeln zeigt, dass die sechs

Nachbarn alle den Wert 1 ausweisen, daher wird der Pixelwert im Ausgangswert ebenfalls den Wert 1 haben.

Abbildung 5.8: Vektor zur Betrachtung der Nachbar-Pixel (aus [83])

Die oben beschriebene Operation wird jetzt auf alle Pixel des Bilder angewendet, sodass das in Abb. 5.9(b) dargestellt Bild entsteht. Es ist deutlich zu sehen, dass die Notenlinie aus dem Bild extrahiert werden konnten. Die hier beschriebene Idee kann in Kapitel 7 zur Horizonterkennung genutzt werden [83].

Das hier vorgestellte Verfahren lässt sich außerdem sehr einfach in OpenCV verwenden. Die verfügbaren Funktionen machen ein flexibles einsetzten des Nachbarschaftsvektors möglich, sodass mit wenig Aufwand verschiedenste Einstellungen des Verfahren ausprobiert werden können. In [83] wird eine Einführung in die Verwendung der entsprechenden Funktionen gegeben.

(a) Originalbild (aus [83])

(b) Bild mit angewandtem horizontalen morphologischen Filter (aus [83])

Abbildung 5.9: Beispiel eines horizontalen morphologischen Filters

Dilatation

Bei der Dilatation werden spezifische Bildbereiche mithilfe eines Filterkerns ausgedehnt. Hierbei wird der Filterkern, der wahlweise kreisförmig oder rechteckig sein kann, über das Bild geführt. Der Filterkern benötigt einen Ankerpunkt, der im Regelfall im Zentrum liegt, an dem der Filter Pixel für Pixel über das Bild geschoben wird. Für jeden Pixel wird der maximale Pixelwert berechnet, der vom Filterkern überdeckt wird. Im Anschluss daran wird der Ankerpixel auf den zuvor errechneten Pixelwert gesetzt.

Mithilfe von OpenCV lässt sich die Dilatation auf ein eingelesenes Bild anwenden [89]. Dabei muss ein Eingabebild, ein Ausgabeort, die Größe des Filterkerns, der Ankerpunkt sowie der Umgang mit Randpixeln angegeben werden.

Abbildung 5.10 zeigt die Anwendung der Dilatation mit OpenCV auf einem Binärbild. Dadurch, dass der Ankerpixel durch den höchsten Pixelwert im näheren Umkreis ersetzt wird, werden durch die weißen Bereiche, die im Binärbild durch eine 1 gekennzeichnet sind, ausgedehnt. Auf welche Weise die Dilatation in der Projektgruppe umgesetzt wird, lässt sich in Kapitel 7 nachlesen.

(a) Binärbild ohne Dilatation (b) Resultat der Dilatation

Abbildung 5.10: Beispiel zur Verwendung der Dilatation (aus [**?**])

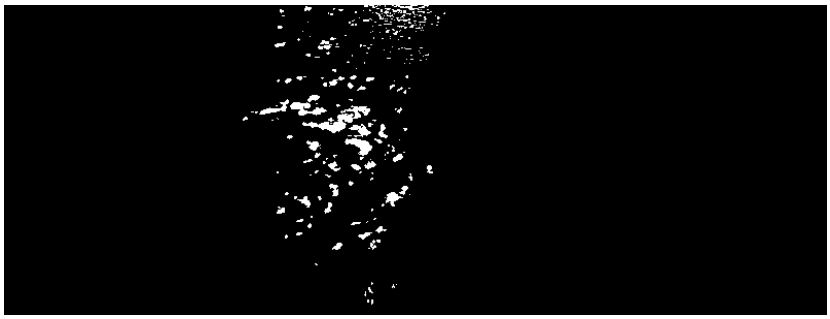
5.2.3 Schwellwertfilter

Der Schwellwertfilter zerlegt ein gegebenes Grauwertbild (Farbbilder sind ebenfalls möglich) in ein Binärbild. Hierbei wird für jedes Pixel a_{ij} des Bildes folgende Überprüfung durchgeführt: Ist der Pixelwert größer als ein gewisser Schwellwert a_{th} oder kleiner. Entsprechend wird der Pixelwert für das Ausgangsbild auf 0 oder 1 gesetzt. Wird das beschriebene Verfahren auf alle Pixel angewendet, so resultiert das erwähnte Binärbild [43, S. 57 f.].

Abbildung 5.11 zeigt ein Beispiel zur Verwendung eines solchen Filters. Das in Abb. 7.36(b) dargestellt Bild wird zunächst in ein Grauwertbild konvertiert. Auffällig sind die durch Reflexion von Sonnenlicht deutlich sichtbaren hellen Bereiche im Bild. Diese lassen sich mit einem Schwellwertfilter einfach ermitteln. Die Pixel des Grauwertbildes erstrecken sich über einen Wertebereich von 0 bis 255 (255 ist weiß). Der Schwellwert kann in diesem Fall Beispielsweise auf den Wert 210 festgelegt werden. Jetzt können alle Pixel des Grauwertbildes mit dem Schwellwert verglichen werden und das Bild in 5.11(b) entsteht. Der Schwellwertfilter ist ein sehr nützliches Werkzeug zum

Entfernen der Reflexionen. Näheres hierzu kann in Kapitel 7 nachgelesen werden.

(a) Wasseroberfläche mit Reflexionen



(b) Resultat des Schwellwertfilters

Abbildung 5.11: Beispiel zur Verwendung von Schwellwertfiltern

Im allgemeinen lässt sich dieses Verfahren auch auf RGB-Bilder übertragen, dies war im Rahmen der Projektgruppe allerdings nicht notwendig. Eine Implementierung des Verfahrens ist in OpenCV verfügbar und über entsprechende Funktionen nutzbar. In [86] werden ausführliche Beispiele zur Verwendung genannt.

5.2.4 Masken

Das Ziel eines Maskenoperators besteht häufig darin, Bildausschnitte aus einer Aufnahme zu entfernen. Beispielsweise ist es im Rahmen der Projektgruppe interessant die in Abb. 7.36(b) zu sehende Reflexion des Sonnenlichts an der Wasseroberfläche zu reduzieren oder zu entfernen (näheres hierzu in Kapitel 7). Um dies zu realisieren ist zunächst das Erstellen eine Maske notwendig. Eine geeignete Maske ist das über den Schwellwert-Filter erzeugte Binärbild in Abb. 5.11(b). Die weißen Bereiche des Binärbilds bilden jetzt die im Originalbild zu maskierenden Bildteile. Mit Hilfe von OpenCV ist es

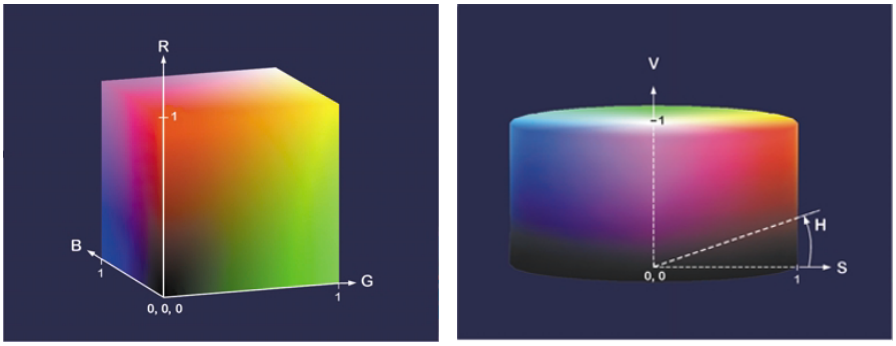
möglich die Pixel im Originalbild nach verschiedenen Methoden zu manipulieren. Beispielsweise könnte man die durchschnittliche Pixelfarbe der Randpixel wählen, um so die Reflexion aus dem Bild zu entfernen.

In OpenCV ist die „Inpainting-Methode“ für das oben beschriebene Verfahren geeignet. In dieser lassen sich verschiedene Verfahren und Einstellungen konfigurieren, sodass das Ergebnis auf den Anwendungsfall angepasst werden kann. Eine detailliere Beschreibung liefert [84].

5.2.5 Farbräume

Für das Verarbeiten von Bildmaterial ist es häufig notwendig den Farbraum des vorliegenden Bildes zu analysieren. Dieser Abschnitt erläutert die Unterschiede zwischen dem RGB-Farbraum und HSV-Farbraum und welche Bedeutung diese für die Umsetzung der Kollisionserkennung haben.

Ein Ansatz zur Erkennung von Objekten besteht darin Objekte anhand ihrer charakteristischen Farbe zu erkennen. Im RGB-Farbraum lassen sich nur die Werte für die Rot-, Grün- und Blau-Anteile des Bildes ableiten, wie in Abb. 5.12(a) zu sehen ist. Wichtige Informationen wie die Helligkeit und Sättigung der einzelnen Pixel sind in diesem Farbraum nicht ersichtlich. Allerdings ist es möglich das Bild auf den HSV-Farbraum abzubilden und diese Werte herauszurechnen. Abbildung 5.12(b) zeigt einen Zylinder, welcher diesen Farbraum beschreibt. Neben den Informationen zum Farbton, sind Sättigung und Helligkeit gut abzuleiten.



(a) RGB-Farbraum (b) HSV-Farbraum

Abbildung 5.12: Vergleich der RGB- und HSV-Farbräume (aus [99])

Der Nachteil bei der Verwendung des RGB-Farbraums ist, dass der Farbton wenig Aussagekraft besitzt, wenn die Farbsättigung unter einem bestimmten Wert liegt [112, S. 3]. Da HSV-Farbraum-Bilder die Helligkeit getrennt von Farbton und Sättigung betrachten, hat diese auch wenig Auswirkung auf die anderen beiden Kanäle, was die Verwendung der Farbwerte sinnvoller macht.

OpenCV stellt eine Funktion bereit, die Bilder aus dem RGB-Farbraum in den HSV-Farbraum-Farbraum transformiert [85].

5.2.6 Histogramme

Mithilfe von Histogrammen lassen sich Häufigkeitsverteilungen bestimmter Merkmale eines Datensatzes graphisch darstellen. Im Rahmen der Projektgruppe ist es angedacht Objekte mittels ihrer farblichen Eigenschaften von der Wasseroberfläche abzuheben. Mithilfe von OpenCV lassen sich Histogramme der Farbwerte eines Bildes oder von Teilbereiche dessen erstellen [90].

Abbildung 5.13 zeigt ein Histogramm, welches die Häufigkeitsverteilung der Farbwerte eines Bildes aus dem RGB-Farbraum aufzeigt. Dabei sind drei Graphen aufgezeigt, die stellvertretend für die Rot-, Grün- und Blau-Anteile stehen. Auf der x-Achse werden die Anteile der jeweiligen Farbe definiert, während die y-Achse die Häufigkeit dieser Farbanteile beschreibt.

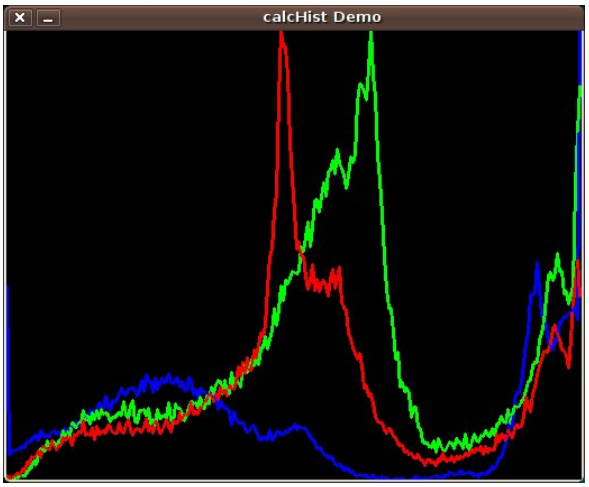


Abbildung 5.13: Histogramm eines Bildes im RGB-Farbraum (aus [90])

Kapitel 6

Konzept

In diesem Kapitel wird das gewählte Konzept zur Realisierung der im Vorfeld definierten Funktionalität erläutert. Ausgehend von den in Kapitel 3 erhobenen Anforderungen, einer Spezifikation der einbezogenen Systemkomponenten sowie einer ausführlichen Betrachtung funktionaler Sicherheitsaspekte erfolgt im Anschluss an eine systematische Verfeinerung der Systemarchitektur die vollständige Beschreibung des geplanten Systems.

6.1 Systemübersicht

Abbildung 6.1 gibt einen ersten Überblick über den geplanten Aufbau des zu realisierenden Gesamtsystems. In ihr sind neben den zentralen Komponenten der *Onboard-* und *Onshore-Systeme* auch verschiedene Fremdsysteme, wie beispielsweise ein GPS-Empfänger und Mensch-Maschine-Schnittstellen, sowie systemeigene Sensoren und Aktoren einbezogen.

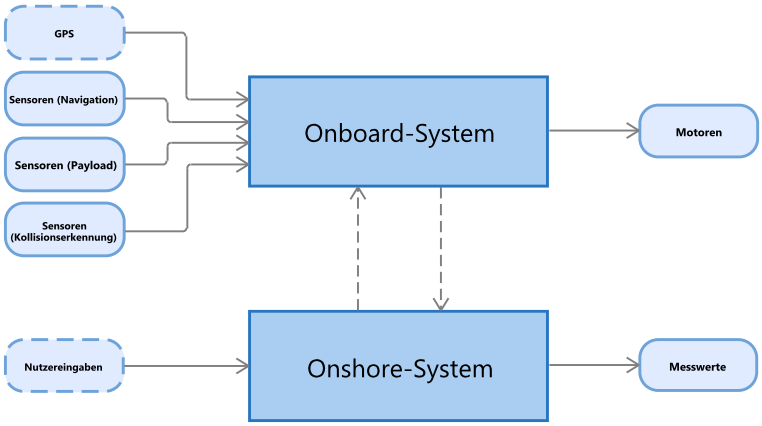


Abbildung 6.1: Übersicht über das Gesamtsystem

Das *Onboard-System* ist vollständig auf dem Boot integriert und bildet das zentrale Element zur Durchführung von Missionen. Für die Navigation verwendet das System das Fremdsystem *GPS* sowie verschiedene Sensoren, die die Lage und Ausrichtung der Plattform auf einem Gewässer berechnen.

Neben der Navigation zeichnet das Onboard-System die Messwerte der *Payload-Sensoren* auf. Als Payload-Sensoren werden ein Sensor zur Bestimmung des Salzgehalts und ein weiterer zur Messung der Trübheit eingesetzt.

Um dynamische Hindernisse zu detektieren, sind auf dem Wasserfahrzeug *Sensoren zur Kollisionserkennung* montiert. Diese werden ebenfalls durch das Onboard-System ausgewertet, um auf Hindernisse reagieren zu können. Auf Basis der eingebauten Navigationsinstrumente werden die Motoren zum Halten eines zuvor festgelegten Kurses gesteuert.

Das *Onshore-System* kommuniziert über eine drahtlose Verbindung mit dem Onboard-System. Die primäre Aufgabe des Onshore-Systems besteht in dem Verarbeiten der Nutzereingaben. Hierzu gehört die Planung von Missionen und das Starten oder Abbrechen einer Mission. Die entsprechenden Daten werden über einen Kommunikationskanal an das Onboard-System übertragen. Eine weitere Aufgabe dieses Systems ist die Auswertung und Darstellung der Messwerte der Payload-Sensoren.

Da das Onshore-System bzgl. der Architektur vorerst nicht geändert wird, beziehen sich die nachfolgenden Betrachtungen ausschließlich auf die Konzepte des Onboard-Systems.

6.2 Funktionale Sicherheit

Der Begriff der *funktionalen Sicherheit* bezeichnet den Teil der Sicherheit eines zu realisierenden Systems, der „eine zuverlässige und sicherheitsbezogene Funktion der (Sub-)Systeme und externer Einrichtungen sowohl im Normalbetrieb als auch bei Ausfällen und Fehlern garantiert“ [50, S. 6.4].

In diesem Abschnitt werden die für die funktionale Sicherheit relevanten Systembestandteile, mögliche Gefahreneinflüsse und ihre Risiken, die daraus abzuleitenden Sicherheitsziele und -anforderungen sowie die funktionalen und technischen Sicherheitskonzepte unseres Systems erläutert.

6.2.1 Relevante Systembestandteile

Um alle Randfälle in geeignetem Maße berücksichtigen zu können, ist für die weitere Betrachtung zunächst eine Abgrenzung systeminterner von -externen Komponenten sowie eine klare Definition der Schnittstellen notwendig.

Da das im vorherigen Abschnitt beschriebene Onshore-System keine sicherheitskritischen Funktionen implementiert und bereits über eine geeignete Fehlererkennung verfügt, beschränken wir uns im Folgenden auf die ausschließliche Betrachtung des Onboard-Systems. Um die Abhängigkeiten zwischen den realen (Sub-)Systemen korrekt abbilden zu können, wird die

Onshore-Software dabei als externes (Sensor-)System behandelt, das Eingabedaten für die im Fokus stehenden Komponenten bereitstellt.

Auch die für die Ansteuerung der Payload-Sensoren sowie dem Aufnehmen, Vorverarbeiten und Speichern von Messwerten vorgesehenen Komponenten werden aufgrund ihres zu vernachlässigenden Einflusses auf die sicherheitskritischen Bestandteile des Systems abstrahiert und bleiben im Verlauf der nachfolgenden Ausführungen weitgehend unbeachtet.

Einen kompositionellen Ansatz für die Analyse dieses Teilsystems verwendend, lassen sich die Eigenschaften des Items *Onboard-System* (I-1) somit systematisch aus den beiden untergeordneten Komponenten *Missionsplanung und -durchführung* (I-1.1) sowie *Kollisionserkennung und -verhütung* (I-1.2) herleiten, deren Definitionen sich wie folgt ergeben:

I-1.1 **Missionsplanung und -durchführung**
Das die funktionalen und nicht-funktionalen Anforderungen F-SW-7, F-SW-8, F-SW-10, F-SW-13, F-KV-1 bis F-KV-3, F-KV-9 bis F-KV-13, N-SW-1, N-SW-3, N-KV-2, N-KV-3 und N-BS-8 unter Berücksichtigung der Rahmenbedingungen R1 bis R4 realisierende Teilsystem.

Alle Komponenten des Onboard-Systems, die in die Entgegennahme der Missionsdaten, die Berechnung eines kollisionsfreien Pfades, die Steuerung der Motoren sowie die Bestimmung des Systemstatus einbezogen sind.

I-1.2 **Kollisionserkennung und -verhütung**
Das die funktionalen und nicht-funktionalen Anforderungen F-SW-11, F-SW-12, F-KE-1 bis F-KE-8, F-KV-1 bis F-KV-8, N-SW-1, N-SW-3, N-KE-1 bis N-KE-13, N-KV-1 und N-BS-9 unter Berücksichtigung der Rahmenbedingungen R1 bis R4 realisierende Teilsystem.

Alle Komponenten des Onboard-Systems, die in die Erkennung, Identifikation und Klassifikation möglicher Kollisionsrisiken sowie die Einleitung von Maßnahmen zur Minimierung des Gefahrenpotentials einbezogen sind.

Missionsplanung und -durchführung Die Architektur des Items *Missionsplanung und -durchführung* (I-1.1) ist in Abb. 6.2 graphisch dargestellt. Neben dem bereits beschriebenen *Onshore-System*, das die Missionsdaten bereitstellt, bezieht das zentrale Steuergerät einen *GPS-Empfänger* zur Positionsbestimmung sowie die systemeigene *Navigationssensorik* ein und berechnet aus den eingehenden Messwerten fortlaufend aktualisierte Motorwerte,

die an die *Antriebseinheit* weitergeleitet werden. Über den Aktor *Statusverwaltung* können zudem Informationen über den aktuellen Systemzustand an eine beliebige Anzahl externer Komponenten weitergeleitet werden.

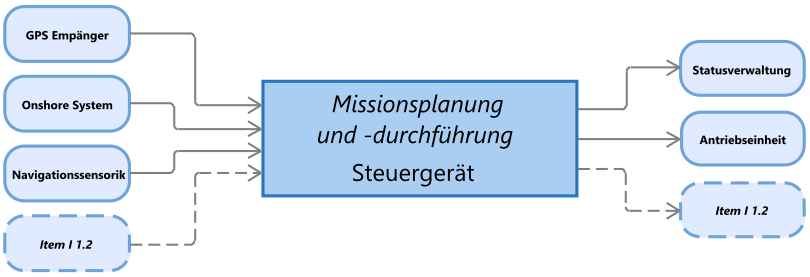


Abbildung 6.2: Struktur des Items *Missionsplanung und -durchführung*

Um während der autonomen Fahrt durch ein unbekanntes Gewässer sowohl ortsfeste als auch bewegliche Hindernisse in die Missionsplanung einbeziehen zu können, werden durch das Steuergerät Kollisionswarnungen des Items I-1.2 entgegengenommen. Da gleichwohl auch diese Komponente über mögliche Kursänderungen informiert werden muss, ist das entsprechende Subsystem zusätzlich auch als Aktor des Items I-1.1 aufgeführt.

Kollisionsvererkennung und -verhütung Analog zu der im vorherigen Abschnitt dargestellten Architektur wird die Struktur des Items I-1.2 in Abb. 6.3 ersichtlich. Eine zuverlässige Erkennung möglicher Kollisionsrisiken forcierend, nimmt das Steuergerät in einem ersten Schritt Informationen über erkannte Hindernisse der sogenannten *Abstandssensorik* entgegen.

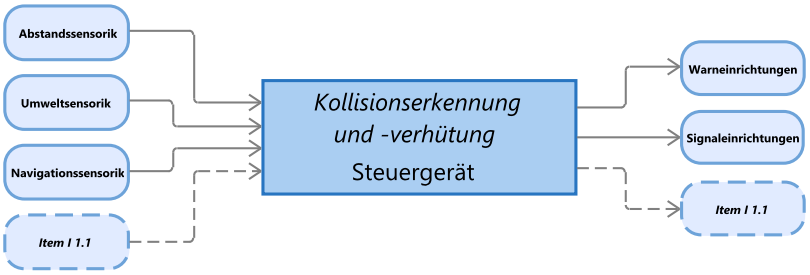


Abbildung 6.3: Struktur des Items *Kollisionserkennung und -verhütung*

Zur Validierung der eingehenden Werte verfügt das System zudem über eine geeignete *Umweltsensorik*, die eine Einschätzung der vorherrschenden Lichtverhältnisse sowie der Wetterbedingungen ermöglicht. Die systemeigene

Navigationssensorik erhöht die Genauigkeit der Berechnungen und bildet die Grundlage für eine präzise Verhaltensprädiktion dynamischer Hindernisse.

Während die *Warneinrichtungen* ein Subsystem bezeichnen, das andere Komponenten über eine drohende Kollision informiert, sind unter dem Begriff der *Signaleinrichtungen* technische Einrichtungen zur Ausgabe von optischen und akustischen Warnungen an die Umwelt zusammengefasst. Die obligatorische Nennung des Items I-1.1 als sensorisches und aktorisches Teilsystem erfolgt analog zu den im vorherigen Abschnitt angebrachten Ausführungen.

6.2.2 Gefährdungsanalyse und Risikoabschätzung

Vor dem Einleiten von Maßnahmen zur Gewährleistung der funktionalen Sicherheit ist es sinnvoll, sich genauer mit den möglichen Gefahreneinflüssen zu befassen, die unter Einsatzbedingungen auf das Gesamtsystem wirken. Das Ziel dieser Arbeitsphase ist daher die Identifikation, Bewertung und Kategorisierung potentieller Risiken, die Fehlfunktionen des spezifizierten Systems auslösen oder Folgefehler nach sich ziehen können.

Mögliche Gefahreneinflüsse

Im Rahmen der Gefährdungsanalyse unseres Projekts konnten eine Reihe von Gefahreneinflüssen (*Hazards*) identifiziert werden, die sich im Wesentlichen auf die nachfolgend aufgelisteten Kategorien zurückführen lassen:

Missionsplanung und -durchführung

H-1.1-1 **Beeinträchtigtes Onshore-System**
Die Funktionalität des Onshore-Systems kann für einen kurzen Zeitraum ($t < 2\,s$) nicht oder lediglich in einem eingeschränkten Umfang bereitgestellt werden.

H-1.1-2 **Ausfall des Onshore-Systems**
Das Onshore-System steht für einen längeren Zeitraum ($t \geq 2\,s$) nicht zur Verfügung oder ist durch einen Hardware-Defekt dauerhaft beschädigt.

H-1.1-3 **Fehlerhafte Onshore-Kommunikation**
Die Missionsdaten werden bei der Übertragung an das Steuergerät verfälscht oder können für einen kurzen Zeitraum ($t < 2\,s$) nicht weitergegeben werden.

H-1.1-4 **Ausfall der Onshore-Kommunikation**
Die Kommunikation zwischen dem Onshore-System und dem Steuergerät bricht für einen längeren Zeitraum ($t \geq 2\,s$) zusammen oder ist dauerhaft gestört.

H-1.1-5 **Störung des Steuergeräts**
Die durch das Steuergerät bereitgestellte Funktionalität steht für einen kurzen Zeitraum ($t < 2\,s$) nicht zur Verfügung oder ist in ihrer Qualität beeinträchtigt.

H-1.1-6 **Ausfall des Steuergeräts**
Das Steuergerät kann die spezifizierte Funktionalität für einen längeren Zeitraum ($t \geq 2\,s$) nicht bereitstellen oder ist durch einen Hardware-Defekt dauerhaft beschädigt.

Kollisionserkennung und -verhütung

H-1.2-1 **Falsch-Positiv-Erkennung**
Das System erkennt ein inexistentes Hindernis.

H-1.2-2 **Falsch-Negativ-Erkennung**
Ein zu erkennendes Hindernis wird nicht detektiert.

H-1.2-3 **Beeinträchtigte Abstandssensorik**
Die Abstandssensorik liefert für einen kurzen Zeitraum ($t < 2\,s$) offensichtlich falsche Werte oder wird durch Umwelteinflüsse beeinträchtigt.

H-1.2-4 **Ausfall der Abstandssensorik**
Die Abstandssensorik liefert für einen längeren Zeitraum ($t \geq 2\,s$) offensichtlich falsche Werte oder ist durch einen Hardware-Defekt gestört.

H-1.2-5 **Fehlerhafte Bus-Kommunikation**
Die Sensordaten werden bei der Übertragung an das Steuergerät verfälscht oder können für einen kurzen Zeitraum ($t < 2\,s$) nicht weitergegeben werden.

H-1.2-6 **Ausfall der Bus-Kommunikation**
Die Kommunikation zwischen der Sensoreinheit und dem Steuergerät bricht für einen längeren Zeitraum ($t \geq 2\,s$) zusammen oder ist dauerhaft gestört.

H-1.2-7 **Störung des Steuergeräts**
Die durch das Steuergerät bereitgestellte Funktionalität steht für einen kurzen Zeitraum ($t < 2\,s$) nicht zur Verfügung oder ist in ihrer Qualität beeinträchtigt.

H-1.2-8 **Ausfall des Steuergeräts**
Das Steuergerät kann die spezifizierte Funktionalität für einen längeren Zeitraum ($t \geq 2\,s$) nicht bereitstellen oder ist durch einen Hardware-Defekt dauerhaft beschädigt.

Risiken und Folgefehler

Für die im vorherigen Abschnitt identifizierten Gefahreneinflüsse lassen sich weiterhin ihre Ursachen sowie mögliche Folgefehler (*Risks*) identifizieren. Auf das Bestimmen der Eintrittswahrscheinlichkeiten muss an dieser Stelle verzichtet werden, da im Rahmen der Projektgruppe keine zuverlässigen Fehler- und Ausfallwahrscheinlichkeiten der verwendeten Komponenten vorliegen.

Aufgrund der Tatsache, dass sich das Risiko eines Gefahreneinflusses aus der „Multiplikation der Schadenshöhe mit der Eintrittswahrscheinlichkeit [...] summiert über die verschiedenen Gefährdungen“ [50, S. 6-27] ergibt, erfolgt in diesem Abschnitt lediglich die informelle Auflistung der denkbaren Auswirkungen auf das System und die Systemumgebung.

Obgleich die Sicherheitsanalyse in Hinblick auf die Genauigkeit und ihre Aussagekraft nicht mit denen realer Industrieprojekte vergleichbar ist, bilden die für die funktionale Sicherheit relevanten Erkenntnisse eine wichtige Grundlage für den Entwurf und die Realisierung des Gesamtsystems.

Missionsplanung und -durchführung

R-1.1-1 **Beeinträchtigtes Onshore-System**
Verzögerung des Starts oder des Abbruchs einer Mission, Beeinträchtigung der Statusüberwachung, Verlängerung der zurückzulegenden Wegstrecke, Missachtung von vordefinierten Gefahrenzonen.

R-1.1-2 **Ausfall des Onshore-Systems**
Verhinderung des Starts oder des Abbruchs einer Mission, Ausfall der Statusüberwachung, Unterbundene Entgegennahme von Benachrichtigungen oder Warnungen.

R-1.1-3 **Fehlerhafte Onshore-Kommunikation**
Siehe R-1.1-1.

R-1.1-4 **Ausfall der Onshore-Kommunikation**
Siehe R-1.1-2.

R-1.1-5 **Störung des Steuergeräts**
Siehe R-1.1-1, Verursachen von Sachschäden, Gefährdung der Gesundheit unbeteiligter Personen.

R-1.1-6 **Ausfall des Steuergeräts**
Siehe R-1.1-2, Verursachen von Sachschäden, Gefährdung der Gesundheit unbeteiligter Personen.

Kollisionserkennung und -verhütung

R-1.2-1 **Falsch-Positiv-Erkennung**
Verlängerung der zurückzulegenden Wegstrecke, unnötige Reduzierung der verbleibenden Batteriekapazität, Einschränkung des ursprünglichen Missionsradius.

R-1.2-2 **Falsch-Negativ-Erkennung**
Nichtbeachtung von Verkehrsregeln, Unterbrechung des Verkehrsflusses, Beeinträchtigung der Umwelt, Beschädigung des Wasserfahrzeugs, Verursachen von Sachschäden, Gefährdung der Gesundheit unbeteiligter Personen.

R-1.2-3 **Beeinträchtigte Abstandssensorik**
Kurzfristig eingeschränkte Verfügbarkeit, Erkennung inexistenter Hindernisse, siehe R-1.2-1 und R-1.2-2.

R-1.2-4 **Ausfall der Abstandssensorik**
Dauerhaft eingeschränkte Verfügbarkeit, siehe R-1.2-2.

R-1.2-5 **Fehlerhafte Bus-Kommunikation**
Kurzfristig eingeschränkte Verfügbarkeit, Weiterleitung inexistenter Hindernisse, Verfälschung objektspezifischer Informationen, siehe R-1.1-1 und R-1.1-2.

R-1.2-6 **Ausfall der Bus-Kommunikation**
Dauerhaft eingeschränkte Verfügbarkeit, siehe R-1.2-2.

R-1.2-7 **Störung des Steuergeräts**
Siehe R-1.2-5.

R-1.2-8 **Ausfall des Steuergeräts**
Dauerhafter Systemausfall, siehe R-1.2-2.

Sicherheitsziele

Aus den im Vorfeld identifizierten Gefahreneinflüssen H-1.1-1 bis H-1.1-6 und H-1.2-1 bis H-1.2-8 sowie den aus ihnen abgeleiteten Risiken R-1.1-1 bis R-1.1-6 und R-1.2-1 bis R-1.2-8 ergeben sich eine Reihe von Sicherheitszielen (*Safety Goals*), die im Folgenden kurz vorgestellt werden sollen.

Die bisherigen Erkenntnisse weiter verfeinernd, bilden die nachfolgend abgeleiteten Zielvorstellungen die Grundlage für die Formulierung der Sicherheitsanforderungen in Abschnitt 6.2.2 und beschreiben das gewünschte Systemverhalten zunächst auf einer hohen Abstraktionsebene. Obgleich die Einzelheiten der konkrekten Implementierung hierbei unbehandelt bleiben, können bereits Ideen für die spätere Realisierung gesammelt werden.

Im Gegensatz zu den vorherigen Kapiteln werden in diesem Abschnitt erstmalig auch konkrete Erwartungen an das übergeordnete *Onboard-System* (I-1) spezifiziert. Den kompositionellen Ansatz der Analyse berücksichtigend, sind diese Zielvorstellungen gleichwohl für die in den vorherigen Abschnitten spezifizierten Items I-1.1 und I-1.2 gültig.

Onboard-System

G-1-1 **Detektion von Steuergeräte-Störungen**
Eine kurzfristige Einschränkung der spezifizierten Funktionalität ($t < 2\,s$) muss erkannt und auf der Bedieneinheit optisch angezeigt werden.

G-1-2 **Detektion von Steuergeräte-Ausfällen**
Eine längerfristige oder dauerhafte Einschränkung der spezifizierten Funktionalität ($t \geq 2\,s$) muss erkannt, auf der Bedieneinheit optisch angezeigt und durch einen Neustart der betroffenen Komponente behandelt werden.

Missionsplanung und -durchführung

G-1.1-1 **Detektion von externen Systemfehlern**
Ein Fehler bei der Verarbeitung und der Bereitstellung der Missionsdaten muss erkannt und auf der Bedieneinheit optisch angezeigt werden.

G-1.1-2 **Detektion von externen Systemausfällen**
Ein dauerhaft fehlerhaftes Verhalten und ein Ausfall des Systems müssen erkannt und auf der Bedieneinheit optisch angezeigt werden.

G-1.1-3 **Detektion von Kommunikationsfehlern**
Ein Fehler bei der Übertragung der Missionsdaten muss erkannt, auf der Bedieneinheit optisch angezeigt und durch die erneute Anforderung der fehlerhaften oder fehlenden Datenpakete behandelt werden.

G-1.1-4 **Detektion von Kommunikationsausfällen**
Eine dauerhaft fehlerhafte Onshore-Kommunikation muss erkannt, auf der Bedieneinheit optisch angezeigt und durch einen automatischen Wiederaufbau der Verbindung behandelt werden.

Kollisionserkennung und -verhütung

G-1.2-1 **Verhinderung von Falsch-Positiv-Erkennungen**
Das System darf keine inexistenten Hindernisse erkennen.

G-1.2-2 **Verhinderung von Falsch-Negativ-Erkennungen**
Das System muss jedes erkennbare Hindernis detektieren.

G-1.2-3 **Detektion von Sensorfehlern**
Ein Fehler bei der Entgegennahme und der Verarbeitung der Sensordaten muss erkannt, auf der Bedieneinheit optisch angezeigt und durch eine Reduzierung der Fahrgeschwindigkeit behandelt werden.

G-1.2-4 **Detektion von Sensorausfällen**
Ein dauerhaft fehlerhaftes Verhalten und ein Ausfall des Systems müssen erkannt, auf der Bedieneinheit optisch angezeigt und durch die Überführung in einen sicheren Zustand behandelt werden.

G-1.2-5 **Detektion von Kommunikationsfehlern**
Ein Fehler bei der Übertragung der Sensordaten muss erkannt, auf der Bedieneinheit optisch angezeigt und durch eine Reduzierung der Fahrgeschwindigkeit sowie eine erneute Anforderung der Messwerte behandelt werden.

G-1.2-6 **Detektion von Kommunikationsausfällen**
Eine dauerhaft fehlerhafte Bus-Kommunikation und ein Ausfall dieser müssen erkannt, auf der Bedieneinheit optisch angezeigt und durch die Überführung in einen sicheren Zustand behandelt werden.

Sicherheitsanforderungen

Um aus den Sicherheitszielen G-1-1 bis G-1-2, G-1.1-1 bis G-1.1-4 und G-1.2-1 bis G-1.2-6 konkrete Rückschlüsse auf das Konzept zur Realisierung unseres Wasserfahrzeugs zu ermöglichen, erfolgt in diesem Abschnitt die Verfeinerung der bisherigen Analyseergebnisse in konkrete Sicherheitsanforderungen.

Im Gegensatz zu den in Kapitel 3.4 definierten funktionalen und nichtfunktionalen Anforderungen folgt die Bezeichnung der Zielvorstellungen in diesem Abschnitt einem alternativen Prinzip. Während die erste Ziffer jedes Anforderungsbezeichners das durch ihn adressierte Item referenziert, entspricht die zweite Nummer der Ordnungszahl des übergeordneten Sicherheitsziels. Die analog angewandte, fortlaufende Indizierung gewährleistet auch weiterhin eine hohe Transparenz der vollzogenen Entwurfsschritte.

In Ergänzung zu den aufgeführten Zielvorstellungen können auch die Anforderungen N-KE-1 bis N-KE-13 angeführt werden, da auch sie sich mit der Behandlung von Einschränkungen der Sichtverhältnisse beschäftigen.

Onboard-System

S-1-1-1	Das System muss Steuergeräte-Störungen detektieren können.
S-1-1-2	Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf Steuergeräte-Störungen aufmerksam machen.
S-1-1-3	Das System muss die Fahrgeschwindigkeit infolge einer Steuergeräte-Störung auf 0 km/h reduzieren.
S-1-1-4	Das System muss die Durchführung der aktuellen Mission infolge einer Steuergeräte Störung pausieren.
S-1-1-5	Das System muss anhaltende Steuergeräte-Störungen ($t \geq 2\,s$) wie einen Steuergeräte-Ausfall behandeln.
S-1-2-1	Das System muss Steuergeräte-Ausfälle detektieren können.
S-1-2-2	Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf Steuergeräte-Ausfälle aufmerksam machen.
S-1-2-3	Das System muss infolge eines Steuergeräte-Ausfalls eigenständig einen Neustart der betroffenen Komponente vornehmen.

Missionsplanung und -durchführung

S-1.1-1-1	Das System muss externe Systemfehler detektieren können.
S-1.1-1-2	Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf externe Systemfehler aufmerksam machen.
S-1.1-1-3	Das System muss anhaltende externe Systemfehler ($t \geq 2\,s$) wie einen externen Systemausfall behandeln.

S-1.1-2-1 Das System muss externe Systemausfälle detektieren können.

S-1.1-2-2 Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf externe Systemausfälle aufmerksam machen.

S-1.1-3-1 Das System muss Kommunikationsfehler detektieren können.

S-1.1-3-2 Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf Kommunikationsfehler aufmerksam machen.

S-1.1-3-3 Das System muss infolge eines Kommunikationsfehlers eigenständig eine erneute Anforderung der fehlerhaften oder fehlenden Datenpakete vornehmen.

S-1.1-3-4 Das System muss anhaltende Kommunikationsfehler ($t \geq 2\,s$) wie einen Kommunikationsausfall behandeln.

S-1.1-4-1 Das System muss Kommunikationsausfälle detektieren können.

S-1.1-4-2 Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf Kommunikationsausfälle aufmerksam machen.

S-1.1-4-3 Das System muss infolge eines Kommunikationsausfalls einen automatischen Wiederaufbau der Verbindung initiieren.

Kollisionserkennung und -verhütung

S-1.2-1-1 Das System muss Falsch-Positiv-Erkennungen verhindern.

S-1.2-1-2 Das System muss für die Kollisionserkennung nichtrelevante Objekte als unkritisch identifizieren.

S-1.2-1-2.1 Das System muss Reflexionen als unkritisch identifizieren.

S-1.2-1-2.2 Das System muss Wellen als unkritisch identifizieren.

S-1.2-1-2.3 Das System muss Brücken als unkritisch identifizieren.

S-1.2-1-2.4 Das System muss Bauten und Gegenstände am Ufer als unkritisch identifizieren.

S-1.2-2-1 Das System muss Falsch-Negativ-Erkennungen verhindern.

S-1.2-2-2 Das System muss für die Kollisionserkennung relevante Objekte als kritisch identifizieren.

S-1.2-2-2.1 Das System muss Segel- und Motorboote als kritisch identifizieren.

S-1.2-2-2.2 Das System muss Schwimmer und Badegäste als kritisch identifizieren.

S-1.2-2-2.3 Das System muss Tiere als kritisch identifizieren.

S-1.2-2-2.4 Das System muss Treibgut als kritisch identifizieren.

S-1.2-2-2.5 Das System muss Seezeichen als kritisch identifizieren.

S-1.2-2-2.6 Das System muss Uferbereiche als kritisch identifizieren.

S-1.2-2-2.7 Das System muss starken Pflanzenbewuchs als kritisch identifizieren.

S-1.2-3-1 Das System muss Sensorfehler detektieren können.

S-1.2-3-2 Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf Sensorfehler aufmerksam machen.

S-1.2-3-3 Das System muss die Fahrgeschwindigkeit infolge eines Sensorfehlers auf 2 km/h reduzieren.

S-1.2-3-4 Das System muss anhaltende Sensorfehler ($t \geq 2\,s$) wie einen Sensorausfall behandeln.

S-1.2-4-1 Das System muss Sensorausfälle detektieren können.

S-1.2-4-2 Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf Sensorausfälle aufmerksam machen.

S-1.2-4-3 Das System muss die Fahrgeschwindigkeit infolge eines Sensorausfalls auf 1 km/h reduzieren und zum Ausgangspunkt der Mission zurückkehren.

S-1.2-5-1	Das System muss Kommunikationsfehler detektieren können.
S-1.2-5-2	Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf Kommunikationsfehler aufmerksam machen.
S-1.2-5-3	Das System muss die Fahrgeschwindigkeit infolge eines Kommunikationsfehlers auf 2 km/h reduzieren.
S-1.2-5-4	Das System muss infolge eines Kommunikationsfehlers eigenständig eine erneute Anforderung der fehlerhaften oder fehlenden Messwerte vornehmen.
S-1.2-5-5	Das System muss anhaltende Kommunikationsfehler ($t \geq 2\,s$) wie einen Kommunikationsausfall behandeln.
S-1.2-6-1	Das System muss Kommunikationsausfälle detektieren können.
S-1.2-6-2	Das System muss den Nutzer durch die Ausgabe einer optischen Warnung auf der Bedieneinheit auf Kommunikationsausfälle aufmerksam machen.
S-1.2-6-3	Das System muss die Fahrgeschwindigkeit infolge eines Kommunikationsausfalls auf 1 km/h reduzieren und zum Ausgangspunkt der Mission zurückkehren.

6.2.3 Funktionales Sicherheitskonzept

Aus den im Vorfeld hergeleiteten Sicherheitszielen und -anforderungen sowie den erlangten Kenntnissen über die Struktur und die Funktionsweise der relevanten Systembestandteile lässt sich in einem nächsten Schritt das funktionale Sicherheitskonzept des *Onboard-Systems* beschreiben.

Zur Realisierung der Sicherheitsanforderungen S-1-1-1 bis S-1-1-5 und S-1-2-1 bis S-1-2-3 ist hierfür zunächst die Überwachung aller eingesetzten Steuergeräte und Systemkomponenten erforderlich. Die fortlaufende Klassifikation der Recheneinheiten in die drei Zustände *funktionsfähig*, *beeinträchtigt* und *ausgefallen* ermöglicht eine zeitnahe und angemessene Reaktion des Systems auf das Auftreten eines fehlerhaften Verhaltens. Durch einen Neustart oder das Zurücksetzen der beeinträchtigten Komponente ist die funktionale Sicherheit des Systems stets gewährleistet.

Damit auch der Nutzer über mögliche Probleme oder Fehler in Kenntnis gesetzt werden kann, leitet das *Onboard-System* entsprechende Benachrichti-

gungen an das *Onshore-System* weiter. Während der Entwicklungsphase erleichtert diese Tatsache zudem die Identifikation möglicher Fehlerquellen und trägt somit nachhaltig zu der Stabilität des Systems bei.

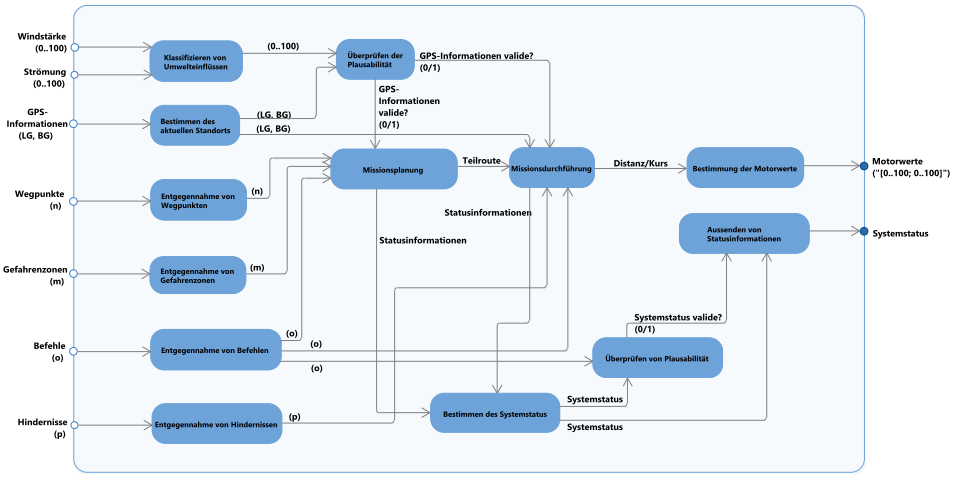


Abbildung 6.4: Funktionales Sicherheitskonzept des Items I-1.1

In Abb. 6.4 ist das funktionale Sicherheitskonzept des Items *Missionsplanung und -durchführung* (I-1.1) graphisch dargestellt. Um das spezifizierte Verhalten des geplanten Teilsystems möglichst detailliert beschreiben zu können, erfolgt die Vorstellung der Funktionalität anhand der drei zentralen Mechanismen, die die zuvor definierten Sicherheitsanforderungen S-1.1-1-1 bis S-1.1-4-3 neben dem eigentlichen Systemverhalten realisieren.

Plausibilität der GPS-Positionsdaten Unter Verwendung der Ergebnisse der initialen Aktivitäten *Klassifizieren von Umwelteinflüssen* und *Bestimmen des aktuellen Standorts* wird in einem ersten Schritt die Plausibilität der GPS-Positionsdaten überprüft. Treten hierbei Abweichungen von dem erwarteten Verhalten auf, so werden die beteiligten Subsysteme über die fehlerhaften Positionsdaten benachrichtigt und eine Fehlermeldung in den zu übertragenden Systemstatus einbezogen.

Plausibilität der Missionsdaten Während das betrachtete Item mithilfe der aus den Aktivitäten *Entgegennahme von Wegpunkten* und *Entgegennahme von Gefahrenzonen* gewonnenen Informationen die Berechnung einer neuen Route vornimmt, erfolgt das periodische *Bestimmen des Systemstatus*. Durch einen Vergleich der aus dem Systemverhalten abgeleiteten Informationen mit den eingehenden Befehlen kann die Plausibilität der Missionsdaten überprüft werden.

Übersteuerung mit RC-Steuersignalen Um eine sichere Durchführung praktischer Systemtests unter realen Einsatzbedingungen zu ermöglichen, können die durch das System erzeugten Motorwerte mithilfe einer RC-Fernbedingung übersteuert werden. Die aufgrund der geringen Bedeutung für die Konzeption des Systems nicht dargestellten Bestandteile stellen so indirekt einen wirksamen Schutz vor möglichen Kollisionen dar und erleichtern das Testen der sicherheitskritischen Funktionen.

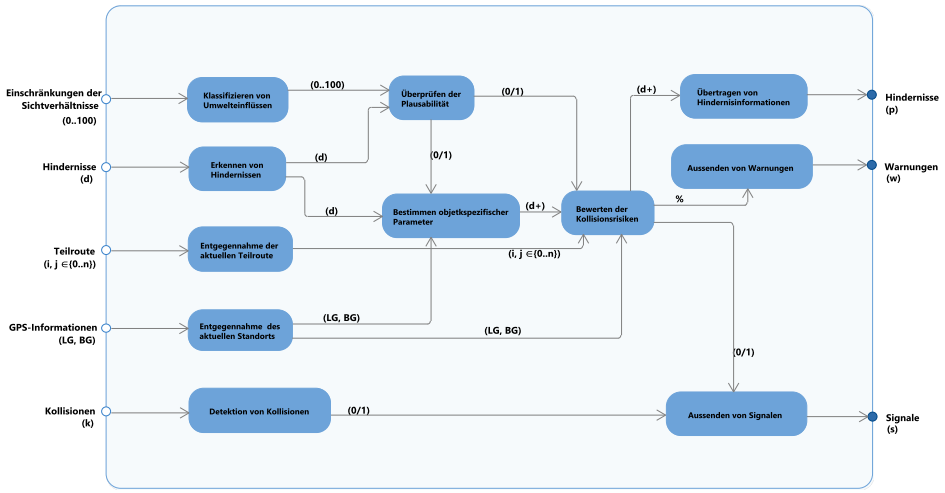


Abbildung 6.5: Funktionales Sicherheitskonzept des Items I-1.2

Das funktionale Sicherheitskonzept des Items *Kollisionserkennung und -verhütung* (I-1.2) ist in Abb. 6.5 dargestellt. Wie auch schon bei der Betrachtung des Items I-1.2 beschränken sich die nachfolgenden Ausführungen zur Erläuterung des Systemverhaltens auf die sicherheitsbezogene Funktionalität, deren Eigenschaften sich unmittelbar aus den Sicherheitsanforderungen S-1.2-1-1 bis S-1.2-6-3 sowie den impliziten Wechselwirkungen mit dem Item *Missionsplanung und -durchführung* (I-1.1) ergeben.

Plausibilität der Abstandsinformationen Unter Einbezug der durch die Aktivität *Klassifizieren von Umwelteinflüssen* bewerteten Umweltbedingungen sowie den Eigenschaften der detektierten Hindernisse lassen sich natürliche Einschränkungen der Sichtverhältnisse, wie beispielsweise Helligkeitsveränderungen im Tagesverlauf oder ein durch Niederschläge bedingtes Signalrauschen, sicher erkennen.

Durch einen Abgleich der vorliegenden Messwerte können die Plausibilität der Abstandsinformationen überprüft, die gewonnenen Erkenntnisse in das *Bestimmen objektspezifischer Parameter* einbezogen und

andere Komponenten durch das *Aussenden von Warnungen* über die Einschränkungen der Sichtverhältnisse informiert werden.

Deaktivierung bei Übersteuerung Um eine wechselseitige Beeinflussung der beiden Items zu verhindern, wird die Funktionalität des Teilsystems *Kollisionserkennung und -verhütung* (I-1.2) im Falle einer Übersteuerung der Motorwerte durch RC-Steuersignale deaktiviert. Die Fähigkeit, erkannten Hindernissen auszuweichen, einbüßend, wird dem Nutzer somit die vollständige Kontrolle über das Wasserfahrzeug übertragen.

6.2.4 Technisches Sicherheitskonzept

Das im vorherigen Abschnitt beschriebene funktionale Sicherheitskonzept weiter verfeinernd, beschreibt das technische Sicherheitskonzept die Allokation der funktionalen Blöcke auf konkrete Hardwarebausteine und formuliert wichtige Anforderungen an die technische Realisierung des *Onboard-Systems*.

Für die Umsetzung der in Abschnitt 6.2.3 beschriebenen sicherheitsorientierten Überwachung der eingesetzten Steuergeräte soll ein Netz sogenannter `heartbeat`-Leitungen in die bestehenden Strukturen integriert werden. Die von jeder Komponente ausgesendeten periodischen Signale werden an eine zentrale `Safety-ECU` weitergeleitet, die die vorhandenen Recheneinheiten in die bereits bekannten Zustände *funktionsfähig* (dauerhafter Empfang), *beeinträchtigt* (kurzfristige Unterbrechung) und *ausgefallen* (dauerhafter Signalverlust) klassifiziert. Die ebenfalls vorhandenen `reset`-Leitungen (low-aktiv) ermöglichen weiterhin den Neustart einzelner Komponenten.

Um den Nutzer durch die Aussendung von Fehlermeldungen auf mögliche Probleme aufmerksam machen zu können, müssen zudem geeignete technische Einrichtungen für das Senden entsprechender Benachrichtigungen entwickelt und in das existierende System integriert werden.

Die vorherigen Ausführungen einbeziehend, ist das technische Sicherheitskonzept des Items *Missionsplanung und -durchführung* (I-1.1) in Abb. 6.6 graphisch dargestellt. Analog zu der Beschreibung des funktionalen Sicherheitskonzepts sind die Erläuterungen der relevanten Mechanismen in vier Abschnitte gegliedert, die sich jeweils mit einem zentralen Aspekt der Implementierung beschäftigen.

Absicherung der XBee-Verbindung Um eine fehlerfreie Übertragung von Missionsdaten und Fehlermeldungen zwischen dem *Onboard-* und dem *Onshore-System* gewährleisten zu können, wird der Kommunikationskanal zwischen den beiden Teilsystemen mit einer zyklischen Redundanzprüfung (CRC-16) versehen. Durch die erneute Anforderung fehlerhafter Datenpakete können die Sicherheit und die Zuverlässigkeit des Systems nachhaltig erhöht werden.

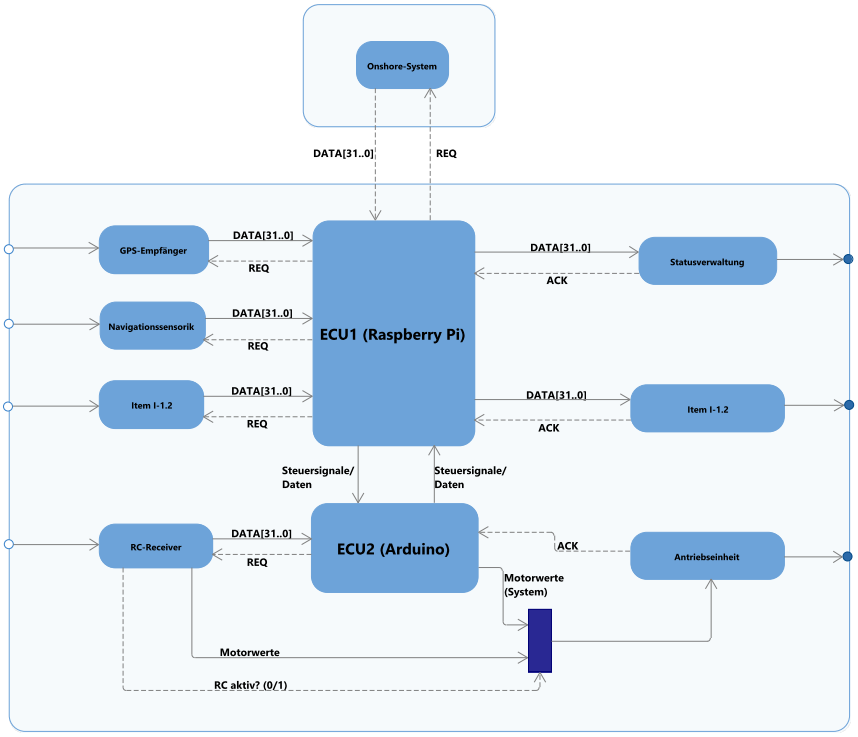


Abbildung 6.6: Technisches Sicherheitskonzept des Items I-1.1

Registrierung von Verbindungsausfällen Während der Einsatz einer zyklischen Redundanzprüfung die unbemerkte Verwendung fehlerhafter Datenpakete verhindert, ergibt sich aus den bereits realisierten Methoden zur Erkennung dieser Artefakte die Möglichkeit, die in den Sicherheitszielen G-1.1-1 bis G-1.1-4 beschriebene Detektion von externen System- und Kommunikationsfehlern ohne Mehraufwand zu realisieren.

Hardwarebasierte Übersteuerungsschaltung Um jede denkbare Beeinflussung durch andere Systembestandteile auszuschließen und die Ausfallsicherheit dieser kritischen Komponente zu erhöhen, wird die Übersteuerungsschaltung in einem Hardwarebaustein implementiert. Eine von den übrigen Steuergeräten unabhängige Stromversorgung hält die Manövrierfähigkeit des Wasserfahrzeugs selbst in außergewöhnlichen und vermeintlich unvorhersehbaren Situationen aufrecht.

REQ/ACK-Konzept für Datenleitungen Um Störungen oder den vollständigen Ausfall der Kommunikation mit einzelnen Teilsystemen erkennen zu können, werden sämtliche Datenleitungen mit einem Rückkanal zur Bestätigung des Erhalts einer Nachricht versehen. Das so

implementierte REQ/ACK-Konzept kann weiterhin zur Detektion von Störungen oder Ausfällen einzelner Steuergeräte verwendet werden.

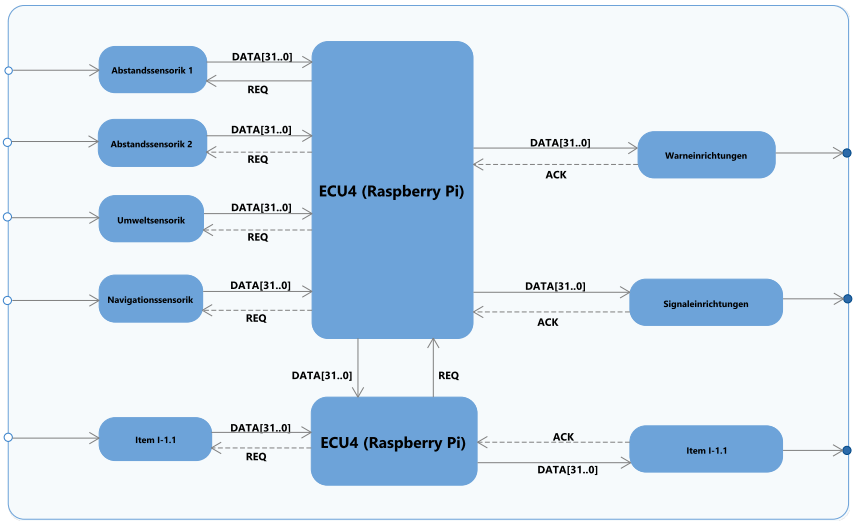


Abbildung 6.7: Technisches Sicherheitskonzept des Items I-1.2

In Abb. 6.7 ist das technische Sicherheitskonzept des Items *Kollisionserkennung und -verhütung* (I-1.2) dargestellt. Für die spätere Umsetzung der spezifizierten Komponente sind dabei insbesondere die beiden nachfolgend aufgeführten Aspekte von Bedeutung, die sich unmittelbar aus den im Vorfeld erörterten Sicherheitszielen und -anforderungen herleiten lassen.

Redundante Abstandssensorik Um die direkten und indirekten Auswirkungen eines Hardwareausfalls auf ein minimales Maß zu reduzieren, wird die Abstandssensorik im Rahmen dieses Projektvorhabens als redundantes System ausgelegt. Die in der Abbildung als *Abstandssensorik 1* und *Abstandssensorik 2* bezeichneten Komponenten werden im späteren System durch eine Kamera und einen Laserscanner instanziiert.

REQ/ACK-Konzept für Datenleitungen Analog zu den vorhergehenden Ausführungen wird die Übertragung von Steuersignalen und Datenpaketen auch im Wirkungsgefüge des Items I-1.2 durch die Implementierung eines einfachen REQ/ACK-Konzepts abgesichert.

6.3 Funktionale Systemarchitektur

In diesem Kapitel wird das gewählte Konzept zur Realisierung der im Vorfeld definierten Funktionalität erläutert. Ausgehend von den in Kapitel 3

erhobenen Anforderungen sowie dem im vorherigen Abschnitt erläuterten Sicherheitskonzept erfolgt im Anschluss an eine systematische Verfeinerung der Systemarchitektur die vollständige Beschreibung des geplanten Systems.

Im Folgenden wird die funktionale Systemarchitektur des *Onboard-Systems* beschrieben, die in Abb. 6.8 dargestellt ist. Dabei werden zunächst die Ein- und Ausgaben des Systems betrachtet, woraufhin die Kernfunktionalitäten des Systems sowie ihre funktionalen Zusammenhänge erläutert werden.

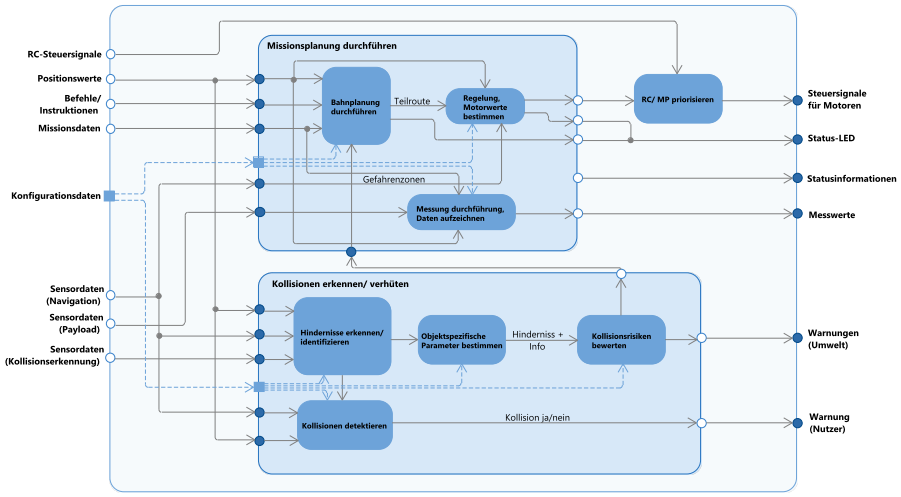


Abbildung 6.8: Funktionale Systemarchitektur des *Onboard-Systems*

Systemeingänge

RC-Steuersignale Die RC-Steuersignale dienen der manuellen Steuerung des Bootes.

Positionswerte Die Positionswerte liefern die aktuellen Positionswerte zusammen mit einem Zeitstempel pro Positionswert.

Befehle/Instruktionen Das System erhält vom Onshore-System Steuerbefehle wie zum Beispiel das Starten einer Mission.

Missionsdaten Die Missionsdaten beinhalten die anzusteuernden Wegpunkte und Gefahrenzonen, die nicht vom Boot befahren werden dürfen.

Sensordaten (Navigation) Zur Unterstützung der Navigation werden zusätzlich Sensordaten aufgenommen, die zum Beispiel Informationen zur Ausrichtung des Bootes liefern.

Sensordaten (Payload) Die Payload-Sensordaten nehmen die zu untersuchenden Umweltparameter auf.

Sensordaten (Kollisionserkennung) Für die Kollisionserkennung werden Sensordaten entgegengenommen, die Informationen über die Umwelt enthalten und Rückschlüsse auf umgebende Hindernisse zulassen.

Konfigurationsdaten Aus Entwicklungsgründen besteht die Möglichkeit bestimmte Funktionalitäten des Systems anzupassen.

Systemausgänge

Steuersignale für Motoren Mithilfe der Steuersignale werden die Bootsmotoren angesteuert.

Status-LED Die Status-LED signalisiert den aktuellen Zustand der Missionsplanung für die Nutzer vor Ort.

Statusinformationen Die Statusinformationen enthalten alle relevanten Informationen zum aktuellen Status des *Onboard-Systems*.

Messwerte Die Messwerte enthalten die aufgenommenen Sensordaten der Payload-Sensoren, die mit Informationen über exakte Position sowie die Uhrzeit der Messung angereichert sind.

Warnungen (Umwelt) Warnungen an die Umwelt werden abgegeben, wenn eine Kollision wahrscheinlich ist, um Anwesende zu warnen.

Warnung (Nutzer) Der Nutzer erhält eine Warnung, wenn eine Kollision detektiert wurde.

Auf funktionaler Ebene besteht das *Onboard-System* aus zwei wesentlichen Bestandteilen. Zum einen die *Durchführung der Missionsplanung* und zum anderen die *Kollisionserkennung/-verhütung*.

Innerhalb der Durchführung der Missionsplanung wird die Bahnplanung vollzogen. Sie berechnet in Abhängigkeit von der aktuellen Position, der vorhandenen Gefahrenzonen und der Zielpositionen eine Teilroute. Diese dient der Regelung als Eingang, um Motorwerte zu bestimmen und das Boot zur Zielposition zu befördern. Falls RC-Steuersignale vorhanden sind, werden diese höher priorisiert als die Steuersignale aus der Regelung der Missionsdurchführung.

Eine weitere Funktion der Missionsdurchführung ist das Erfassen von Messdaten. Dort werden im Abgleich zwischen den Wegpunkten einer Mission und der aktuellen Position Messwerte der Payload-Sensoren aufgezeichnet.

Die Erkennung und Verhütung von Kollisionen teilt sich in vier funktionale Komponenten auf. Die Erkennung und Identifizierung von Hindernissen erfolgt in Abhängigkeit von den Positionswerten, den Sensordaten zur Navigationssensorik und spezifischen Sensordaten zur Kollisionserkennung.

Für die erkannten Hindernisse werden im Folgenden objektspezifische Parameter wie zum Beispiel der Kurs und die Geschwindigkeit der möglichen

Hindernisse ermittelt. Eine Bewertungsfunktion wertet die Gesamtinformationen zu einem Hindernis aus und bestimmt ein Kollisionsrisiko, welches dann zu einer Warnung der Umwelt führen kann oder der Bahnplanung eine dynamische Gefahrenzone hinzufügt, welcher ausgewichen werden muss.

Außerdem wird zusammen mit den aktuellen Positionswerten, den Sensordaten zur Navigation und den identifizierten Hindernissen überprüft, ob eine Kollision stattgefunden hat. Im Falle einer Kollision kann der Nutzer so über den unvermeidbaren Zusammenstoß benachrichtigt werden und erhält eine Mitteilung über den aktuellen Standort des beschädigten Wasserfahrzeugs.

6.4 Technische Systemarchitektur

Die aus der funktionalen Systemarchitektur abgeleiteten technischen Systemarchitektur ist in Abb. 6.9 graphisch dargestellt. Bei dem Steuergerät `ECU1` handelt es sich um einen Raspberry Pi, der Schnittstellen zu GPS und der Messeinheit (IMU) bereitstellt und die Daten abfragt. In der Steuereinheit wird zudem die Bahnplanung und Regelung vorgenommen. Weiterhin kommuniziert die `ECU1` mit dem XBee-Modul, welches für die Übertragung der Daten mit der Onshore-Software zuständig ist.

Die Steuereinheit `ECU2` kann Steuersignale mit der `ECU1` austauschen. Die Regelung kann somit beispielsweise Steuerimpulse an die Motoren weiterleiten oder Motordaten empfangen. Sollte die Fernbedienung (`RC`) aktiv sein, so werden Signale der `ECU1` ignoriert. Die Steuersignale für die Motoren werden in der `Leistungselektronik` verstärkt und an die Motoren weitergeleitet.

Das Arduino Modul `ECU3` nimmt Sensordaten entgegen und leitet diese an `ECU1` weiter. Diese versieht die Daten mit den aktuellen Positionsdaten und einem Zeitstempel für die spätere Auswertung.

Die `ECU4` ist eine eigene Steuereinheit und erfüllt sämtliche Aufgaben der `Kollisionserkennung`. Sie nimmt die Daten der Abstands- und Umweltsensorik entgegen und verarbeitet diese, indem erkannte Objekte ausgegeben werden. Die integrierten Module der `Kollisionsverhütung` stellen kollisionsverhütende Maßnahmen bereit und beeinflussen im Falle einer drohenden Kollision die Regelung im Steuergerät `ECU1`.

6.5 Entwurf der Teilkomponenten

Im nachfolgenden Kapitel werden die für die Realisierung der Teilkomponenten gewählten Konzepte vorgestellt. Hierfür werden die Systeme zunächst von ihrer unmittelbaren Systemumgebung abgegrenzt und ihre Schnittstellen definiert. Weiterhin werden erste Architekturentwürfe erarbeitet.

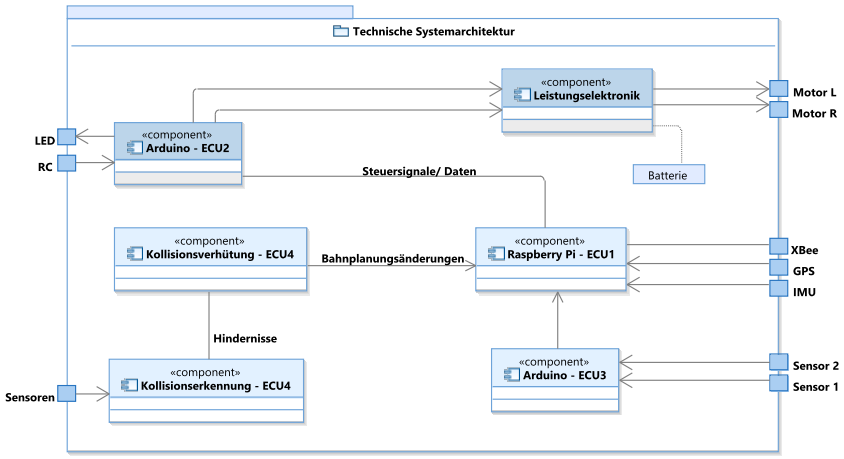


Abbildung 6.9: Technische Systemarchitektur des *Onboard-Systems*

6.5.1 Kollisionserkennung

Die Kollisionserkennung ist ein weit gefasster Begriff. Um den Entwicklungsprozess strukturieren zu können, muss eine Abgrenzung der Systemumgebung vorgenommen werden. Auch die notwendigen Rahmenbedingungen für den Einsatz des Systems werden ausgearbeitet und präsentiert.

Abgrenzung der Systemumgebung

Wie bereits in den Anforderungen aus Kapitel 3.4 und den funktionalen und technischen Systemarchitekturen aus Abschnitt 6.3 und 6.4 ersichtlich wird, lässt sich das System der Kollisionserkennung von den Modulen der Kollisionsverhütung abgrenzen.

Unter der Kollisionserkennung wird im Allgemeinen das Erkennen von möglichen Berührung zweier oder mehrerer geometrischen Objekten im zwei- oder dreidimensionalen Raum bezeichnet, wobei wir uns im Rahmen dieser Projektgruppe auf den zuletzt genannten beschränken. Erst im Anschluss an die Erkennung der Objekte erfolgt die sogenannte Kollisionsverhütung oder -behandlung, die sich mit konkreten Maßnahmen und Handlungen befasst, um eine Kollision mit den detektierten Objekten zu verhindern.

Schnittstellen

Nachdem in den vorherigen Kapiteln bereits die vollständigen Architekturen des übergeordneten *Onboard-Systems* hergeleitet wurden, erfolgt an dieser Stelle die Ableitung der Schnittstellen der Teilkomponente *Kollisionserken-*

nung. In Abbildung 6.10 sind die zu diesem Zweck identifizierten Ein- und Ausgänge vereinfacht dargestellt.

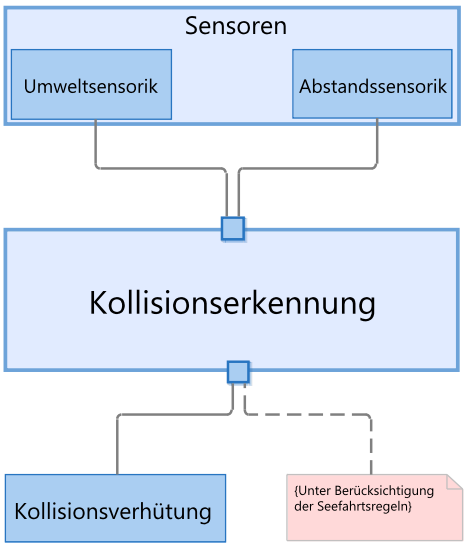


Abbildung 6.10: Schnittstellen der *Kollisionserkennung*

Da die Detektion von kritischen Objekten sowohl von Hardware- als auch von Softwarekomponenten abhängig ist, muss ein Kommunikationsweg geschaffen werden, über den die Sensordaten an die entsprechenden Erkennungsmodule übergeben werden. Von den Sensoren werden Daten erfasst, die in ihrem Rohzustand noch wenig Aussagekraft besitzen.

Im Anschluss an die Aufbereitung der möglicherweise heterogenen Rohdaten müssen diese über weitere Schnittstellen an die Komponenten übergeben werden, die für die Kollisionsverhütung relevant sind. Entweder liegt ein Seezeichen vor, das mit den bekannten Seefahrtsregeln abgeglichen werden muss, oder ein unbekanntes Objekt wird erkannt. In beiden Fällen muss eine Schnittstelle zur Kollisionsverhütung definiert werden.

Ablauf der Kollisionserkennung

Der vorgesehene Ablauf der Kollisionserkennung kann anhand des Aktivitätsdiagramms aus Abb. 6.11 nachvollzogen werden. Die in dem Diagramm verwendeten Aktivitäten werden im Folgenden noch einmal näher erläutert.

Der Prozess der Kollisionserkennung startet zeitgleich mit dem Beginn der Mission und soll den gesamten Zeitraum über aktiv sein, während sich das Boot auf einem Kurs bewegt. Sie ist dafür zuständig, die Umgebung kontinuierlich zu beobachten.

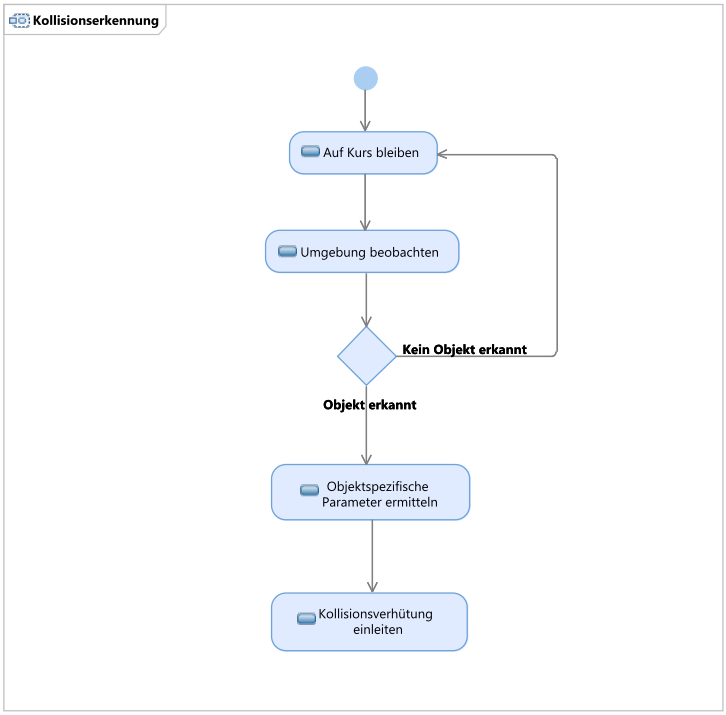


Abbildung 6.11: Aktivitätsdiagramm der *Kollisionserkennung*

Solange kein Hindernis auf der Route erkannt wird, soll der geplante Kurs eingehalten werden. Sobald die Kollisionserkennung ein Objekt erkannt hat, werden zunächst die objektspezifischen Parameter des Hindernisses ermittelt und ein gesammelten Daten an die Kollisionsverhütung weitergegeben.

Beobachtung der Umgebung Um kontinuierliche Sicherheit gewährleisten zu können, ist es notwendig zu jeder Zeit eine Aussage über mögliche Hindernisse treffen zu können. Hierzu ist es sinnvoll, nicht nur Objekte vor dem Boot, sondern auch Objekte an den Seiten oder im Bereich hinter dem Heck zu erfassen. Auch unter Wasser kann es durch Untiefen oder große Objekte zu Kollisionen kommen.

Erfassen von Objektdaten Ein erkanntes Objekt kann Eigenschaften besitzen, die für die Verhütung von Kollisionen entscheidend sind. Daher ist es für den Prozess der Kollisionserkennung notwendig, grundlegende objektspezifische Parameter zu ermitteln. Dazu gehören insbesondere:

Beschaffenheit Im Anwendungsfeld der von uns entwickelten Plattform kann es sich entweder ein statisches oder ein bewegtes Hindernis handeln.

Ferner ist es interessant zu wissen, ob das entsprechende Objekt ein Lebewesen, ein Gegenstand oder ein Seezeichen ist.

Größe Um ein Kollisionsrisiko bewerten zu können, ist das Wissen über die Größe des Objekts unerlässlich.

Entfernung Die Entfernung eines Objektes ist ein ausschlaggebender Faktor bei der Abwendung von Gefahren. Je nach Distanz zum Objekt müssen kollisionsverhütende Maßnahmen zu einem früheren oder einem späteren Zeitpunkt eingeleitet werden.

Richtung Neben den aufgeführten Punkten ist das Verhalten bewegter Objekte schwerer vorauszusehen. Die Richtung eines Objektes kann Aufschluss darüber geben, ob es in naher Zukunft zu einer Kollision kommen könnte.

Geschwindigkeit Analog zu der Richtung ist auch die Geschwindigkeit ein relevanter Parameter, um die Kollisionswahrscheinlichkeit eines nahenden Objektes einzuschätzen.

Übergabe von Parametern Nach Erfassung aller Objekte, die ein Kollisionsrisiken für das zu realisierende System darstellen könnten, muss eine Ausweichroute geplant werden. Dazu werden die Eigenschaften der erkannten Objekte in geeigneter Form an die Kollisionsverhütung weitergegeben. Im Falle eines erkannten Seezeichens werden zusätzliche Informationen mit konkreten Handlungsanweisungen übermittelt.

Architektur der Teilkomponente

Die allgemeine Architektur des Systems zur Kollisionserkennung lässt sich einteilen in einen Hardware- und einen Softwareteil.

Der Hardwareteil beinhaltet mindestens eine Komponente, um Informationen über die Umgebung zu sammeln. Diese Informationen werden über eine Datenleitung an ein Modul gegeben, welches die Daten der Komponenten sammelt und aufbereitet (*Sensor Fusion Module*).

Die aufbereiteten Daten werden weitergeleitet an ein Modul, welches sich mit der Auswertung dieser Daten beschäftigt (*Object Detection Module*). Relevante Informationen werden hierbei bewertet und zusammengefasst, sodass im Anschluss Objekte vom System erkannt wurden.

Die Positionsdaten werden nun im Identifizierungsmodul (*Object Identification Module*) verarbeitet und den Objekten Identifier zugewiesen, um eine spätere Kursberechnung durchführen zu können. Anhand der gesammelten Daten kann nun entschieden werden, ob es sich um ein Seezeichen oder kollisionsgefährdendes Objekt handelt. Im Anschluss an die Bewertung werden die angereicherten Informationen an die Kollisionsverhütung übertragen.

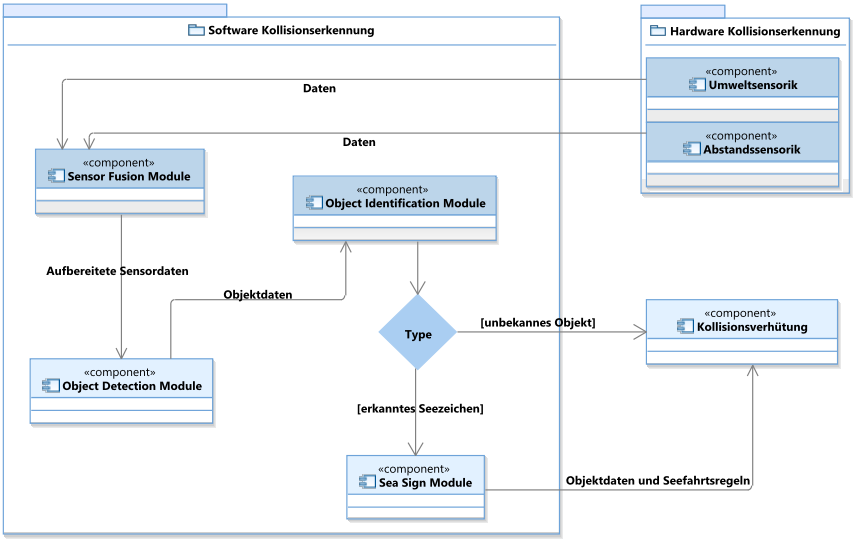


Abbildung 6.12: Architektur der *Kollisionserkennung*

6.5.2 Kollisionsverhütung

Um ein System zur Kollisionsverhütung zu entwickeln, ist der Ablauf und der Zusammenhang zur Kollisionserkennung von besonderem Interesse. Daher wird in diesem Abschnitt ein Aktivitätsdiagramm zur Kollisionsverhütung vorgestellt. Des Weiteren spielt die Regelung des Wasserfahrzeugs auf einem Ausweichkurs eine bedeutende Rolle. Hierzu wird die Reglerstruktur erläutert und die Auswahl eines geeigneten Reglers beschrieben.

Ablauf der Kollisionsverhütung

In Abbildung 6.13 ist das Verfahren zur Kollisionsverhütung graphisch dargestellt. Ausgangspunkt um eine Kollision zu vermeiden ist, dass diese im Vorfeld mit geeigneten Hilfsmitteln erkannt wurde (vgl. Abschnitt 6.5.1).

Um einen Kurs möglichst effizient zu berechnen, ist es sinnvoll Parameter wie die Wind- oder Strömungsgeschwindigkeit des Wasser in der Berechnung zu berücksichtigen. Die Umweltdaten sollen zu diesem Zweck gesammelt, ausgewertet und nach Möglichkeit berücksichtigt werden.

Zur Überprüfung der Sensoren wie GPS und der Motoren sollte im nächsten Schritt eine Diagnose über entsprechende Komponenten vorliegen, sodass wenn nötig Warnungen über einen Ausfall ausgegeben werden können und die Missionsplanung informiert wird.

Unter Berücksichtigung der Hindernisposition und der Umweltdaten wird im nächsten Schritt eine korrigierte Route berechnet. Hierbei wird festge-

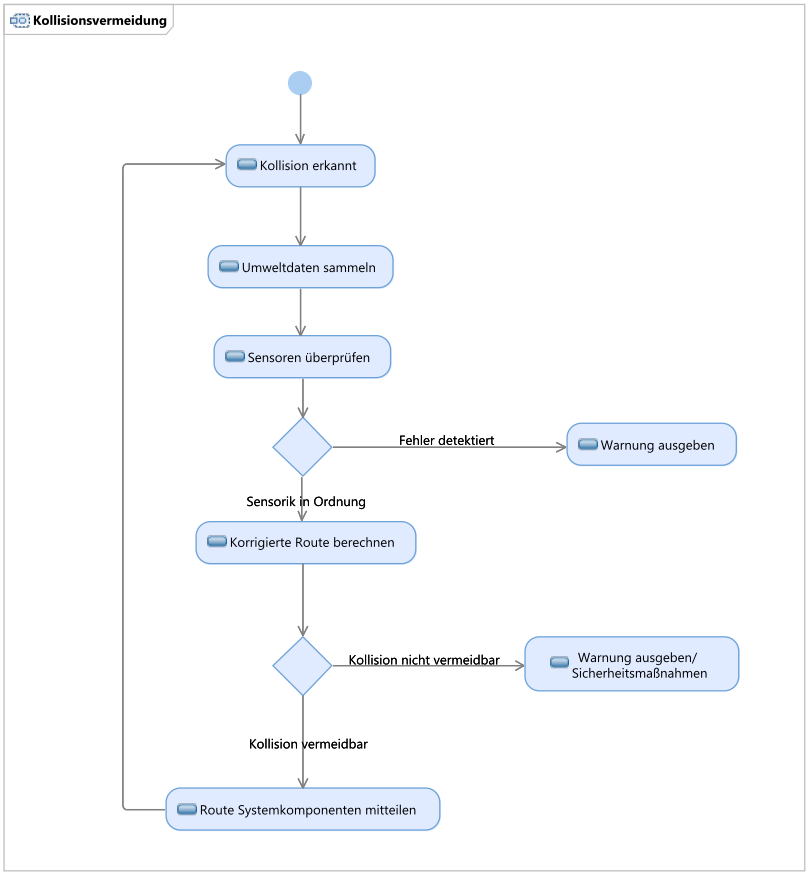


Abbildung 6.13: Aktivitätsdiagramm der *Kollisionsverhütung*

stellt, ob eine Kollision vermeidbar ist oder nicht. Ist dies nicht der Fall sollen Warnungen ausgegeben werden, die mögliche Teilnehmer an der Kollision auf das Wasserfahrzeug aufmerksam machen. Außerdem sollen Sicherheitsmaßnahmen getroffen werden, um das Ausmaß der Kollision zu verringern.

Ist eine Kollision vermeidbar, soll die korrigierte Route anderen Systemkomponenten wie der Missionsplanung mitgeteilt werden. Hierfür sollen die Daten derart gespeichert werden, dass diese aus nach Systemausfall wiederverwendet werden können.

Um einer möglichen Kollision entgehen zu können ist es essentiell, dass die zum Ausweichen angefahrene Position möglichst schnell und genau angefahren wird. Um dieses sicherstellen zu können, muss die Positionsregelung überarbeitet werden.

Reglerstruktur

Um eine effektive Regelung aufbauen zu können, muss zuerst eine Systemanalyse bzgl. der Sensoren und Aktoren durchgeführt werden. In Abb. 6.14 wird gezeigt, dass es seitens der Sensorik den GPS-Empfänger gibt. Hiermit können neben der Position auch der Kurs und die Geschwindigkeit bestimmt werden. Seitens der Aktoren stehen die Motoren mit zugehöriger Leistungselektronik zur Verfügung. Hiermit muss sowohl die Geschwindigkeit als auch der Kurs eingestellt werden. Die Auswertung des GPS-Signals und die Berechnung der Sollwerte für Kurs und Geschwindigkeit werden in der Recheneinheit vollzogen.

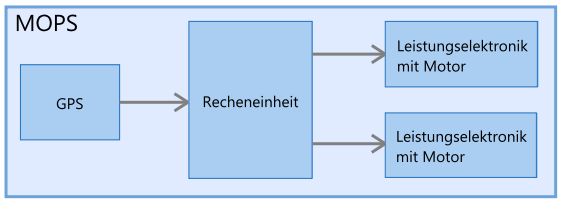
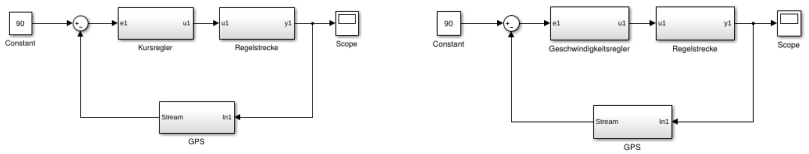


Abbildung 6.14: Komponenten des Regelungssystems

Wie in Abb. 6.14 zu sehen ist, sind die einzigen Aktoren zum Einstellen des Kurses die beiden Motoren. Daher bietet es sich an, das Prinzip des Torque-Vectorings hierfür zu nutzen. Hierbei werden mindestens zwei Antriebseinheiten so angesteuert, dass sich eine Richtungsänderung des anzutreibenden Fahrzeuges ergibt. Die Stellgröße der Geschwindigkeit und die des Kurses können addiert werden und damit wäre auch eine Kursänderung ohne aktiven Vortrieb möglich.



(a) Kursregler (b) Geschwindigkeitsregler

Abbildung 6.15: Exemplarische Regler für Kurs und Geschwindigkeit

Die Abb. 6.15 zeigt den prinzipiellen Aufbau der Regler für Geschwindigkeit und Kurs. Die Regelstrecke ist für beide Regler gleich, diese wird durch die beiden Motoren dargestellt. Da beide Ausgangssignale der Regler die Motoren ansteuern, gibt es den konzeptionellen Ansatz beide Signale zu addieren

und die Regler parallel zu betreiben. Der Geschwindigkeitsregler wird für beide Motoren einen Ansteuerwert zwischen $-100\,\%$ und $+100\,\%$ ausgeben.

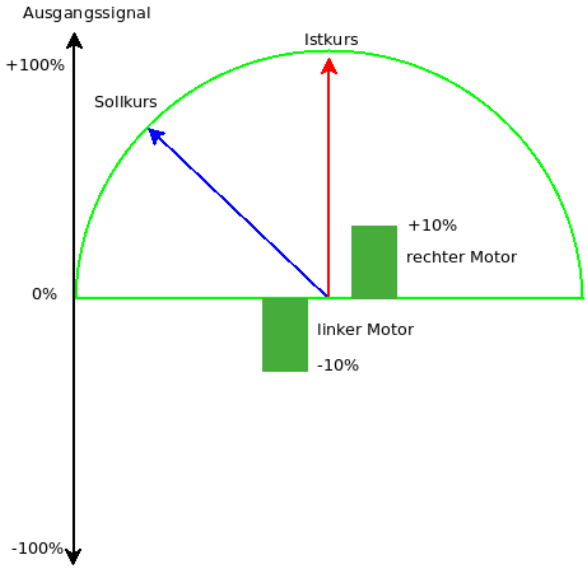


Abbildung 6.16: Ausgangssignale des Kursreglers

Der Kursregler wird einen für beide Motoren unterschiedlichen Wert ausgeben, mit dem die Richtungsänderung des Wasserfahrzeugs beschrieben wird. Abbildung 6.16 zeigt dieses exemplarisch. Beide Werte werden zum Ansteuersignal addiert. Abbildung 6.17 zeigt die beiden Regler und deren Zugriff auf die Regelstrecke. Um die Sollwerte für die Geschwindigkeits- und Kursregelung zu generieren, werden die beiden Regler mit einer Positionsregelung überlagert. Hier wird aus der Differenz von Soll- und Istposition der Sollkurs berechnet. Die Geschwindigkeit wird in Abhängigkeit des Abstands berechnet. Hieraus ergibt sich, dass die Positionsregelung einen Vektor generiert, der auf eine Länge von Null geregelt werden soll.

Im weiteren Verlauf des Projekts gilt es die nötigen Regelparameter zu bestimmen und die Reglerstruktur bzgl. ihrer Wirksamkeit zu verifizieren.

Reglerauswahl

Um die hier geforderten Aufgaben zu erfüllen, stehen der PID-Regler und die Fuzzy-Regelung und das neuronale Netz als Positionsregler zur Verfügung. Aufgrund der Komplexität wird von der Implementierung eines neuronalen Netzes abgesehen. Es sollte aber in den nachfolgenden Projektgruppen geprüft werden, ob ein Neuronales Netz als selbstlernender Regler in Betracht gezogen werden sollte.

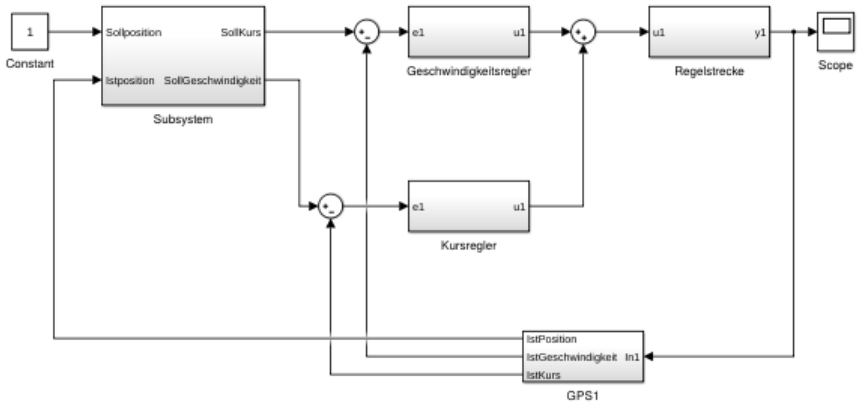


Abbildung 6.17: Aufbau des Regelungssystems

Für den Kursregler sollte ein PID-Regler gewählt werden, da die Regelstrecke an sich schon ein integrierendes Verhalten aufweist und ein differenzierendes Verhalten evtl. nicht notwendig erscheint, müsste in nur der proportionale Anteil des Reglers bestimmt werden.

Der Geschwindigkeitsregler bekommt als Eingangsgröße den zu überwindenden Abstand geliefert. Dieser Wert könnte durch Ableitung in eine Geschwindigkeit überführt werden auf die geregelt werden soll. Sollte dieses Verfahren gewählt werden, würde sich ein PID-Regler anbieten, bei dem somit alle Geschwindigkeitsstufen zwischen $\pm$ 100 % genutzt werden könnten.

Es könnte aber auch der Abstandswert durch eine Fuzzy-Regelung direkt genutzt werden und durch Erstellen von Regeln in eine Geschwindigkeit überführt werden. Vorteil wäre hierbei, dass die Ableitung des Abstandswertes ausbleiben würde. Nachteil ist, dass die Geschwindigkeit nur in Anzahl der festgelegten Regeln erfolgt. Werden beispielsweise fünf Regeln in der Fuzzy-Regelung definiert, so könnten auch nur fünf verschiedene Geschwindigkeiten angefahren werden.

Zurzeit wird davon ausgegangen, dass der Geschwindigkeitsregler durch einen Fuzzy-Regler realisiert wird. Aus derzeitiger Sicht spricht nichts dagegen das Wasserfahrzeug mit einer begrenzten Anzahl an Geschwindigkeitsstufen zu betreiben. Dieser Nachteil wird dadurch ausgeglichen das keine Parameteridentifikation stattfinden muss. Verifiziert wird die Entscheidung im Laufe des Projekts. Sollte sich die Fuzzy-Regelung als nicht praktikabel erweisen, wird ein PID-Regler als Geschwindigkeitsregler implementiert.

Kapitel 7

Realisierung

Im folgenden Kapitel wird die Realisierung der ausgearbeiteten Konzepte beschrieben. Dabei wird zuerst auf die Hardware der Plattform eingegangen. Diese beinhaltet die Beschreibung der entwickelten Architekturen, den Aufbau des elektrischen Systems sowie Änderungen, die an dem Boot durchgeführt wurden. Darauf folgt die Implementierung der Softwarekomponenten. Diese sind weiter unterteilt in die Implementierung des Onboard- und Onshore-Systems sowie die Kommunikation der Komponenten untereinander. Des Weiteren wird die Realisierung der kamera- und laserbasierten Kollisionserkennung vorgestellt. Abschließend wird auf die Realisierung der Kollisionsverhütung und Regelung eingegangen.

7.1 Hardware

Im Abschnitt Hardware werden die Veränderungen an der Hardware beschrieben und die umgesetzten Verbesserungen am Boot deutlich gemacht. Es wird aufgezeigt und beschrieben, was sich gegenüber der dritten Version, in den Bereichen Architektur und Hardware, des Bootes verändert hat. Da nach der Umsetzung auch die Bedienung nicht mehr aktuell war, wurde von uns ein neues Handbuch angefertigt. Hier sind alle Änderungen detailliert erklärt und dokumentiert. Das Handbuch ist im Anhang unter Kapitel C.4 zu finden.

7.1.1 Architektur

Abbildung 7.1 zeigt einen Überblick über die zentralen Systemkomponenten. Bei den blau markierten Kästchen handelt es sich um die auf dem Wasserfahrzeug untergebrachten Raspberry Pis. Hierbei steht das „MP“ in der Abb. für Missionplanning und „CD“ für Collision Detection. In der Missionsplanung werden Aufgaben wie das Berechnen des Kurses und das Ansteuern der Motoren durchgeführt. Über diesen Rechner ist außerdem eine Kommunikation per XBee mit der Onshore-Software möglich. Darüber hinaus kön-

nen Informationen über erkannte Hindernisse mit der Collision-Detection-Software ausgetauscht werden. Hierfür ist eine Verbindung des Rechners mit dem Access-Point notwendig, damit der Datenaustausch per TCP erfolgen kann. Der Onboard-CD-Rechner führt die Kollisionserkennung per Kamera oder Laserscanner durch. Dieser kann die berechneten Informationen über WLAN mit Hilfe des TCP-Protokolls an den Missionplanning-Rechner, den Laserscanner Viewer und die Onshore-Software senden.

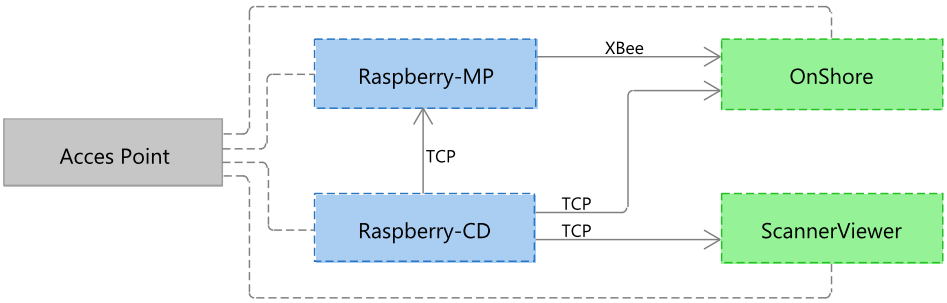


Abbildung 7.1: Übersicht über die Rechnerarchitektur

Bei den grün markierten Kästchen handelt es sich um Programme, die auf einem beliebigen Rechner ausgeführt werden können. Bei den Tests innerhalb der Projektgruppe handelte es sich dabei um einen Laptop, der am Ufer des Sees platziert war. Die Onshore-Software wurde als Basis von der Vorgängergruppen verwendet. Mit Hilfe eines an den Laptop angeschlossenen XBee-Moduls kann die Software mit der Onboard-MP-Software Daten für die Missionsplanung austauschen. Außerdem kann sie von der Collisions-Detection-Software Hindernisse empfangen.

Der Laserscanner Viewer wird zum Überwachen der Hinderniserkennung eingesetzt und detailiert in 7.2.3 beschrieben. Der Viewer kann auf einem beliebigen Rechner ausgeführt werden, muss allerdings für den Datenaustausch per TCP mit dem Access-Point verbunden sein.

Die folgenden zwei Kapitel beschreiben ausführlich die mit den Raspberry-Pis verbundenen Komponenten. Der Fokus liegt hierbei auf der eingesetzten Hardware.

7.1.2 Komponenten derMissionsplanung

Die Missionsplanung und die Steuerung des Wasserfahrzeugs finden auf dem Onboard-Raspberry-Pi statt. Hierfür sind verschiedene Komponenten notwendig, die in Abb. 7.2 dargestellt sind.

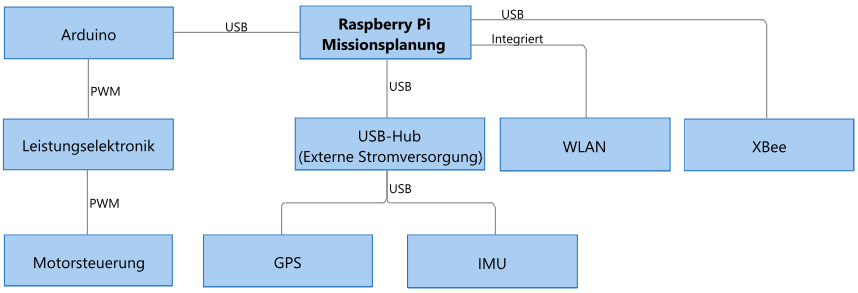


Abbildung 7.2: Übersicht über den Onboard-Rechner

Für die Steuerung der Motoren ist der Rechner mit einem Arduino verbunden. Um die Motoren auf bestimmte Drehzahlen einzustellen, müssen an diese PWM-Signale gesendet werden. Daher besteht die Aufgabe des Arduino darin, die von der Onboard-Software generierten Motorwerte in PWM-Signale umzuwandeln. Da die PWM-Signalstärke für die Außenbordmotoren nicht ausreichend ist, werden diese durch eine Leistungselektronik verstärkt.

Da der Onboard-Rechner nicht über ausreichend USB-Port verfügt, ist an diesem ein USB-Hub angeschlossen, das mit einer externen Stromversorgung ausgestattet ist. Über das Hub werden das GPS-Modul und das IMU-Board mit dem Raspberry Pi verbunden. Über einen weiteren direkten USB-Anschluss ist das XBee-Modul zur Datenübertragung mit der Onshore-Software an den Rechner angeschlossen. Außerdem ist der Onboard-Raspberry mit dem Access-Point per integriertem WLAN-Modul verbunden.

7.1.3 Komponenten der Kollisionserkennung

Abbildung 7.3 zeigt die zentralen Komponenten des Onboard-CD-Rechners. An diesem ist der Laserscanner über eine direkte LAN-Verbindung angeschlossen. Es hat sich gezeigt, dass der Scanner nicht über den Access-Point betrieben werden kann. Auf die Implementierung der laserbasierten Kollisionserkennung und die dafür verwendete Hardware wird in Abschnitt 7.3.2 eingegangen.

Des Weiteren ist ein USB-Hub an den Rechner angeschlossen, welches mit einer Kamera verbunden ist. Dieser Schritt war nötig, da die Kamera während des Betriebs mehr Strom benötigt als ein USB-Port des Raspberry Pis bereitstellt. Anfängliche Tests haben bei direkter Verbindung der Kamera zu diversen Abstürzen des Rechners geführt. Durch die externe Stromversorgung des Hubs konnte dieses Problem gelöst werden. Weitere Details zur Kamera werden in Abschnitt 7.3.1 beschrieben.

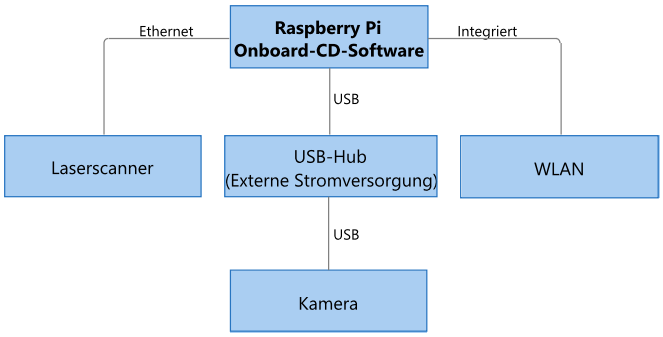


Abbildung 7.3: Übersicht über den Onboard-CD-Rechner

7.1.4 Elektrisches System

Das elektrische System, das die vorangegangen Projektgruppen entwickelt haben, bestand in der Vergangenheit aus vielen selbst gelöteten Spannungswandlern und direkt zu Komponenten verlegten Kabeln. Bereits während des ersten Praxistests zeigten sich einige Schwächen des Systems, die für diverse Stromausfälle auf dem Wasserfahrzeug sorgten und das Gesamtsystem auf diese Weise zum Absturz gebracht haben. Aus diesem Grund wurde die gesamte Elektrik, die für die Stromversorgung nötig ist, intensiv überprüft und im Anschluss überarbeitet.

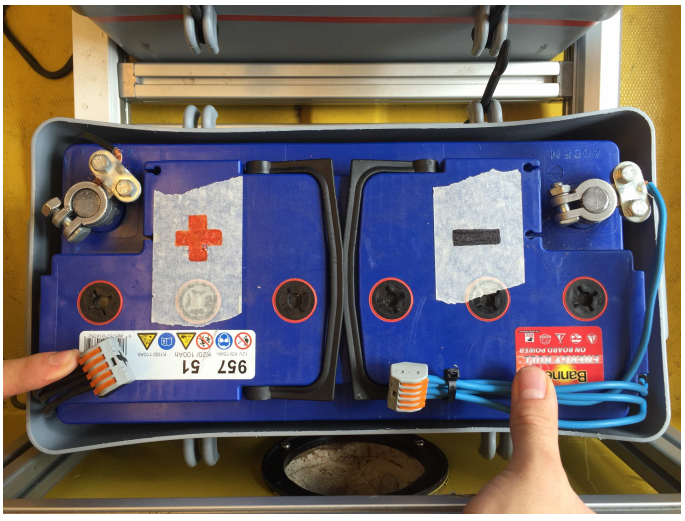
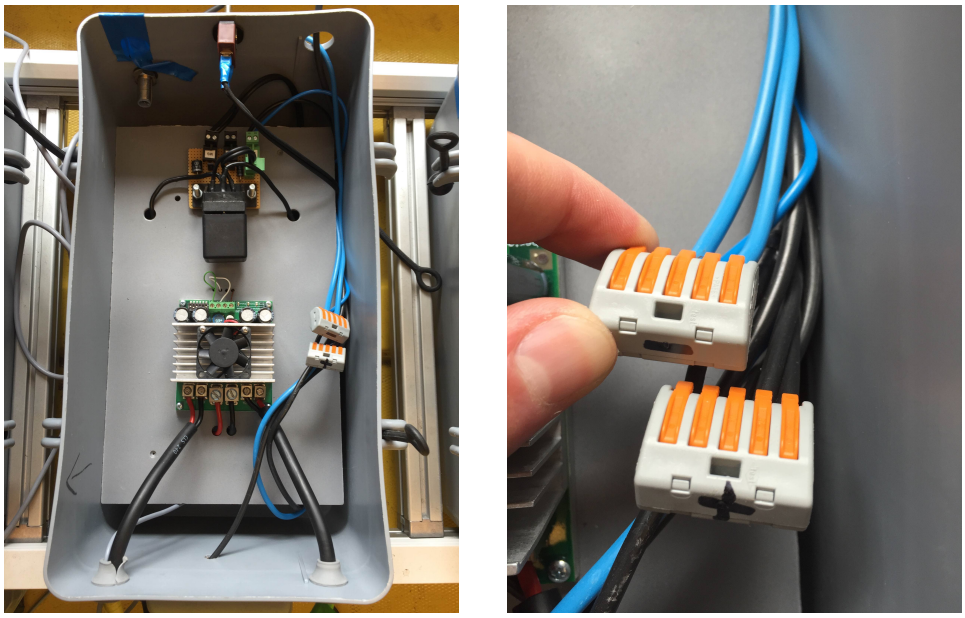


Abbildung 7.4: Verteilerklemmen zum Aufteilen des Batteriestroms

Da nicht in jeder Box eine eigene Leitung zur Spannungsversorgung vorhanden war, wurde beschlossen, dies zunächst umzusetzen. Hierzu wurden in der Batterie-Box Verteilerklemmen angebracht, welche in Abb. 7.4 dargestellt sind. Über die Verteilerklemmen wurde jeweils ein Kabel für den Plus- und Minuspol in jede einzelnen Otterbox verlegt, sodass jede Box über eine 12V Spannungsversorgung verfügt. Durch die Verteilerklemmen können mit wenig Aufwand beliebig viele Verbraucher angeschlossen werden. Die Klemmen bieten zudem den Vorteil, dass keine Schrauben im Vergleich zu Lüsterklemmen gelöst werden müssen, um ein weiteres Kabel anzuschließen. Es ist lediglich notwendig den orangefarbenen Hebel in eine senkrechte Position zu bringen, sodass dann ein weiteres Kabel eingeführt werden kann.

Abbildung 7.5 zeigt die eingebauten Verteilerklemmen in der Otterbox für die Leistungselektronik. Außerdem ist eine Detailansicht der Klemmen dargestellt, die entsprechend der besseren Unterscheidbarkeit mit + und - gekennzeichnet sind. In der Praxis erwiesen sich die Klemmen durch das schnelle hinzufügen oder entfernen von Kabeln als sehr nützlich. Des Weiteren mussten im weiteren Verlauf keine neuen Kabel verlegt werden.



(a) Verteilerklemmen in Otterbox (b) Verteilerklemmen Detailansicht

Abbildung 7.5: Eingebaute Leistungselektronik-Verteilerklemmen

Ein weiterer Verbesserungspunkt war der von der vorherigen Projektgruppe eingebaute Notaus-Schalter. Hierbei handelte es sich um eine Zugkonstruktion, die ihre Zweck nicht im geringsten erfüllte. Dies sei laut Aussage der Vorgänger auf eine Falschbestellung des Schalters zurückzuführen. Aus die-

sem Grund wurde das Notaus-System überarbeitet. Hierzu wurde zunächst ein Notaus-Schalter eingebaut, der das vollständige System vom Strom trennt. Dieser ist in Abb. 7.6(a) dargestellt und lässt sich durch Betätigen des roten Schalters auslösen.

Für die Praxistests war es zu Beginn der Projektgruppe sinnvoll, den Onboard-Raspberry Pi manuell neustarten zu können, da dieser während der Anfangsphase gelegentlich abstürzte. Damit nicht jedes Mal die Otterboxen auf dem See geöffnet werden mussten, wurde der Schalter in Abb. 7.6(b) hinzugefügt. Mit diesem lässt sich die Stromzufuhr zum Raspberry Pi öffnen und schließen. Da der Schalter an der Front des Bootes angebracht ist, konnte mit wenig Aufwand ein Neustart des Rechners erreicht werden.

(a) Notaus-Schalter für das Gesamtsystem (b) An-Aus-Schalter für den Raspberry Pi

Abbildung 7.6: Notaus- und An-Aus-Schalter für den Raspberry Pi

Innerhalb des Wasserfahrzeugs werden diverse Bauteile (z.B.: zwei Raspberry Pis, zwei USB-Hubs) verwendet, die eine 5 V Versorgungsspannung benötigen. Da die Batterie 12 V liefert, ist hier eine Spannungswandlung notwendig. Hierzu wurden in der Vergangenheit selbst gelötete Spannungsteiler eingesetzt. Diese wurde durch entsprechende DC/DC-Konverter ersetzt. Dies bietet einige Vorteile: Zum einen wird das Löten einer Platine und die damit verbundene Fehleranfälligkeit reduziert und zum anderen lassen sich an die DC/DC-Konverter direkt per USB Verbraucher anschließen. Da sowohl der Raspberry Pi als auch die eingesetzten USB-Hubs ihren Strom per USB beziehen, hat der Einsatz der Konverter eine enorme Arbeitserleichterung mit sich gebracht.

Abbildung 7.7 zeigt einen beispielhaften Aufbau. Der Konverter wird zunächst an die Verbindungsklemmen angeschlossen. Jetzt lassen sich über den USB-Anschluss beliebige Verbraucher, die eine 5 V-Versorgungsspannung benötigen, anschließen.

Der eingesetzte Access-Point benötigt eine Spannung von 9 V. Auch hier ist der Aufbau identisch zu Abb. 7.7. Allerdings wurde an dieser Stelle ein 9 V-Konverter verwendet und es musste ein Kabel gelötet werden, dass ein

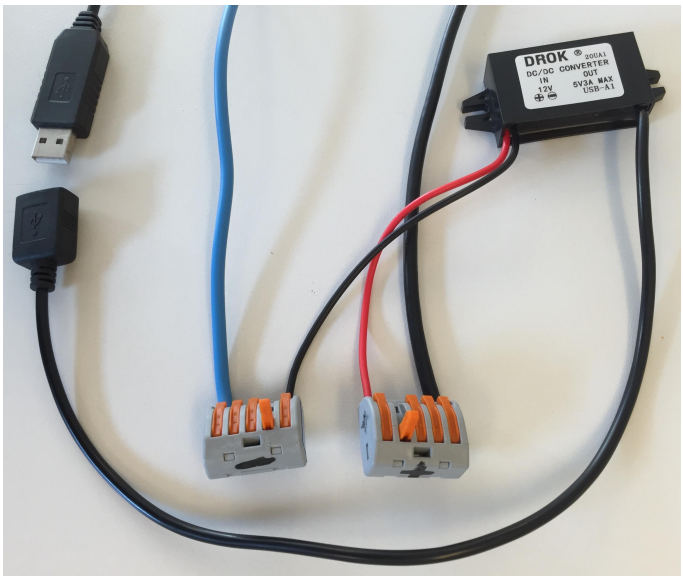


Abbildung 7.7: Beispielaufbau DC/DC-Konverter

Anschließen ermöglicht, denn der Access-Point ist ursprünglich für den Hausbetrieb mit einem handelsüblichen Netzteil ausgelegt gewesen.

7.1.5 Systemumbau

Da das derzeitige System diverse Sicherheitsmängel und Behinderungen für die Schiffsdynamik aufweist, haben wir uns entschieden den grundlegenden Aufbau und die Verkabelung zu optimieren. Des Weiteren wurden zusätzliche Komponenten wie Halterungen für Laser und Kamera dem Boot hinzugefügt und an dieser Stelle beschrieben.

Neuanordnung der Otterboxen

Für eine bessere Balance des Bootes und einer besseren Verkabelung der einzelnen Komponenten wurden die Otterboxen neu angeordnet. Die neue Anordnung ist auf dem Bild 7.8 zu sehen. Für eine bessere Gewichtsverteilung wurde die Batterie neu untergebracht. Anstatt auf der linken Seite, ist die Batterie nun hinten und mittig zu finden. Somit ist das Gewicht der Batterie nun besser verteilt und es muss auf der gegenüberliegenden Seite kein Gegengewicht mehr angebracht werden. Aus diesem Grund entfällt die fünfte Otterbox. Daneben befindet sich die Leistungselektronik, damit im Gegensatz zu vorher, die Kabelwege möglichst kurz zur Batterie sind.

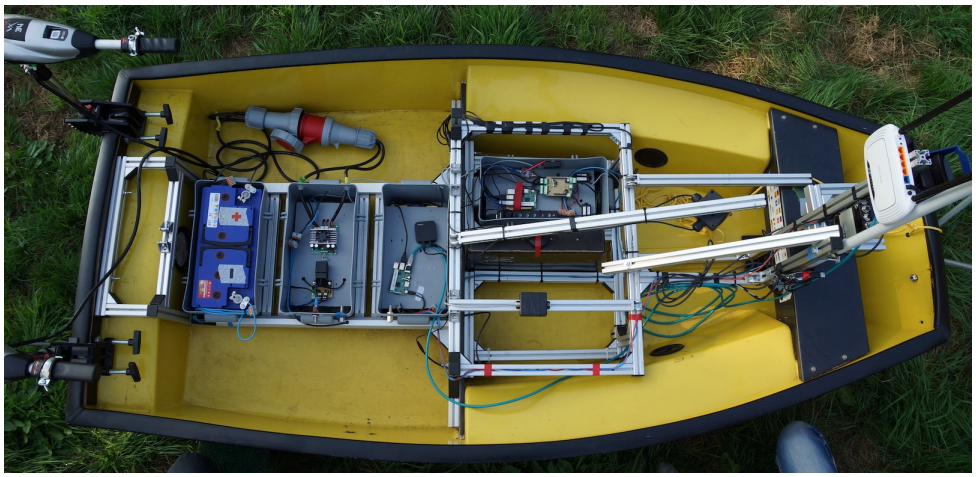


Abbildung 7.8: Anordnung der Otterboxen

Kabelkanäle

Für eine bessere Struktur und damit die Kabel nicht freiliegend sind, wurden von uns Kabelkanäle angebracht. Außerdem schützen sie vor der Sonneneinstrahlung und der damit verbundenen Wärmeentwicklung. Wie auf den folgenden Bildern 7.9 und 7.10 zu sehen, sind die Kabel nun von allen Seiten geschützt.

Abbildung 7.9: Kabelkanal der Otterboxen

Abbildung 7.10: Kabelkanal zum Aluprofil

Hardwarebefestigung

Die Hardware wurde bisher mit einfachem Klebeband am zweiten Boden der Otterboxen befestigt. Um die Befestigung sicherer und einfacher zu gestalten, kamen mehrere Lösungen infrage.

Alternativen

- **Hutschienen:** Die Verwendung von Hutschienen mit entsprechenden Adaptern für die Platinen. Das Problem hierbei ist, dass die Adapter wahrscheinlich nicht für alle Vorhandenen Hardwarekomponenten geeignet sind.
- **Holzrahmen:** Das Herstellen eines Rahmens aus Holzblöcken für die einzelnen Komponenten. Gesichert werden die Platinen zusätzlich von oben durch einen Gummiverschluss, Klettverschluss oder einer Klappe.
- **3D-Druck:** Durch das Herstellen eines Rahmens im 3D-Drucker, kann ein passgenauer Rahmen erstellt werden. Ein Problem ist die sehr aufwendige und komplizierte 3D-Erstellung in einer entsprechenden Software.
- **Magnetklebeband:** Mittels Magnetklebeband, welches in den Otterboxen angebracht wird, sollen die Platinen von unten gehalten werden. Problematisch könnten dabei aber eventuelle Störungen an der Elektronik sein, die durch das Magnetband hervorgerufen werden.

- **Spanngurt:** Durch gebohrte Löcher im zweiten Boden der Otterboxen kleine Spanngurte ziehen und die Platinen so befestigen. Als Problem könnten sich die Metallschnallen der Spanngurte erweisen.

Entscheidung Umgesetzt wurde eine Holzrahmenkonstruktion mit Winkelleisten, die an den zweiten Boden der Otterboxen mit einem extra-starken, wasserfesten Sekunden-Kleber befestigt wurden. Damit die Elektronik nicht aus dem Rahmen herausrutscht, werden die Komponenten mithilfe eines Klettbands an diesem fixiert. An den Ecken der Rahmen wurden Holzscheiben untergelegt, um den Rahmen zu erhöhen. Dadurch kann der Spanngurt unter dem Rahmen gespannt werden und die Elektronik hat nach unten etwas Luft. Zusätzlich wurde durch das Einsägen von Kerben Platz für die Stecker gelassen. Im Gegensatz zum vorher verwendeten Klebeband ist diese Konstruktion sicherer und sie nutzt sich nicht ab. Die folgenden Abbildungen zeigen die Konstruktion der Hardwarebefestigung und wie die Hardware in der Konstruktion befestigt ist.

Kosten

- Winkelleiste: 2.29 Euro
- Klettband: 4.49 Euro
- Holzleiste für die Füße: 1.69 Euro
- Sekundenkleber: 4.49 Euro

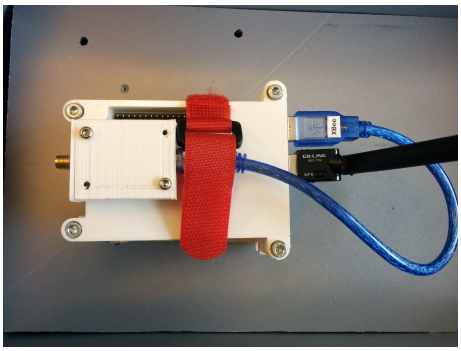
Kamera- und Laserbefestigung

Für die Objekterkennung muss sowohl der Laserscanner SICK-LMS151-10100 als auch die Kamera Logitech C920 am Boot angebracht werden. Unsere Umsetzung wird nachfolgend näher erläutert.

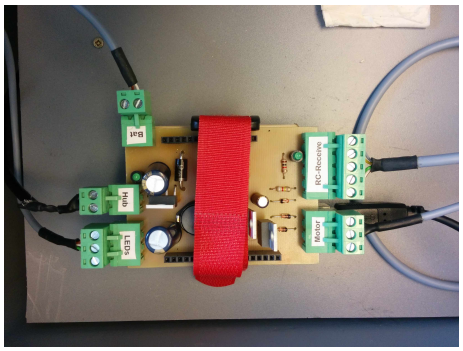
Laserscanner

Für den Laserscanner sollte eine Halterung entstehen, welche direkt am Gestell angebracht wird. Dafür haben wir noch vorhandene Aluprofile auf die entsprechende Länge gekürzt. Die zugeschnittenen Aluprofile wurden mit Nutensteinen, U-Förmig an die vertikale Alustange angebracht. Zu sehen ist die Halterung auf der Abbilung 7.12.

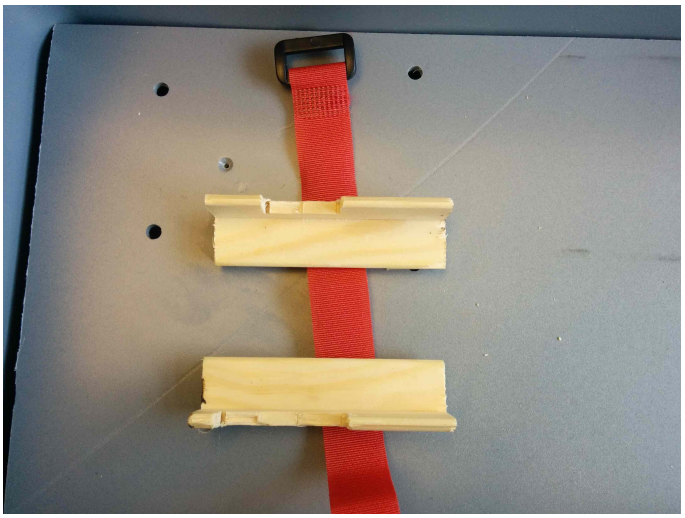
Der Laser sitzt sehr fest und hat die Möglichkeit, die Umgebung im 270°-Winkel zu scannen. Um den Laserscanner abzunehmen, müssen lediglich zwei Schrauben am Aluprofil gelöst werden.



(a) Befestigung RaspberryPi



(b) Befestigung Arduino und Leistungselektronik



(c) Rahmen

Abbildung 7.11: Hardwarebefestigung

Kamera

Für die Kollisionserkennung in der Nähe des Bootes haben wir uns entschieden, eine Kamera zu verwenden. Nach ausgiebiger Recherche fiel die Entscheidung auf die Logitech C920. Für dieses Kameramodell gibt es im Internet mehrere Halterungen zu finden, welche sich mittels eines 3D-Druckers herstellen lassen. Um die Halterungen zu produzieren, durften wir den 3D-Drucker Replicator x2 von MakerBot benutzen. Wir haben uns auf zwei Halterungen festgelegt. Eine der beiden Varianten ist auf dem Bild 7.13 in der MakerBot Software zu sehen.

Die Abbildung 7.14 zeigt den 3D-Drucker während des Drucks der Halterung. Die Fertigstellung des Drucks dauerte mehrere Stunden.

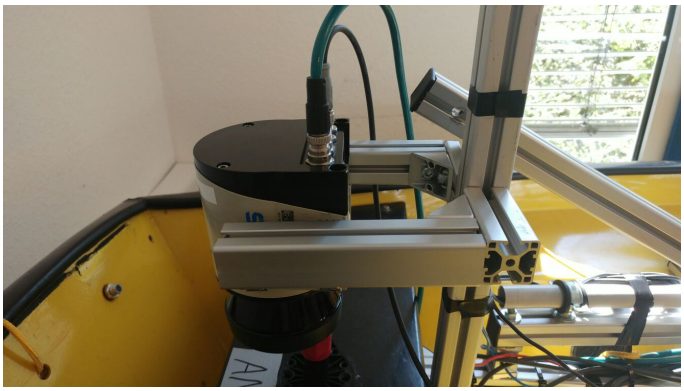


Abbildung 7.12: Halterung mit Scanner

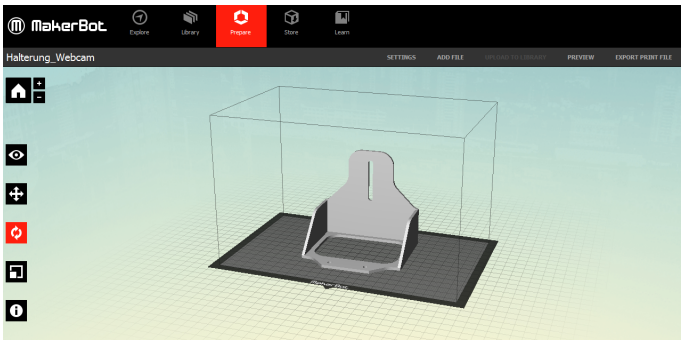


Abbildung 7.13: Halterung 3D

Auch diese Halterung wurde mittels Nutsteinen und den dazugehörigen Schrauben am vertikalen Aluprofil angebracht. Die Halterung für die Kamera befindet sich oberhalb der Halterung für den Laserscanner. Auf Bild 7.15 sind die angebrachte Halterung sowie die zweite Halterung für die Webcam zu sehen.

Das Gelenk der Kamera ist durch diese Konstruktion frei bewegbar. Das Holster für die Kamera (in weiß) kann auch durch die, vielleicht etwas stabilere Halterung(in blau), ausgetauscht werden. Ob die Kamera nun genügend Halt hat oder ob der besagte Austausch sinnvoller ist, muss beim nächsten Praxistest auf dem Tweelbäker See getestet werden.

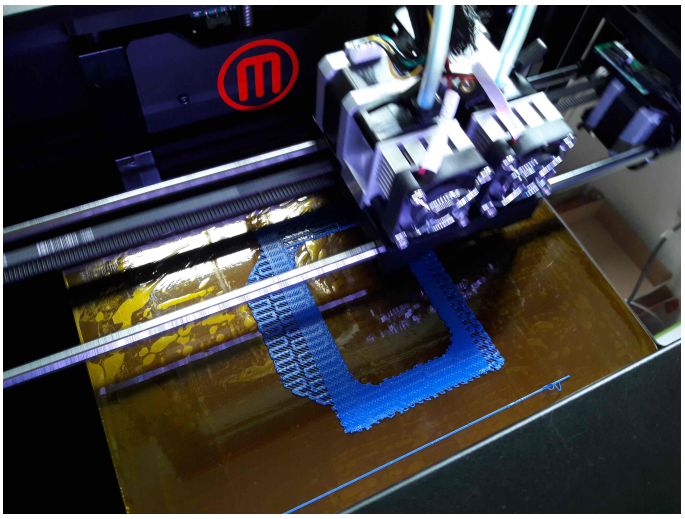


Abbildung 7.14: Live-Druck

(a) Aktuelle Kamerahalterung

(b) Alternative Kamerahalterung

Abbildung 7.15: Kamerahalterungen

Halterung für den Router

Der Router wurde am Mast für die Kamera und den Lidar befestigt, allerdings auf der Rückseite. Dazu wurde der Router mit einer Schraube als Halterung und mit Kabelbindern fixiert. Das Ergebnis ist auf Bild 7.16 zu sehen.

Abbildung 7.16: Halterung für den Router

7.1.6 Moon-Pool

Das Anbringen eines Moon-Pools wurde zu Beginn der Projektphase in Erwägung gezogen. Letztendlich ist die Umsetzung für die Moonpoolanbringung von der MOPS IV-Gruppe aber aus folgenden Gründen nicht mehr umgesetzt worden:

Andere Priorisierungen Im laufe des Projekts hat sich herausgestellt, dass die Gruppe andere Bereiche zur Umsetzung favorisiert. Konkret sollte sich auf die Kollisionserkennung und -vermeidung gekümmert werden.

Wechselseitige Abhängigkeiten Umbau ändert die grundlegende Beschaffenheit des Bootes und dadurch wird direkter Einfluss auf andere Bereiche ausgeübt.

Zeitliche Einschränkungen Durch die Konzentrierung auf andere Bereiche, wie die Kollisionserkennung und -vermeidung, die unserer Gruppe wichtiger erschienen, bleibt für eine vernünftige Umsetzung und Anbringung keine Zeit. Auch der komplette Umbau der Elektronik spielte hier eine Rolle. Dazu kommt, dass die Sensoranbringung nur ein optionales Thema war, welches nur bei ausreichend Zeit umgesetzt werden sollte.

Beeinflussung der Regelung Der Umbau hätte direkt am Anfang stattfinden müssen, damit später die die gemessenen Daten nicht verfälscht und die eingesetzte Regelung nicht behindert wird. Dies sollte später bei den nachfolgenden Projektgruppen beachtet werden.

7.2 Software

Der folgende Abschnitt beschreibt die Implementierung verschiedener Softwarekomponenten und die Verbesserungen des Java Build-Systems. Letztere wird folgenden Unterabschnitt beschrieben. Der zweite Unterabschnitt geht auf die Umsetzungen der Onshore-Software ein. Neben einigen Verbesserungen, die die Benutzerfreundlichkeit der Onshore-Software betreffen, werden auch neu implementierte Funktionen vorgestellt. Im dritten Abschnitt wird die Realisierung des Onboard-Systems beschrieben. Dieses ist aufgeteilt in den bereits bestehenden Teil der Missionsplanung und in den neu hinzugekommenen Teil der Kollisionserkennung. Die Aufteilung erfolgt, da beide Komponenten unabhängig voneinander implementiert wurden. Im abschließenden Kapitel wird auf die Kommunikation innerhalb des Onboard-Systems mittels TCP eingegangen, sowie die Kommunikation zwischen Missionsplanung und Onshore-System mittels XBee.

7.2.1 Umstellung des Build-Systems auf Maven

Das auf Ant basierende Build-System für die Onboard- und Onshore-Software wurde zu Maven portiert. Dadurch sinkt die Einrichtungszeit für die Entwicklungsumgebung und es ermöglicht eine Versionierung. Maven ist ein Build-Manager für Java Projekte. Zentraler Bestandteil eines Maven Projekts ist die `pom.xml`-Datei, welche Hauptverzeichnis des Projekts liegt. Diese beinhaltet mindestens die identifizierenden Eigenschaften `groupId`, `artifactId` und `version`. Weiterhin kann die `pom.xml` weitere Eigenschaften, wie zum Beispiel die Dateicodierung, sowie Plugins, die den Build-Prozess anpassen oder auch Abhängigkeiten zu anderen Projekten beinhalten.

Die allgemeine Projektstruktur der Vorgänger wurde beibehalten, sodass es vier Bibliothekprojekte sowie die zwei Hauptprojekte gibt. Diese sind in Abb. 7.17 zusammen mit ihren Abhängigkeiten untereinander visualisiert. Um Redundanzen zwischen den Projekten zu vermindern wurde eine übergeordnete `pom.xml` definiert. Diese beinhaltet die grundlegende Konfiguration aller Projekte ohne dabei ausführbaren Sourcecode zu besitzen.

Um eine projektunabhängige Entwicklung der Komponenten zu ermöglichen, wurde auf dem bereitgestellten Server ein Maven-Repository aufgesetzt. In diesem werden die versionierten Artefakte der Bibliotheken hochgeladen.

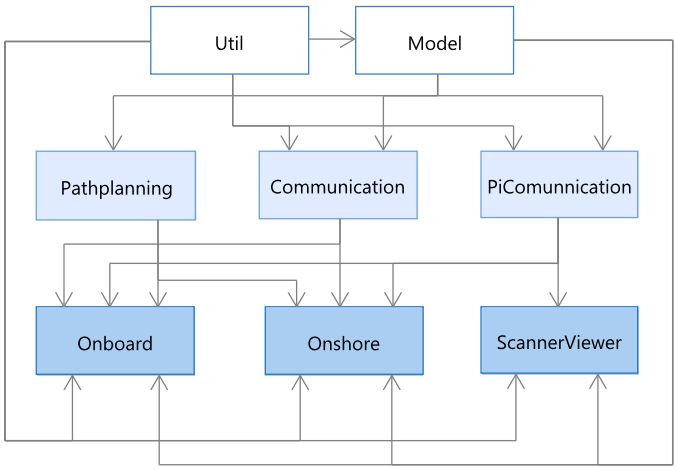


Abbildung 7.17: Abhängigkeiten der Onboard- und Onshore-Software

7.2.2 Onshore-Software

Die bereits vorhandene Onshore-Software zur Missionsplanung und Steuerung der Plattform wurde um zusätzliche Funktionen erweitert, die die Benutzerfreundlichkeit erhöhen. Die implementierten Funktionen werden im Folgenden beschrieben.

Darstellung dynamischer Hindernisse in der Onshore-Software

Die Onshore-Software wurde um die Darstellung von dynamischen Hindernisse erweitert. Über die TCP-Schnittstelle die im Abschnitt 7.2.5 genauer beschrieben wird erhält die Software Informationen über die Hindernisse die von der Kollisionserkennung erkannt wurden. Diese werden analog wie in der Onboard-Software zu Gefahrenzonen konvertiert (eine detaillierte Erlärung dazu findet sich im Kapitel 7.4), die dann auf der Karte gezeichnet werden sollen. Um die dynamischen Gefahrenzonen von den statischen zu Unterscheiden werden diese in einem helleren Gelb gezeichnet. Abb. 7.18 zeigt die Simulation eines dynamischen Hindernisses.

Speichern/Laden-Dialoge

Während der Überprüfung des Ist-Zustandes der Software hat sich heraus gestellt, dass die Dialoge zum Speichern und Laden von Missionen nicht korrekt implementiert waren. Dies wurde in einem ersten Schritt korrigiert. Des Weiteren war es beim Speichern einer Mission möglich, dieser einen Namen mit beliebiger Dateiendung zu geben. Dies wurde durch eine geeignete Über-

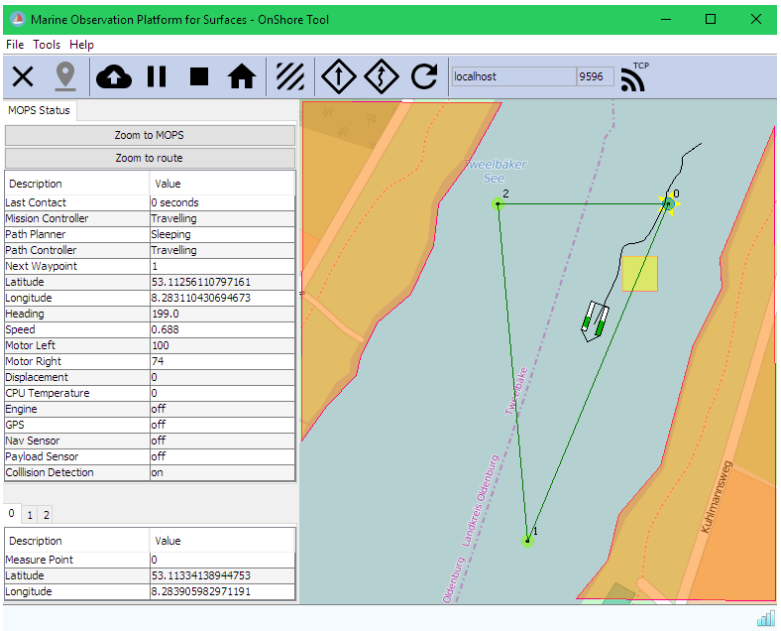


Abbildung 7.18: Beispielhafte Visualisierung dynamischer Hindernisse

prüfung der Eingabe unterbunden. Die Dateiendung für eine Mission wurde auf `.mission` festgelegt.

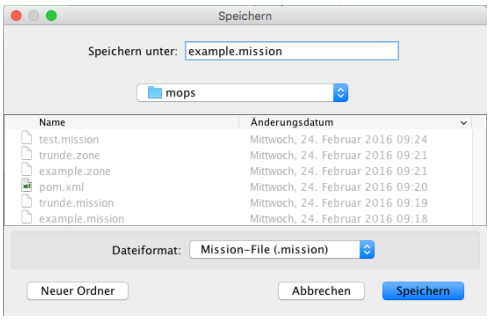
Abbildung 7.19(a) zeigt den Dialog. Wird ein Missionsname ohne Dateiendung in das entsprechende Feld eingegeben, so wird diese nach Betätigen des Speicher-Buttons automatisch ergänzt. Wird eine unzulässige Dateiendung eingegeben, so wird diese ebenfalls korrigiert.

Beim Laden einer Mission war es möglich, Dateien mit beliebiger Endung auszuwählen und als Mission zu laden. Da dies zu Fehlern führen kann, wurde der Dialog mit einem Filter versehen, der nur das Auswählen von Missionen mit der korrekten Dateiendung zulässt. Abbildung 7.19(b) zeigt den Laden-Dialog. Nicht zulässige Missionsnamen werden ausgegraut dargestellt und können durch den Nutzer nicht ausgewählt werden.

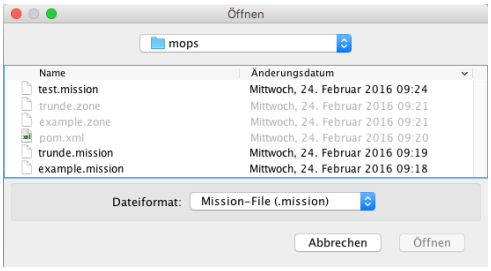
Das analoge Problem zeigte sich beim Im- und Exportieren von Gefahrenzonen. Das oben beschriebene Prinzip wurde für die entsprechenden Dialoge übernommen. Als Dateiendung für eine Gefahrenzone wurde `.zone` festgelegt.

Darstellung der Missionspunkte

Die Analyse des Ist-Zustandes der Onshore-Software hat ergeben, dass die vorhandene GUI optimiert werden kann. Beispielsweise werden die Missionspunkte zwar in einer Karte dargestellt, allerdings wird die Route, die der Bahnplanungsalgorithmus berechnet, nicht eingezeichnet. Dies hat zur Fol-



(a) Dialog zum Speichern einer Mission



(b) Dialog zum Laden einer Mission

Abbildung 7.19: Dialoge zum Speichern und Laden einer Mission

ge, dass die Route des Wasserfahrzeugs bei komplexen Missionen nur schwer nachvollziehbar ist. Aus diesem Grund ist die Onshore-Software dahingehend erweitert worden, dass sowohl die berechnete Route als auch die real von der Plattform befahrene in die Karte eingezeichnet werden. Dies erleichtert zum einen das Erkennen der festgelegten Route, zum anderen kann die Route mit dem real zurückgelegten Weg einfacher verglichen werden, was bei der Wahl und Auslegung eines Positionsregler von Bedeutung ist. Visualisiert wird die geplante Route durch eine grüne und die befahrene durch eine schwarze Linie.

Auch die Darstellung der Informationen zu den einzelnen Wegpunkte wurde angepasst. Bisher konnten die Informationen nur über einen Klick auf entsprechenden Marker abgerufen werden. Eine Tableiste listet alle vorhandenen Wegpunkte mit den dazugehörigen Koordinaten auf.

Abbildung 7.20 zeigt einen Ausschnitt der Karte mit den entsprechenden Visualisierungen. Wird eine Mission geplant, so wird nach jedem Hinzufügen oder Entfernen eines Missionspunktes oder einer Sperrzone die Route neu berechnet. Auf diese Weise ist die in die Karte eingezeichnete Route immer auf dem aktuellen Stand. Die Abbildung zeigt auch, dass die befahrene Route die eingefügten Sperrzonen nicht durchfährt. Beispielsweise wird von Missionspunkt 3 zu 4 ein Umweg über die Spitze des Sterns gefahren.

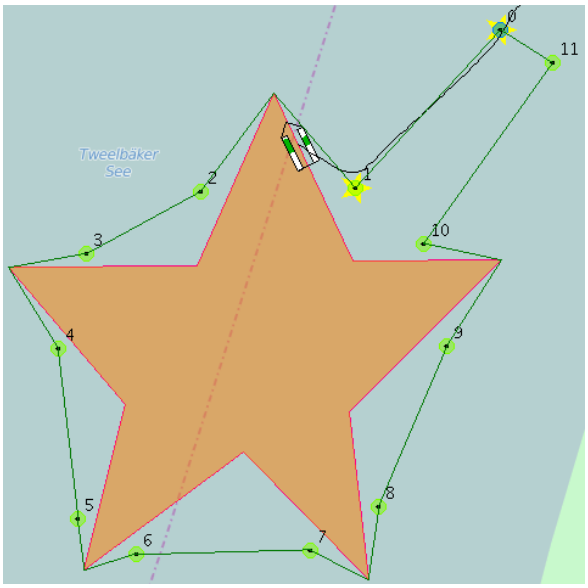


Abbildung 7.20: Visualisierung eines beispielhaften Missionsplans

Um leichter den Unterschied zwischen einem bearbeiteten und einem noch nicht bearbeiteten Missionspunkt erkennen zu können, werden bearbeiteten Missionspunkte durch einen gelben Stern gekennzeichnet. Dies ist in Abb. 7.20 bei den Punkten 1 und 2 zu sehen.

Anzeige der Statusinformationen

Bisher wurden die Statusinformationen des Wasserfahrzeugs nur angezeigt, wenn innerhalb der Karte auf das Symbol für die Plattform geklickt wurde. Da diese Informationen zur Missionsüberwachung allerdings immer eine hohe Relevanz haben, wurde die GUI um eine permanente Statusanzeige ergänzt.

Abbildung 7.21 zeigt diese. Sie beinhaltet neben den Informationen (Position, Motoreinstellungen, usw.) des Wasserfahrzeugs auch zwei Buttons. Der erste zoomt auf die Plattform, wenn man diese beim Erkunden der Karte verloren hat. Der zweite zoomt auf die Route. Hierbei wird immer der als nächstes erreichte Missionspunkt in den Mittelpunkt der Karte gebracht. Die Buttons erleichtern ein schnelles Navigieren durch die Karte.

Tabelle 7.1 listet die in der Onshore-Software verfügbaren Statusinformationen der Plattform mit einer Beschreibung auf. Neu hinzugekommen sind Informationen über den Status der Teilkomponenten des Onboard-Systems.

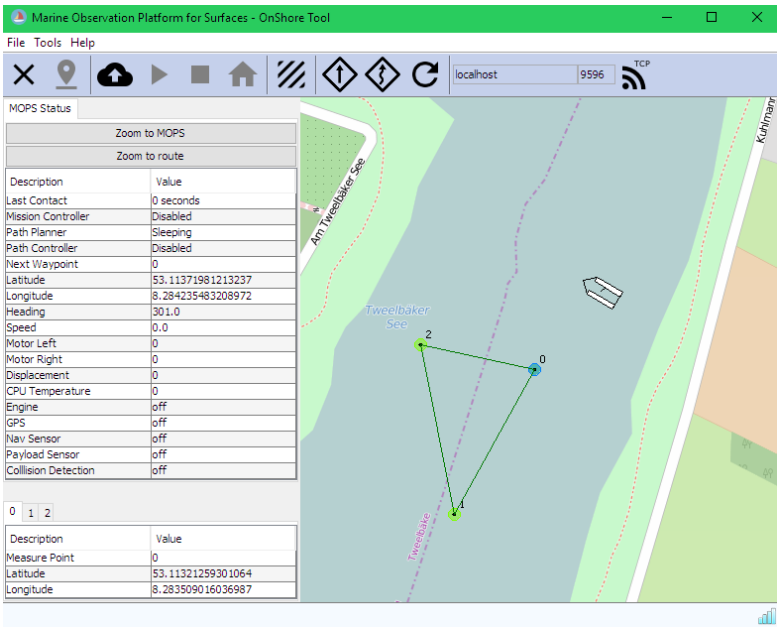


Abbildung 7.21: Statusanzeige in der Onshore-Software

Name	*Beschreibung*
Last Contact	Anzahl der Sekunden seit dem letztem XBee-Kontakt mit dem Boot
Mission Controller	Aktueller Zustand des Missioncontrollers
Path Planner	Aktueller Zustand des Path Planners
Path Controller	Aktueller Zustand des Path Controllers
Next Waypoint	ID des nächsten Waypoints der angefahren werden soll
Latitude	Geographische Breitenposition des Bootes
Longitude	Geographische Längenposition des Bootes
Heading	Ausrichtung des Bootes
Speed	Geschwindigkeit in km/h
Motor Left	Relativer Motorwert für den linken Motor (-100 bis 100)
Motor Right	Relativer Motorwert für den rechten Motor (-100 bis 100)
Engine	Zustand der Motoransteuerung der Missionsplanung
GPS	Zustand des GPS-Moduls des Bootes

Nav Sensor	Zustand des Navigationssensors
Payload Sensor	Zustand der Payload Sensor Ansteuerung der Missionsplanung
Collision Detection	Verbindungszustand der Kollisionserkennung mit der Missionsplanung

Tabelle 7.1: Statusinformationen in der Onshore-Software

Xbee-Konfiguration

Bisher war das Einstellen des COM-Port zum Verwenden des XBee-Moduls nur direkt im Code möglich. Dies hatte zur Folge, dass der Nutzer der Onshore-Software zunächst über den Gerätemanager oder ähnliches den richtigen Port suchen musste, damit er diesen im Anschluss an der passenden Stelle im Quelltext einpflegen konnte. Um diesen Vorgang zu optimieren, wurde der in Abb. 7.22 dargestellt Dialog implementiert. Dieser ermöglicht über ein Drop-Down-Menü die Auswahl des korrekten Ports, sodass die oben beschriebenen Einstellungen nicht mehr nötig sind.

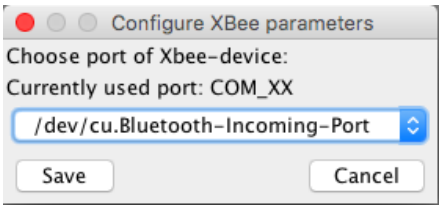


Abbildung 7.22: Dialog zum Konfigurieren des XBee-Ports

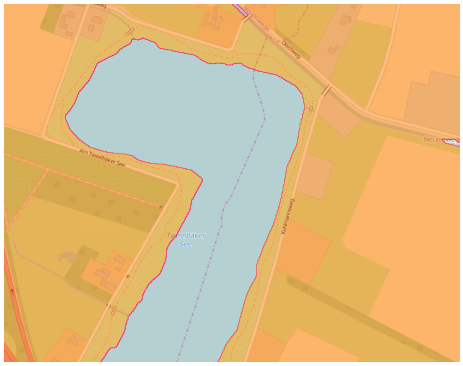
Logging

Sämtliche Softwarekomponenten verwendeten bisher einen von der vorherigen Projektgruppe implementierten Logger, um verschiedenste Meldungen zu protokollieren. Diese Implementierung wurde durch das Framework „log4j2“ ersetzt. Dies bietet folgende Vorteile: Die Logger bietet eine Vielfalt an Ausgabemöglichkeiten, wie zum Beispiel die Ausgabe über Konsole, in Datei oder in Output-Stream. Hierbei lassen sich die Ausgabemöglichkeiten einfach und zentral über eine Konfigurationsdatei realisieren. In dieser Datei gibt es außerdem umfangreiche Filterfunktionen, sodass Protokolldateien mit verschiedenem Inhalt erzeugt werden können. Unterteile wurde diese zunächst in eine Datei, die sämtliche Logs enthält, eine die nur Error-Meldungen enthält, eine die nur GPS-Informationen enthalt und eine letzte, in der Informationen

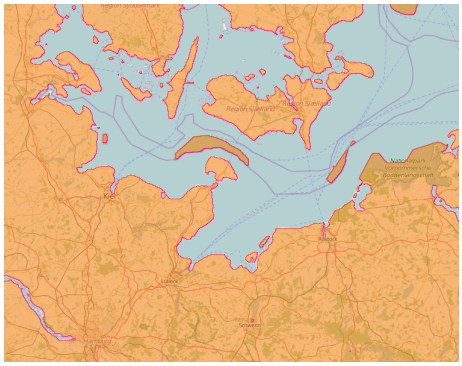
zur Routenplanung abgelegt werden. Hierbei lassen sich neben den Standard-Log-Level (`trace`, `info`, `debug`, `error` und `warn`) auch eigene Log-Level definieren. Dies war besonders im Hinblick auf das Protokollieren von den GPS- und Routen-Informationen hilfreich. Des Weiteren ist das Framework sehr performant implementiert.

Karten-Parsing

Zum Planen einer Mission ist das definieren von Gefahrenzonen wie Uferbereichen oder Stegen von besonderer Bedeutung, um einen ungestörten Betrieb zu gewährleisten. Um die Planung im Hinblick auf das Festlegen der Gefahrenzonen zu vereinfachen, ist ein Algorithmus zur automatischen Detektion von Gefahrenzonen, die durch Uferbereiche beschrieben werden, implementiert worden. Das Ziel diese Softwarekomponente ist es, aus dem verfügbaren Kartenmaterial Landbereiche zu extrahieren und diese als Gefahrenzonen zu deklarieren. Abbildung 7.23 zeigt zwei Beispiele, die der Algorithmus liefert. Abbildung 7.23(a) zeigt die detektierten Zonen am Tweelbäker See, Abbildung 7.23(b) zeigt einen Kartenausschnitt an der Ostsee. Der zweite Ausschnitt weist aufgrund der viele Inseln und komplexeren Ufergrenze einen deutlich komplizierteren Aufbau auf.



(a) Automatisch detektierte Gefahrenzone rund um den Tweelbäker See

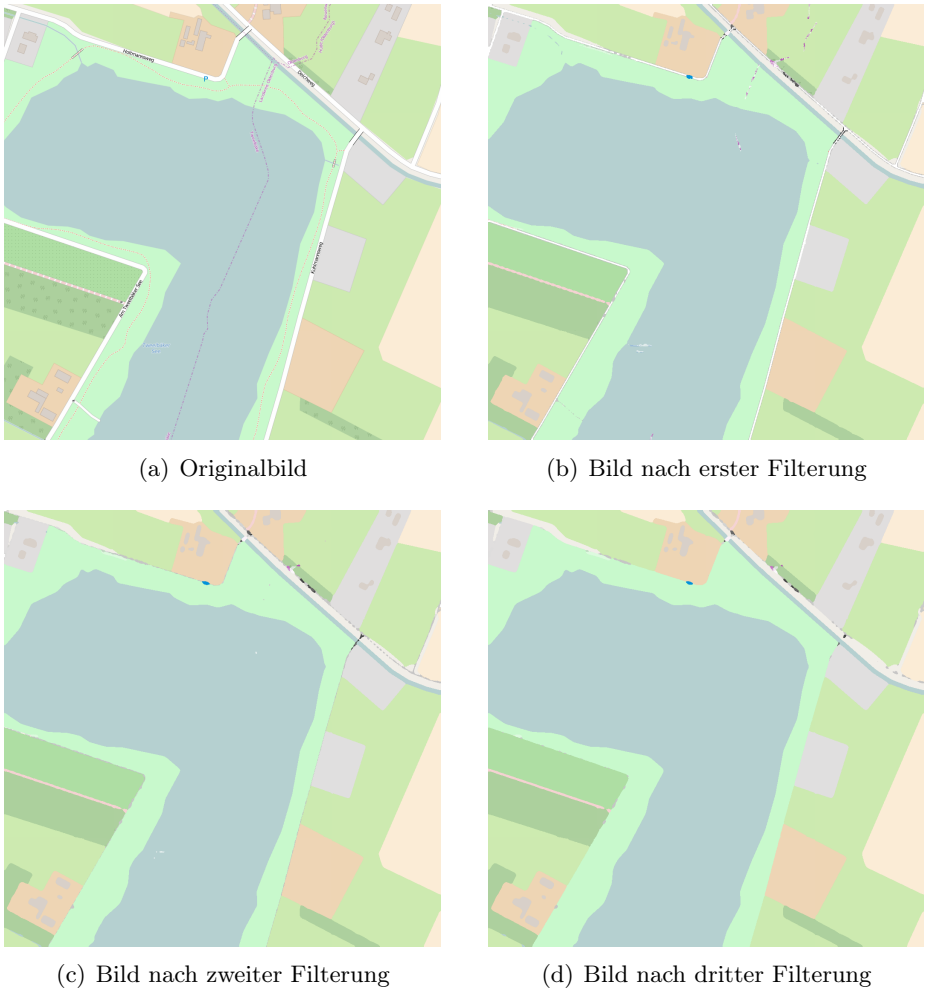


(b) Automatisch detektiere Gefahrenzonen Norddeutschland/Dänemark

Abbildung 7.23: Beispiele für automatisch detektierte Gefahrenzonen

Der Nutzer kann über einen Button in der Toolbar der Onshore-Software das Parsen des aktuell betrachteten Kartenausschnittes starten. Die verwendete `OpenStreetMap`-Schnittstelle bietet die Möglichkeit Kartenausschnitte als Bild zur Verfügung zu stellen. Da die Schnittstelle die Bilddaten über ein Kacheln basiertes Koordinatensystem verwaltet, muss der aktuell betrachtete Ausschnitt vor der Verarbeitung erst zu einem Bild zusammengesetzt werden. Nach der Zusammensetzung zu einem Gesamtbild erfolgt die mehrfache

Durchführung eines Filter-Algorithmus auf das Bild, dadurch werden Grenzlinien und Textbausteine aus der Abbildung entfernt. Abbildung 7.24 stellt die Teilergebnisse des Algorithmus chronologisch dar. Weiterhin bietet die Schnittstelle Funktionen die eine Umrechnung zwischen Pixeln und Geokoordinaten ermöglicht. Dies ist notwendig um die Gefahrenzonen im Koordinatensystem des Bildes in Geokoordinaten umrechnen zu können.



(a) Originalbild

(b) Bild nach erster Filterung

(c) Bild nach zweiter Filterung

(d) Bild nach dritter Filterung

Abbildung 7.24: Entwicklung der Bilder nach Anwenden des Filteralgorithmus

Ist die Glättung der Kartenausschnitte durchgeführt, besteht der nächste Schritt zum Extrahieren der Landzonen darin, das bisherige Bild in ein Binärbild zu überführen. Hierzu wird jeder blaue Pixel, der Wasser beschreibt, als Wert null gespeichert, alle übrigens Pixelwerte werden mit eins gespeichert.

Das Resultat dieses Schrittes ist in Abb. 7.25(a) dargestellt. Die roten Fläche sind Landbereiche, die grauen Wasser.

Der nächste Schritt zum Bestimmen der einzelnen Gefahrenzonen besteht darin, die einzelnen unabhängigen Landzonen im dem Binärbild zu finden. Ein Verfahren zum Ermitteln dieser Bereiche ist das „Connected Component Labeling“. Das Verfahren selbst wird an dieser Stelle nicht beschrieben, sondern auf die Literatur [43, S. 223 ff.] verwiesen. Das Ergebnis dieses Schrittes ist in Abb. 7.25(b) und 7.25(c) dargestellt. Während der Kartenausschnitt des Tweelbäker Sees im wesentlichen nur eine Gefahrenzone aufweist, die grün markiert ist, sieht das Bild für die Europakarte deutlich komplizierter aus. Hier sind durch verschiedene Farben die einzelnen Gefahrenzonen gekennzeichnet. Das Wasser ist in beiden Fällen mit Schwarz markiert.

Auf Basis der bestimmten Regionen in dem Kartenausschnitt ist zum Einzeichnen der Gefahrenzonen die exakte Position der Konturen der einzelnen Zonen von Interesse. Hierzu wurde ein Verfahren implementiert, das von dem Algorithmus von Tremaux [115, S. 38 ff.] inspiriert wurde. Die Idee des Algorithmus von Tremaux ist, ein Labyrinth immer derart zur durchqueren, sodass sich eine Wand des Labyrinths immer auf der rechten Seite (rechte Hand Regel) seiner aktuellen Position befindet. Alternativ kann die Wand auch auf der linken Seite der Position sein. Im folgenden wird das Labyrinth Schritt für Schritt unter Einhaltung der rechten Hand Regel durchquert, bis das Ziel erreicht ist.

Bezogen auf das vorliegenden Problem wurde der Algorithmus etwas angepasst: Zunächst wird ein Konturpunkt einer zuvor ermittelten Zone ausgewählt. Jetzt wird die Konturlinie unter Einhaltung der rechten Hand Regel durchlaufen bis der Ausgangspunkt erreicht wird. Auf diese Weise wird die gesamte Konturlinie der Gefahrenzone durchlaufen und die entsprechenden Koordianten der Kontur werden bestimmt. Beispielsweise könnte ein beliebiger Konturpunkte der grünen Fläche in Abb. 7.25(c) ausgewählt werden. Wird nun das Verfahren angewendet, so wird nach endlicher Zeit der Ausgangspunkt erreicht und die erste Gefahrenzone ist bestimmt. Analog wird mit den übrigen Gefahrenzonen verfahren. Ein Problem, welches dieses Algorithmus nicht löst, ist das Erkennen von Seen oder Meeren innerhalb einer Gefahrenzone. Dies ist in Abb. 7.25(c) in der grünen Fläche rechts oben zu sehen. Die schwarzen Bereiche in der grünen Fläche sind Seen, die vom dem aktuell verwendeten Algorithmus nicht als Wasser erkannt werden. Die Erkennung diese Wasserzonen innerhalb eines Landbereichs wurde vorerst nicht implementiert, um den Fokus in der folgenden Bearbeitungszeit stärker auf die Kollisionserkennung und -vermeidung zu legen.

Da ein Großteil der aktuell eingesetzten Verfahren von der Gruppe implementiert wurde, ist es möglicherweise eine gute Wahl einen Teil der Algorithmen in die Bibliothek `ImageJ` auszulagern. ImageJ ist eine umfangreiche Sammlung von Bildverarbeitungsalgorithmen, die in Java performant implementiert wurden. Vor allem aus Performanz sicht, ist es sinnvoll diese in einer

(a) Berechnetes Binärbild

(b) Farblich gekennzeichnete Komponenten des Binärbildes

(c) Farblich gekennzeichnete Komponenten eines komplizierten Binärbildes

Abbildung 7.25: Automatische Detektion von Gefahrenzonen

späteren Softwareversion einzusetzen. Eine Einführung in die Bibliothek und das Thema Konturenerkennung von Regionen wird in [43, S. 234 ff.] gegeben.

Abschließend ist zu sagen, dass das implementierte Verfahren zur automatischen Generierung von Gefahrenzonen seinen Zweck erfüllt. Die Implementierung weist jedoch eine geringe Performanz auf. Zusätzlich weisen die generierten Gefahrenzonen eine sehr hohe Anzahl an Eckpunkten auf. Dies beeinträchtigt den Bahnplanungsalgorithmus in einem hohen Ausmaß, sodass eine Missionssdurchführung auf dem aktuellen Zielsystem aufgrund der geringen Rechenleistung praktisch nicht durchführbar ist. Neben der Performanzoptimierung der Berechnung der Gefahrenzonen müsste in der Zukunft auch noch eine Optimierung der Gefahrenzonen stattfinden, durch die die berechneten Eckpunkte verringert werden.

7.2.3 Laserscanner Viewer

Der Laserscanner Viewer ist ein in Java entwickeltes Tool zur Darstellung der aktuellen Scandaten, während des Messvorgangs. Abbildung 7.26 zeigt die Benutzeroberfläche. In der Toolbar können IP-Adresse, sowie der Port des Hosts angegeben werden. Nachdem der Button für das Verbinden betätigt wurde, wird eine TCP Verbindung hergestellt. Ist dieser Vorgang erfolgreich, empfängt die Software die Rohdaten des Laserscanners und die prognostizierten Hindernisdaten der Kollisionserkennung. Die Rohdaten werden als kleine schwarze Punkte angezeigt, während die Hindernisse aus der kamerabasierten Kollisionserkennung als grüne Rechtecke dargestellt werden. Da über die Bildverarbeitung keine Entfernung geschätzt werden kann, werden die Hindernisse standartmäßig in der Drei-Meter-Zone des Laserscanner Viewers abgebildet. Die Hindernisse, die aus den Daten des Laserscanners abgeleitet werden, werden als rote Rechtecke dargestellt. Über den Slider lässt sich die Distanz einstellen, in welcher die Hindernisse berücksichtigt werden sollen. Falls keine Verbindung hergestellt werden kann, wird dies dem Nutzer in der Statusleiste angezeigt. Die Verbindung zum Host lässt sich jederzeit trennen.

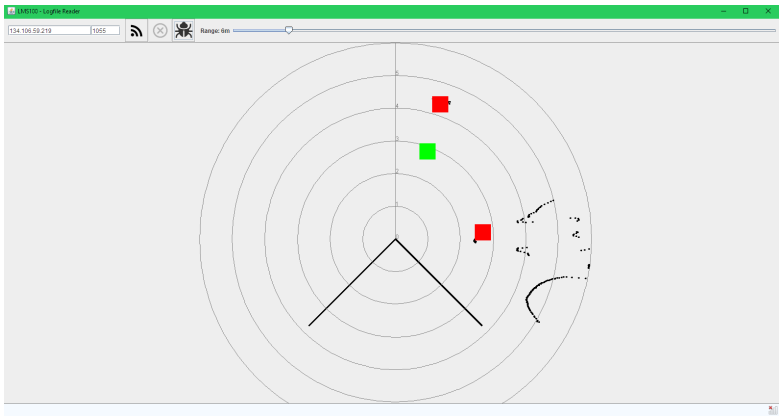


Abbildung 7.26: Benutzeroberfläche des Laserscanner Viewers

7.2.4 Onboard-Software

Die Onboard-Software teilt sich auf in Missionsplanung und die Kollisionserkennung auf. Für die Kollisionserkennung wurde eine eigene C++-Applikation entwickelt, die auf einem separatem RaspberryPi läuft.

Onboard-Missionsplanung-Software

Die Software die für die Missionsplanung und Durchführung zuständig ist wurde vollständig von der Vorgängergruppe (vgl. [31, S. 22ff]) übernommen. Deswegen werden im Folgenden nur die Veränderungen der Software beschrieben.

TCPCommunicationComponent Um Informationen über Hindernisse zu erhalten wurde dem System die IO-Komponente `TCPCommunicationComponent` hinzugefügt. Diese regelt den Verbindungsaufbau mit der Onboard-CD-Software und nimmt die TCP-Pakete entgegen die Hindernissinformationen enthalten. In der `init`-Methode wird der TCP-Client initialisiert und ein Listener für die Informationen initialisiert. Im Listener werden die empfangenen Hindernisse als JSON-String empfangen und in `Obstacle`-Objekte umgewandelt. Diese werden dann zwischengespeichert und in der `update`-Methode weiterverarbeitet. Die Weiterverarbeitung ist nötig, da die Hindernisse relative Positionsinformationen zum Boot besitzen. Diese müssen in absolute Geokoordinaten konvertiert werden. Über die aktuelle Position und Ausrichtung des Bootes zusammen mit den Informationen zu Winkel, Distanz und abgeschätzter Größe des Hindernisses wird eine rechteckige Gefahrenzone generiert, die dann für die Kollisionsverhütung verwendet werden kann.

Anpassung des Missioncontrollers Der bestehende Missioncontroller wurde um den Zustand `Found Obstacle` erweitert. Dieser wird betreten, sobald eine neue dynamische Gefahrenzone dem A* Graphen zur Bahnplanung hinzugefügt wurde. Abb. 7.27 stellt den erweiterten Zustandsautomaten für den `Missioncontroller` dar. Der `Found Obstacle` Zustand wird nur dann betreten, wenn das Boot eine Mission besitzt und diese durchführt.

Onboard-CD-Software

Bei der Onboard-CD-Software (CD: collision detection) handelt es sich um einen vollständig neu entwickelten Softwareteil. Die primäre Aufgabe der Software besteht darin, Hindernisse in der Umgebung des Wasserfahrzeugs zu erkennen. Hierfür hat die Software Zugriff auf eine Kamera und einen Laserscanner. Geschrieben wurde die Software in C++ und wird auf einem Raspberry Pi, der sich auf dem Wasserfahrzeug befindet, ausgeführt. Der Rechner ist mit dem Access-Point verbunden, sodass über das TCP-Protokoll Daten ausgetauscht werden können.

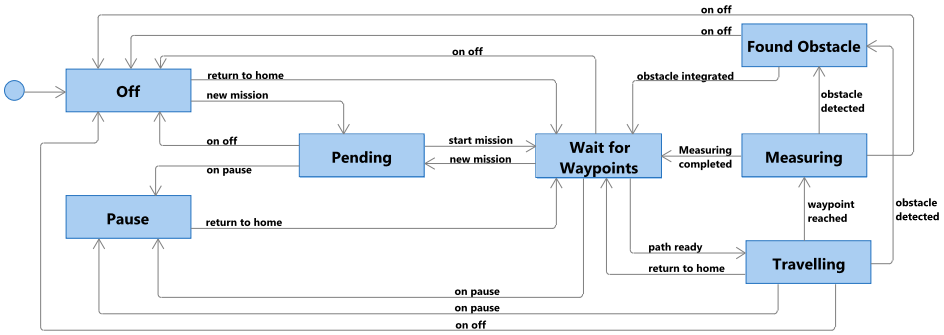


Abbildung 7.27: Zustandsdiagramm des Missioncontrollers

Abbildung 7.28 zeigt ein Aktivitätsdiagramm der Onboard-CD-Software. An diesem wird im folgenden die Funktion erklärt.

Nach dem Start des Programms finden zunächst verschiedene Initialisierungsvorgänge statt. Neben einigen Datenstrukturen werden die Kamera, der Laserscanner und eine Klasse für die TCP-Kommunikation initialisiert. Hierbei werden einige für die Einstellung notwendigen Parameter benötigt: Der Laserscanner ist per Lan mit dem Raspberry Pi verbunden, sodass die IP-Adresse des Scanners aus einer JSON-Konfigurationsdatei gelesen wird. Eine Beispiel JSON-Datei ist in Quellcode 7.1 abgebildet und wird näher in Kapitel 7.2.5 beschrieben. Darüber hinaus ist für die TCP-Kommunikation ein Port notwendig, dieser wird ebenfalls über die JSON-Datei eingelesen.

Mit dem nächsten Schritt beginnen zwei parallele Vorgänge. Die Laserrohdaten werden in einem eigenen Thread Permanent ausgelesen, sodass für die Berechnungen der Kollisionserkennung immer aktuelle Daten verfügbar sind. Parallel dazu findet die Detektion von Hindernissen statt. Zunächst wird ein neues Kamerabild gelesen und verarbeitet, wird gemäß der entwickelten Algorithmen (vgl. Kapitel 7.3.1) ein oder mehrere Hindernisse erkannte, so werden diese in einem entsprechenden Array bereitgestellt.

Als nächstes wird die Kollisionserkennung per Laser durchgeführt. Diese wird gemäß der Ausführungen in Kapitel 7.3.2 berechnet. Analog zu oben werden die Hindernisse in einem Array gesammelt. Nachdem die Hindernisdetektion abgeschlossen ist, werden die Informationen bzgl. Hindernissen und auch die Rohdaten des Laserscanner an die verbundenen Clients gesendet.

Implementierung Der folgende Abschnitt geht auf einige wichtige Hinweise bzgl. der Implementierung der Onshore-CD-Software ein. Bei der Verwendung des Laserscanners wurde zunächst ein Treiber aus dem „Mobile Robot Programming Toolkit“ (MRPT) verwendet [77]. Bei MRPT handelt es sich um eine umfangreiche Sammlung verschiedene Treiber und Computer Vision

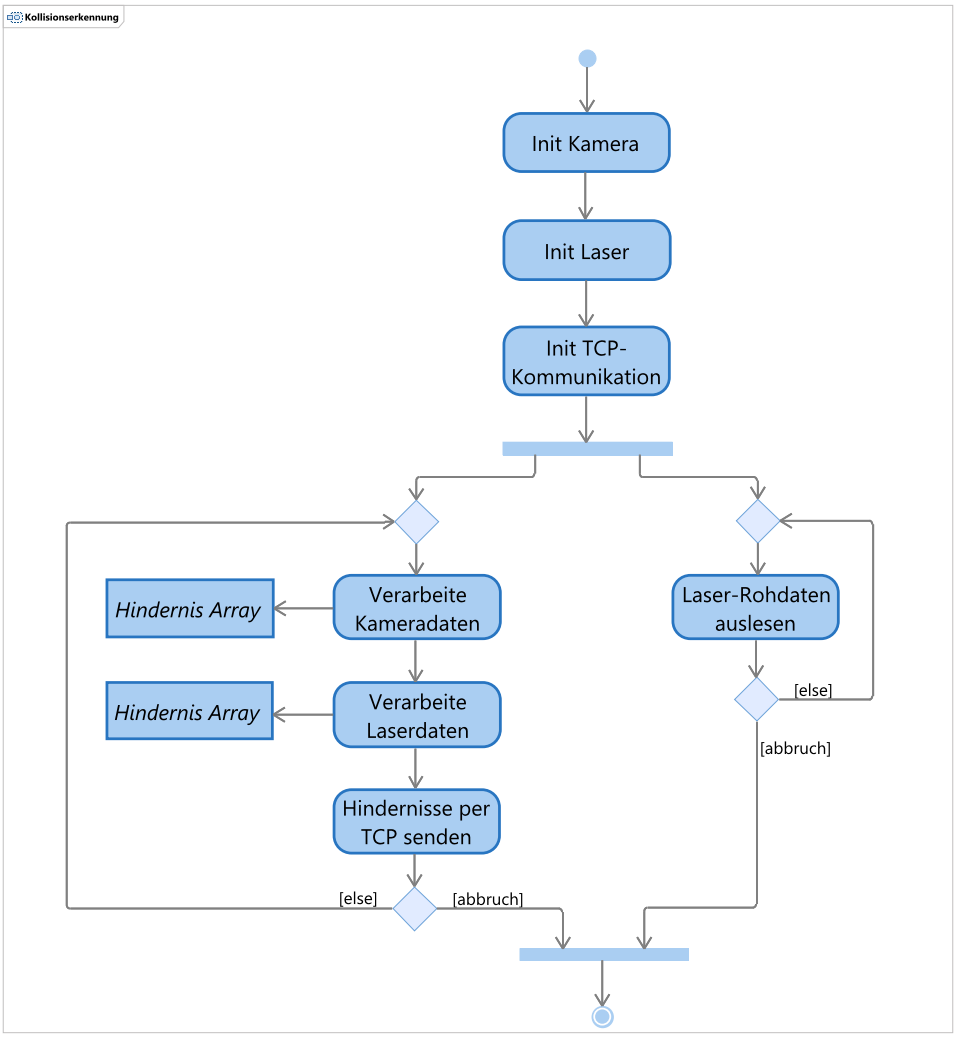


Abbildung 7.28: Aktivitätsdiagramm der Onboard-CD-Software

Algorithmen für die Entwicklung in der Robotik. Zum Verwenden des Treiber war ein installieren der vollständigen MRPT-Bibliothek nötig. Dies war zwar auf einem Laptop ohne weiteres möglich, allerdings zeigten sich beim Installieren auf dem Raspberry Pi enormen Probleme, sodass eine Alternative Lösung verwendet werden musste.

Eine weitere Möglichkeit einen bestehenden Treiber zu verwenden, war die Nutung des „Robot Operating Systems“. Hierbei handelt es sich um ein Robotik-Framework, dass diverse Treiber und Algorithmen bereitstellt. Allerdings wurde dieses mal entschieden, die relevanten Treiber-Klassen aus dem Framework zu extrahieren und in C++-Code zu migrieren [97]. Dies sparte

zum einen das Kompilieren der gesamten Bibliothek und der Codeumfang konnte deutlich reduziert werden, sodass ein kompilieren auf dem Raspberry Pi problemlos möglich wurde.

Hilfreich für das Konfigurieren und Einstellen von verschiedenen Parameters der Onboard-CD-Software war das verwenden einer Konfigurationsdaten. Diese ist in Quellcode 7.1 dargestellt. Neben verschiedenen Parametern für die TCP-Kommunikation, lassen sich auch die Kamera oder das Lidarsystem in der Verwendung abschalten. Dies hatte beim Test vor allem praktische Gründe, wenn es nur darum ging eine der Komponenten zu testen. Des Weiteren finden sich in der Datei Einstellungen zur Kamera oder zum Lidar. Vorteil des Verwenden einer Konfigurationsdatei ist, dass das Programm beim Ändern von Einstellungen nicht neu kompiliert werden muss. Für eine einfache Verwendung innerhalb der C++-Software wurde die Datei im JSON-Format angelegt.

```
{
   "lidar_adress":"192.168.1.55",
   "lidar_occupancy_cutoff":5,
   "lidar_port":2111,
   "lidar_scale_factor":10,
   "lidar_visibility_range":1000,
   "software_run_count":21,
   "software_save_camera_image":1,
   "software_show_camera_image":1,
   "software_use_camera":1,
   "software_use_lidar":1,
   "tcp_port_viewer":1055,
   "tcp_port_onshore":1056,
   "tcp_viewer_send_intervall_millis":100,
   "tcp_onshore_send_intervall_millis":2000
}
```

Quellcode 7.1: Konfigurationsdatei für die Kollisionserkennung

Die Entscheidung diesen Softwareteil in C++ zu schreiben, ist auf verschiedene Gründe zurückzuführen: Da sich intensiv mit Bildverarbeitung beschäftigt wurde, war das Verwenden von OpenCV naheliegend. Da OpenCV in C/C++ geschrieben ist, lässt sich dies sehr leicht verwenden. Des Weiteren kann mit Hilfe der Bibliothek `Boost` relativ einfach ein TCP-Kommunikationskonzept umgesetzt werden. Auch dieses Framework ist in C++ verfügbar. Die verwendeten Bibliotheken werden im folgenden Abschnitt näher erläutert. Ein weiterer Grund, C++ zu verwenden besteht in der Performanz, da die Sprache deutlich hardwarenäher ist als beispielsweise Java.

Verwendete Bibliotheken Für die Entwicklung in C++ wurden verschiedene Bibliotheken verwendet, die insbesondere die Bildverarbeitung enorm

erleichtert haben. Diese werden im folgenden in einem kurzen Überblick vorgestellt.

OpenCV OpenCV ist eine Computer-Vision Bibliothek, deren Fokus auf der Implementierung von Bildverarbeitungsalgorithmen liegen. Durch eine sehr gute Dokumentation und viele Beispiele ist die Verwendung schnell erlernbar und war innerhalb der Projektgruppe in der kamerabasierten Kollisionserkennung von großem Nutzen.

Boost Die Bibliothek Boost ist eine umfangreiche Sammlung von Algorithmen aus diversen Bereichen. Innerhalb der Projektgruppe war besonders das Paket zur TCP-Kommunikation von Interesse. Außerdem wurden die Zeitfunktionen benutzt, um Laufzeitmessungen durchzuführen. Darüber hinaus sind aber auch Graphalgorithmen, Sortieralgorithmen oder ähnliches enthalten.

CMake Zum Kompilieren von C++-Programmen hat sich CMake etabliert. Über eine Konfigurationsdatei können verschiedene Bibliotheken eingebunden und eine ausführbare Datei erstellt werden. Hierbei werden dem Nutzer auch die Befehle für das Linken und Zusammenführen von verschiedene Quelldateien abgenommen.

7.2.5 Kommunikation

Das Wasserfahrzeug nutzt unterschiedliche Kommunikationkanäle. Zum einen wird eine Datenübertragung per XBee verwendet, um Missionsdaten zwischen Onboard- und Onshore-Software auszutauschen. Dies wurde bis auf kleine Anpassungen von der vorangegangen Projektgruppen umgesetzt, daher wird an dieser Stelle auf weitere Details verzichtet. Zum anderen wurde eine TCP-Kommunikation implementiert die der Kollisionserkennung und -verhütung dient. Im Folgenden wird die TCP-Kommunikation beschrieben, sowie die Veränderungen an der bestehende XBee-Kommunikation.

Auf dem Wasserfahrzeug selbst sind inzwischen zwei Raspberry Pis im Einsatz, die unterschiedliche Aufgaben bearbeiten:

1. Der erste Rechner (Onboard-Missionsplanung) übernimmt, wie aus der vorherigen Projektgruppen bekannt, die Steuerung der Mission. Hierzu gehören das Berechnen einer Route, das Ansteuern der Motoren und der Datenaustausch mit der Onshore-Software per XBee.

2. Der zweite Pi (Onboard-Kollisionserkennung) wurde von der aktuellen Projektgruppe ergänzt und ist für die Kollisionserkennung zuständig. Hierzu sind eine Kamera und ein Laserscanner mit dem Rechner verbunden, um Hindernisse in der Nähe des Wasserfahrzeugs zu detektieren.

Abbildung 7.29: Montierter Access-Point

Die Informationen über die erkannten Hindernisse müssen für eine Kollisionsvermeidung an den ersten Rechner weitergereicht werden, damit dieser gegebenenfalls eine Kurskorrektur durchführen kann. Hierfür ist eine geeignetes Kommunikationssystem notwendig. Umgesetzt wurde dies mit Hilfe eine TP-Link TL-WR841N Access-Points. Dieser ist in Abb. 7.29 dargestellt. Die beiden Rechner können sich per integriertem WLAN-Modul mit dem Access-Point verbinden und über das TCP-Protokoll Informationen austauschen. Der folgende Abschnitt gibt einen detaillierten Einblick in das umgesetzte Kommunikationsverfahren und die Vorteile, die sich aus dem Verwenden von TCP ergeben.

Architektur

Der Datenfluss zum Austauschen der Laserrohdaten und der erkannten Hindernisse erfolgt über den in Abb. 7.29 dargestellten Access-Point per WLAN. Auf Basis des TCP-Protokoll hat sich die in Abb. 7.30 dargestellte Kommunikationsarchitektur ergeben.

Die Abb. 7.30 enthält insgesamt vier Teilnehmer, die Informationen austauschen. Diese vier Teilnehmer entsprechen dabei Programmen die jeweils auf unterschiedlichen Rechnern ausgeführt werden. Die beiden oberen Kästchen sind die Onboard-Software und die Onboard-CD-Software, die jeweils

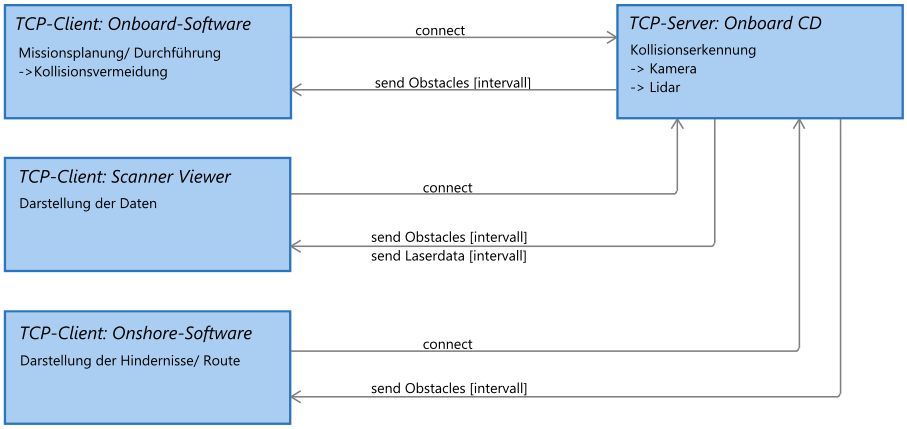


Abbildung 7.30: Architektur der TCP-Kommunikation

auf einem eigenen Raspberry Pi ausgeführt werden. Das Kästchen in der zweiten Reihen beschreibt den Scanner-Viewer. Dieser kann auf einem beliebigen Rechner ausgeführt werden. Hierbei handelt es sich üblicherweise um einen Rechner, der am Ufer positioniert ist, sodass über den Viewer die Kollisionserkennung überwacht werden kann. Dies hat sich vor allem während der Testphase als sehr praktisch herausgestellt. Als letzter Teilnehmer kann sich die Onshore-Software mit dem TCP-Server verbinden. Diese empfängt im Anschluss die detektierten Hindernisse, sodass diese in die Karte der Onshore-Software eingezeichnet werden können. Auf diese Weise kann auch innerhalb der Missionskontrolle die Hindernisserkennung überwacht werden.

Die Architektur lehnt sich an ein Server-Client-Prinzip an. Hierbei wird innerhalb der Onboard-CD-Software ein TCP-Server gestartet. Dieser lässt die Verbindung beliebig viele Clients zu. Über eine entsprechende Connect-Methode können sich die informationsinteressierten Clients verbinden. Der Server ist dabei so ausgelegt, dass in einem vorgegebenen Zeitintervall die Informationen periodisch gesendet werden.

Das Festlegen der nötigen Parameter wie Port und Sendeintervall erfolgt über eine JSON-Datei. Diese ist in Quellcode 7.1 dargestellt. Grund für das Einführen einer JSON-Datei zum Konfigurieren war, dass beim Ändern von Parametern kein erneutes Kompilieren des vollständigen Programms notwendig ist. Insbesondere auf dem Raspberry Pi ist die Kompilierzeit für die Onboard-CD-Sofware sehr lang (ca. 40 Sekunden).

Die hier vorgestellte Architektur bietet verschiedene Vorteile: Bei einem Datenaustausch via TCP lassen sich softwareseitig verschiedene Bibliotheken verwenden, die sowohl in Verwendbarkeit als auch in der Umsetzung sehr gut nutzbar sind. Des Weiteren existieren sowohl für Java als auch für C++ die

entsprechenden Bibliotheken. Da der Viewer in Java und die Onboard-CD-Software in C++ geschrieben sind, war dies an dieser Stelle sehr praktisch. Hierin ist ein weiterer Vorteil für die Verwendung von TCP enthalten, denn ein Datenaustausch ist aufgrund der vorhandenen Bibliotheken zwischen unterschiedlichen Programmiersprachen sehr einfach möglich. Verglichen mit der Nutzung von XBee, was weiterhin für die Übertragung der Missionsdaten eingesetzt wird, ist die Verwendung von TCP in Software deutlich leichter.

Bei der Verwendung des Access-Points stellte sich heraus, dass die Reichweite auf dem See für Experimente absolut ausreichend ist. Der Scanner-Viewer, welcher üblichweise am Ufer auf einem Rechner ausgeführt wurde, konnte ohne weiteres verwendet werden, um die Kollisionserkennung zu überwachen. Ein Übertragungsintervall von 100 ms war problemlos möglich.

Datenpakete

Die Datenpakete zum Austauschen der Laserrohdaten und der Hindernisinformationen wurden in einer JSON-Datei organisiert. Quellcode 7.2 zeigt einen Ausschnitt einer solchen Datei.

Unter `laserRaw` ist ein Array von Laserdaten abgelegt. Die Daten werden hierbei in Millimeter übertragen. Unter der Kennzeichnung `obstacles` werden die Hindernisinformationen aufgeführt. Diese setzen sich aus dem Winkel unter dem ein Hindernis zu sehen ist, der Distanz, der geschätzten Größe und der Quelle zusammen. Bei der Quelle kann es sich entweder um das Lidar oder die Kamera handeln. Da die Kollisionserkennung per Kamera keine Distanz zum Objekt und keine geschätzte Größe liefert, wird hier als Standardwert -1 übertragen.

```
{
   "laserRaw": [123, 123, 125, ... , 312, 312, 321, 321],
   "obstacles": [{
      "angle": 92.25,
      "distance": 3,
      "estimatedSize": 10,
      "source": "lidar"
   }],
   "obstacles": [{
      "angle": 122.25,
      "distance": -1,
      "estimatedSize": -1,
      "source": "camera"
   }]
   ...
}
```

Quellcode 7.2: Beispiel eines Datenpakets im JSON-Format

Implementierung

Die Implementierung zwischen der Onboard-CD-Software und dem Scanner-Viewer unterscheidet sich zunächst in der verwendeten Programmiersprache. Die Onboard-CD-Teil ist in C++ geschrieben.

Für die TCP-Nutzung wurde innerhalb der C++-Software die Bibliothek Boost verwendet. Insbesondere wurde hierbei das Paket ASIO genutzt [69]. Dieses stellt eine Server-Implementierung, die in einem eigenen Thread die Verarbeitung der Daten verwaltet, bereit. Diese musste allerdings um eine Verwaltung der verbundenen Clients erweitert werden. Der zu nutzende Port lässt sich über die JSON-Datei in 7.1 einstellen. Ist der Server innerhalb der Software initialisiert worden, wird an alle verbundenen Clients in dem vorgegebenen Intervall eine Nachricht mit den Laserdaten und Hindernisinformationen gesandt.

Die Client seitige Implementierung der TCP-Kommunikation ist in Java geschrieben, da sämtliche Clients Java Applikationen sind. Für die Implementierung wurden non-blocking Sockets aus der Java-NIO API verwendet. Grundlage der Implementierung ist die `CommunicationClient` Klasse. Über einen `Builder` kann eine Instanz des Clients mit einem Host und Port definiert werden. Nach der Initialisierung kann über dem Client ein Listener des Typs `CommunicationClient` übergeben werden. Über diesen Listener werden die Applikationen über ankommende Nachrichten sowie den Verbindungszustand zum Server benachrichtigt. Die empfangenen JSON-Strings werden mithilfe der JSON-Java-Bibliothek geparst und zu Java-Objekten konvertiert.

Zusätzlich wurde ein Java seitiger Server implementiert. Dieser dient der Simulation der Kollisionserkennung, um die Kollisionsverhütung testen zu können. Die Applikation ist eine einfache Konsolenanwendung die es über Eingaben in der Konsole ermöglicht Hindernisse an die verbundenen Clients zu schicken.

Xbee-Anpassungen

Die Stabilität der XBee-Verbindung konnte im Vergleich zur vorherigen Projektgruppe verbessert werden. Ein Problem war, dass für das Versenden eines Datenpaketes von der Onshore-Software aus wurde ein neuer loser Thread geöffnet. Dies führte dazu, dass es Pakete in einer falschen Reihenfolge versendet wurden, woraufhin dann die Kommunikation fehlerhafte Ergebnisse lieferte. Dies wurde durch die Verwendung eines `ListeningExecutorService` mit einem Threadpool, welcher maximal einen Thread öffnen kann behoben. Dadurch wird abgesichert, dass die versendeten Pakete in chronologischer Reihenfolge versendet werden.

7.3 Kollisionserkennung

Das Erkennen von Objekten, die zu einer Kollision mit dem Boot führen könnten, war einer der Schwerpunkte in der Projektgruppe. Im folgenden Kapitel wird beschrieben, wie die Kollisionserkennung umgesetzt wurde. Hierbei wurde versucht über zwei Arten Hindernisse zu erkennen. Zum einen wurde auf Kameras zurückgegriffen und über verschiedene Ansätze versucht die Bildinformationen so zu verarbeiten, dass Objekte herausgefiltert werden können. Auf der anderen Seite wurde auch ein Laserscanner genutzt, um den Abstand zu Objekten im Sichtfeld des Bootes zu messen und gegebenenfalls einzelne Messpunkte zu einem möglichen Hindernis zusammenzufassen.

7.3.1 Kamerabasierte Kollisionserkennung

Objekte, die zu einer Kollision mit dem Boot führen, tauchen früher oder später im Sichtfeld des Bootes auf. Mit angebrachten Kameras lässt sich das Umfeld des Wasserfahrzeugs beobachten. Das entstandene Bildmaterial muss im Bezug auf eine mögliche Kollision analysiert werden. Im folgenden Abschnitt wird auf die Realisierung der Objekterkennung mittels Kameras eingegangen. Dabei werden neben der Hardware auch die verwendeten Algorithmen und die softwareseitige Implementierung erläutert.

Hardware

Um das Sichtfeld vor dem Wasserfahrzeug mit einer Kamera zu erfassen, muss zunächst ein geeignetes Modell ausgewählt werden. Die Projektgruppe hat sich hierbei für eine Logitech C920 entschieden. Bei dieser Kamera handelt es sich um eine 15-Megapixel-Webcam, die sich problemlos mit dem Raspberry Pi verbinden lässt. Darüber hinaus waren die Anschaffungskosten mit 72.90 Euro relativ gering. Mit der Kamera lässt sich ein Sichtfeld von 95° erfassen. Dies deckt zwar keine Rundumsicht um das Wasserfahrzeug ab, allerdings ist das Anschaffen weiterer Kameras aufgrund des geringen Preises denkbar. Letztendlich hat die Projektgruppe aufgrund der geringen Investitionskosten und der guten Kompatibilität zur bereits vorhandenen Hardware das Modell ausgewählt, um eine kamerabasierte Kollisionserkennung zu erproben.

Im Folgenden werden in den Abbildungen auch Bilder, die mit einer GoPro aufgezeichnet wurden, verwendet. Dies hat den Grund, dass die GoPro bereits zur Anfangsphase des Projektes vorhanden war, sodass hiermit gleich zu Beginn erste Aufnahmen von Testfahrten gesammelt wurden. Die im folgenden vorgestellten Algorithmen arbeiten sowohl auf dem Bildmaterial der GoPro als auch der Logitech Webcam.

Idee

Aus einer ersten Analyse der Rohbilddaten wurde die Grundidee der kamerabasierten Kollisionserkennung abgeleitet.

Abbildung 7.31: Beispielhaftes Ausgangsbild

Wie in Abb. 7.31 zu sehen, ist der Horizont des Bildes gekrümmt. Dies soll über eine Kamerakalibrierung zunächst korrigiert werden, sodass eine einfache Extraktion des Horizontes möglich wird. Der erkannte Horizont soll im Anschluss genutzt werden, um die Wasseroberfläche aus dem übrigen Bild herauszuschneiden. Dies hat den Grund, dass die Projektgruppe vereinfacht angenommen hat, dass für eine Kollision relevante Objekte zunächst nur auf der Wasseroberfläche erscheinen. Des Weiteren fällt auf, dass in den Rohbilddaten die Wasseroberfläche stark reflektiert. Dies soll mit geeigneten Algorithmen reduziert werden, um eventuelle Falscherkennungen minimieren zu können. Die bisher vorgestellten Teilschritte der Objekterkennung lassen sich als eine Art Vorverarbeitung auffassen. Abbildung 7.32 fasst die einzelnen Schritte in einer Prozesskette zusammen und kennzeichnet die bisher erwähnte Vorverarbeitung.

Auf Grundlage des bis hier gewonnenen Bildmaterials der Wasseroberfläche wird jetzt die Objekterkennung durchgeführt (vgl. Abb. 7.32). Hierbei hat die Projektgruppe zum einen den Canny-Edge-Ansatz erarbeitet und implementiert und einen zweiten histogrammbasierten Ansatz als eine weitere Möglichkeit der Objekterkennung vorgestellt. Die zweite Möglichkeit wurde nicht implementiert. Die erkannten Objekt werden in einem Obstacle-Array gesammelt und an die Onboard-Software zur Weiterverwendung gesendet. Die folgenden Abbschnitte beschreiben detailliert die in Abb. 7.32 dargestellte Prozesskette und gehen auf die Implementierung ein. Außerdem werden die zwei Varianten zur Objekterkennung vorgestellt.

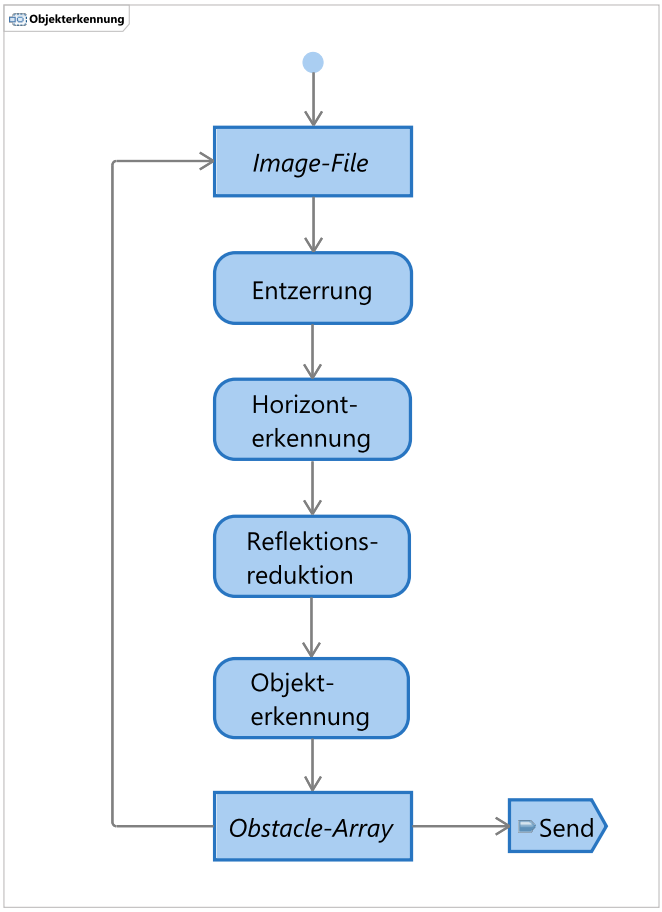


Abbildung 7.32: Ablauf der Kollisionserkennung auf Kamerabasis

Horizonterkennung

Die in Abb. 7.33 verwendete Kamera ist an der Vorderseite des Wasserfahrzeugs befestigt, um eine Objekterkennung in Fahrtrichtung zu ermöglichen. In der Abb. ist deutlich der Horizont, der das Wasser von der übrigen Umgebung trennt, zu erkennen. Um die Objekterkennung zunächst nur auf die Wasseroberfläche zu beschränken, ist die Idee den Horizont automatisch aus den Bilddaten zu extrahieren, sodass im Anschluss nur der untere Teil des Bildes für die Hindernisserkennung betrachtet werden muss. Des Weiteren reduziert sich der Rechenaufwand bei der Analyse des Bildausschnittes immens im Vergleich zur Analyse des gesamten Bildes.

In Abb. 7.33 ist zu erkennen, dass der Horizont eine sehr scharfe Trennlinie zwischen Wasser und Bäumen aufweist. Diese soll mit Hilfe morphologischer Operationen und dem Hough-Transformator extrahiert werden.

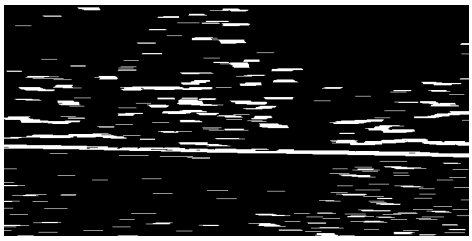
Abbildung 7.33: Beispielaufnahme für die Horizonterkennung

Der erste Schritt zum Ermitteln der Horizontlinie besteht darin, das Bild in ein Grauwertbild zu konvertieren, da viele der verwendeten OpenCV auf Grauwertbildern arbeiten. Als nächstes werden horizontale Linien mit dem in Abschnitt 5.2.2 beschriebenen morphologischen Filtern ermittelt. Das Resultat ist in Abb. 7.34(a) dargestellt. Auffällig an diesem Zwischenergebnis ist, dass das Bild eine enorme Menge an detektierten Linie aufweist. Auf Anhieb ist die Horizontlinie offensichtlich nicht zu detektieren. Des Weiteren fällt auf, dass die Horizontlinie deutlich durch einen weiße, fast durchgezogene Linie zu erkennen ist.

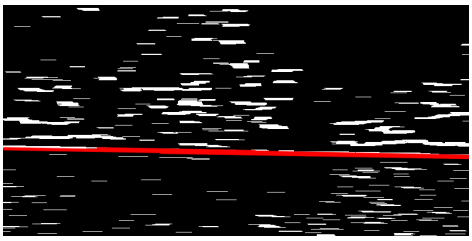
Der nächste Schritt besteht nun darin, die weiße Linie aus dem Binärbild zu ermitteln. Hierfür wird ein weiterer Liniendetektor verwendet, der Hough-Line-Transformator. Dieser wird ausführlich in [91] beschrieben. Der Vorteil dieses Detektors besteht darin, dass dieser Linien in sämtlichen Orientierungen und Längen erkennt. Das in Abschnitt 5.2.2 vorgestellte Verfahren hat im Gegensatz dazu nur Linien erkannt, die eine Länge von sechs Pixeln hatten und horizontal ausgerichtet waren. Durch Anwenden des Detektors und Einzeichnen der detektierten Linien wird das in Abb. 7.34(b) dargestellt Resultat erzielt. Hierbei ist zu bemerken, dass die in rot eingezeichneten Linien bereits gefiltert sind.

Da es sich bei den in Abb. 7.34(b) noch um viele einzelnen Linie handelt, ist es notwendig eine Ausgleichsgerade durch die Geradenschar zu legen. Hierfür eignet sich die Funktion „cv::fitline(...)“ von OpenCV, die eine entsprechende Gerade berechnet. Eine Beschreibung zu der entsprechenden Funktion findet sich in [92]. Das Ergebnis und die eingezeichnete Ausgleichsgerade sind in Abb. 7.34(c) dargestellt.

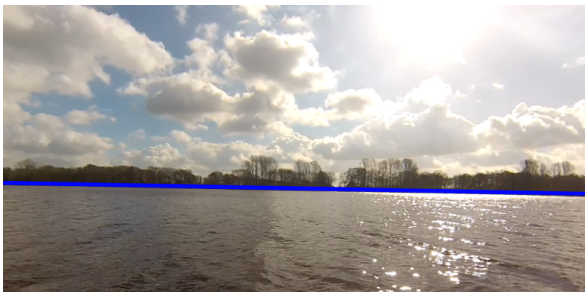
Abbildung 7.35 zeigt weitere Beispielaufnahmen, in denen der Horizont erkannt wurde. In der ersten Abbildung ist zu erkennen, dass der Horizont durch das links in Erscheinung tretende Ufer unterbrochen wird. In dieser Situation wird die Ausgleichsgerade durch das gesamte Bild gelegt. Im zwei-



(a) Anwenden von morphologischen Operationen



(b) Bestimmen von Hough Lines



(c) Berechnete Horizontlinie im Originalbild

Abbildung 7.34: Teilschritte der Horizonterkennung

ten Bild wird die Ufergrenze als Horizont erkannt. Dies ist ebenfalls ein gut verwertbares Ergebnis, denn alles südlich der erkannten Linie kann zur Objekterkennung verwendet werden.

Mit Hilfe des entwickelten Algorithmus lässt sich der Horizont zuverlässig erkennen, sodass im nächsten Schritt der Bereich südlich der Horizontlinie ausgeschnitten werden kann, um auf diesem Bildausschnitt die Objekterkennung durchzuführen. Ein ausgeschnittener Bildabschnitt ist in Abb. 7.36(a) dargestellt.

Reflexionsreduktion

Abbildung 7.36(a) zeigt eine Aufnahme des Sees, welche mit der Frontkamera des Wasserfahrzeugs aufgenommen wurde. Offensichtlich kommt es an der Wasseroberfläche zu starken Reflexionserscheinungen. Diese können bei der Objekterkennung dazu führen, dass an den entsprechenden hellen Stellen Fehlinterpretationen auftreten und die Bereiche als Hindernis erkannt werden. Daher ist die Idee, die Reflexion nachträglich mit geeigneten Verfahren zu reduzieren. Abbildung 7.36(b) zeigt das Ergebnis, dass der entworfene Algorithmus liefert. Es ist eine deutliche Reduktion der Reflexion zu erkennen. Der folgende Abschnitt erläutert die Idee des implementierten Algorithmus.

(a) Horizonterkennung mit Uferabschnitt

(b) Horizonterkennung am Ufer

Abbildung 7.35: Horizonterkennung in verschiedenen Situationen

Auf Basis des in Aufnahme 7.36(a) gezeigten Bildes, wird zunächst ein Schwellwertfilter (vgl. Kapitel 5.2.3) angewendet. Hierzu wird das RGB-Bild vorher in ein Grauwertbild konvertiert. Bei einem geeigneten Schwellwert werden jetzt im Grauwertbild alle Pixel, die über diesem Wert liegen, herausgefiltert. Im Beispiel wurde ein Schwellwert von $s = 190$ festgelegt. Das heißt, dass alle Pixel die einen Grauwert über 190 aufweisen, detektiert werden. Das Ergebnis dieser Operation ist in einem Binärbild in Abb. 7.37(a) dargestellt.

Das durch den Schwellwertfilter gewonnene Binärbild soll im Folgenden als Maske zum Manipulieren der Pixel im Originalbild dienen. Um die Übergänge etwas weicher zu gestalten, wird ein Dilatationsfilter (vgl. Kapitel 5.2.2) auf das Binärbild angewandt, was dazu führt, dass die weißen Pixel etwas gestreckt werden. Das Resultat der Filteroperation ist in Abb. 7.37(b) dargestellt

(a) Wasseroberfläche ohne Reflexionsreduktion

(b) Wasseroberfläche mit Reflexionsreduktion

Abbildung 7.36: Vorher/Nachher-Vergleich der Reflexionsreduktion

Das jetzt entstandene Binärbild kann in einer nächsten Operation als eine Maske verwendet werden (vgl. Kapitel 5.2.4). Hierbei werden die weißen Bereiche über das Originalbild gelegt und die darunterliegenden Pixel werden mit einem Mittelwert der umliegenden Pixel überschrieben. Dies führt dazu, dass die zuvor durch Reflexion weißen Bildbereiche den Nachbarpixeln angeglichen werden und das in Abb. 7.36(b) dargestellt Bild entsteht.

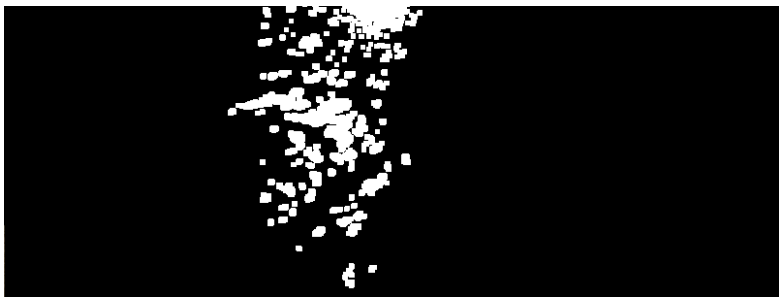
Durch Hinzufügen dieses Algorithmus kommt es deutlich seltener zu Fehlerinterpretationen bei der Objekterkennung und die Zuverlässigkeit des Systems kann deutlich erhöht werden.

Canny-Edge basierte Objekterkennung

Auf Basis des im vorherigen Abschnitt berechneten Bildes mit reduzierter Reflexion findet in dem folgenden Schritt das Detektieren von Objekten statt. Hierfür wird der Canny-Edge-Algorithmus verwendet. Grundlage für die Implementierung sind die Quellen [44] und [87].

Der Canny-Edge-Algorithmus wurde von John Francis Canny entwickelt, mit dem Ziel Kanten in Bildern zu ermitteln. Bezogen auf den See ergibt sich folgende Idee: Die Wasseroberfläche (vgl. Abb.7.38(c)) weist in der Regel wenige Kanten auf, sondern hat eher weiche Verläufe, die sich farblich sehr

(a) Anwenden eines Schwellwertfilters auf das Originalbild



(b) Strecken der weißen Pixel mit Hilfe eines Dilatationsfilters

Abbildung 7.37: Operationen zur Reflexionsreduktion

ähneln. Das rote Boot hingegen ist durch einige Kanten geprägt. Mit Hilfe des Canny-Edge-Verfahrens sollen diese Kanten im Folgenden bestimmt werden.

Der Algorithmus selbst wird an dieser Stelle nicht im Detail beschrieben, sondern es wird nur die Idee anhand eines Beispiels erklärt. Detailliert kann die Beschreibung des Verfahrens in [44] nachgelesen werden.

Grundlage des Verfahrens ist die Berechnung des Gradientenbildes vom Original. Hierbei kann ein Schwellwert festgelegt werden. Die Abbildungen 7.38(a) und 7.38(b) zeigen zwei Beispiele. Ein niedriger Schwellwert $T_{low_1} = 50$ führt zum Erkennen von deutlich mehr Kanten als ein entsprechend höherer Wert $T_{low_2} = 95$. Da ein kleiner Schwellwert zu deutlich mehr Fehlerkennungen führt, wurde der Wert innerhalb der implementierten Software auf 95 eingestellt. Dies liefert für die Objekterkennung gute Ergebnisse. Das so entstandene Bild ist in Abb. 7.38(b) dargestellt.

Aus dem Kantenbild werden jetzt Rechtecke berechnet, die ein erkanntes Objekt wie in Abb. 7.38(c) repräsentieren. Hierfür wird ein Rechteck berechnet, das eine zusammenhängende Kontur wie in Abb. 7.38(b) umschließt. Des Weiteren ist eine Berechnung des Winkels, unter dem das Objekt zum Boot steht, interessant. Da der Blickwinkel der Kamera bekannt ist und der Mittelpunkt des roten Rechtecks ebenfalls leicht bestimmt werden kann, ist die Berechnung des Winkels, den das Boot zum Objekt einnimmt, möglich.

(a) Ecken-Detektion mit $T_{low} = 50$

(b) Ecken-Detektrion mit $T_{low} = 95$

(c) Berechneter Rahmen für das erkannte Hindernis

Abbildung 7.38: Canny-Edge basierte Objekterkennung

Für die Implementierung wurde der in OpenCv bereitgestellte Canny-Edge-Algorithmus verwendet [87]. Hierbei war primär das Finden geeigneter Einstellungen für das Verfahren nötig. Das Bilden von Rechtecken aus den Kanten-Konturen wurde ebenfalls mit einigen hilfreichen OpenCv-Funktionen umgesetzt.

Die berechneten Hindernisse können jetzt per TCP an die Onboard-Software geschickt werden. Das Format, in dem die Daten verschickt werden, kann Kapitel 7.2.5 entnommen werden. Näheres zum Aufbau der Onboard-CD-Software, in der sämtliche Bildverarbeitungsalgorithmen implementiert sind, wird in Abschnitt 7.2.4 erläutert.

Das hier vorgestellte Verfahren, um auf Kamerabasis Objekte zu erkennen, weist im Vergleich zur Laser-Variante verschiedene Vor- und Nachteile auf. Diese werden intensiv in Kapitel 8.1 diskutiert.

Histogrammbasierte Objekterkennung

Ein Ansatz, der zu Beginn verfolgt wurde, war mithilfe von Histogrammen und Bildausschnitten Objekte in einem Szenario zu erkennen. Ein Bild, wel-

ches hauptsächlich eine Wasserlandschaft zeigt, besitzt charakteristische Farbmerkmale. Ein Objekt, wie beispielsweise eine Boje, weist stattdessen ganz andere Farbmerkmale auf. Die Grundidee ist nun, die Farbmerkmale im Bild zu analysieren und die Positionen möglicher Hindernisse auszumachen.

Um diesen Ansatz zu realisieren wird ein Algorithmus verwendet, der zunächst ein Histogramm der RGB-Werte für das gesamte Frame errechnet und in einem gesonderten Fenster anzeigt (Siehe Abb. 7.39(c)). Weiterhin wird per Mausklick ein Teilausschnitt von einer fest definierten Größe aus dem Ausgangsframe herausgeschnitten und angezeigt. Von diesem Auschnitt ausgehend wird nun ein weiteres Histogramm der RGB-Werte erzeugt und angezeigt (siehe Abb. 7.39(b)). Nun können die beiden Graphen miteinander verglichen werden. Der Gedanke dabei ist, dass sich ein Auschnitt mit einem Objekt, wie beispielsweise einer blauen Kiste (Siehe Abb. 7.39(b)), stark von dem Histogramm des Gesamtbildes unterscheidet.

Nach weiterer Recherche zu diesem Thema wurde eine Abhandlung gefunden, die einen ähnlichen Ansatz verfolgt [112]. Im Folgenden sollen der Workflow dieses Projektes und die Unterschiede zu dem eigens entwickelten Ansatz vorgestellt werden.

Die vorgestellte Implementierung der Objekterkennung aus [112] lässt sich auf vier Stufen zusammenfassen. Zunächst wird ein Gauß-Filter auf das Bild angewendet, um Rauschen zu vermindern. Im nächsten Schritt wird das Bild in den HSV-Farbraum transformiert. Daraufhin wird ein Referenzhistogramm erstellt, welches im letzten Schritt mit den einzelnen Bildbereichen verglichen wird.

Die Unterschiede zum Ansatz, der in der Projektgruppe verfolgt wurde, sind zum Einem, dass für das Erstellen des Histogramms anstelle des RGB-Farbraums, der HSV-Farbraum gewählt wurde. Die Unterschiede lassen sich in 5.2.5 nachlesen. Ein weiterer Unterschied ist die Wahl des Referenzhistogramms. Anstelle des gesamten Bildbereichs, wurde eine trapezförmige Fläche ausgewählt. Abbildung 7.40 zeigt inwieweit sich die Wahl der Größe des Trapez auf die Qualität der Objekterkennung auswirkt.

Der Ansatz der histogrammbasierten Kollisionserkennung wurde allerdings nicht weiterverfolgt. Die Gründe dafür sind, dass der zuvor beschriebene Canny-Edge Ansatz gute Ergebnisse erzielt hat und die Zeit für die Implementierung eines weiteren nicht ausreichend war.

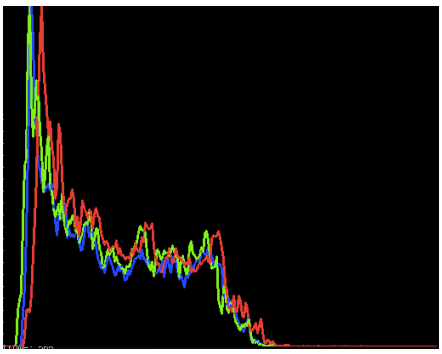
Frequenzbasierte Objekterkennung

Während der Bearbeitungsphase wurde neben den Canny Edge- und Histogrammbasierten Ansätzen zur Kollisionserkennung ein weiteres Verfahren – die sogenannte *Frequenzbasierte Objekterkennung* – prototypisch implementiert und ihre Eignung für den Einsatz im Rahmen des Projekts evaluiert.

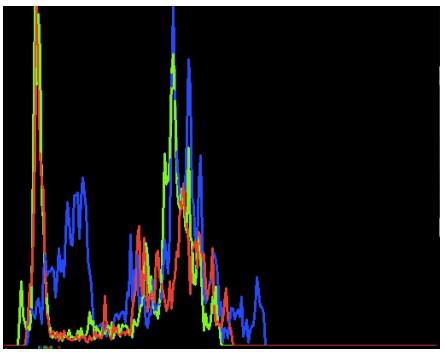
Die Grundlage für diese eigens entwickelte Vorgehensweise bildet die *Diskrete Fourier-Transformation (DFT)*, die ein gegebenes Signal aus dem Zeit-

(a) Referenzbild nach Horizonterkennung und Anwendung eines Weichzeichners

(b) Ausgewählter Teilbereich



(c) Histogramm der RGB-Werte des Referenzbildes



(d) Histogramm des ausgewählten Teilbereichs

Abbildung 7.39: Analyse der Farbwerte mittels Histogrammen

bereich in den Frequenzbereich überführt. Das aus der Signalverarbeitung bekannte Verfahren mithilfe der Gleichung

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-i2\pi(\frac{ki}{N}\frac{lj}{N})}$$

mit $e^{ix} = \cos x + i \sin x$ auf zweidimensionale Helligkeitsinformationen anwendend, kann das Frequenzspektrum jedes beliebigen Ausgangsbildes bestimmt werden. Während aus der Zerlegung in die sinus- und cosinusförmigen Signalanteile ein komplexwertiges Amplituden- und Phasenspektrum resultiert, ist für den gewählten Ansatz lediglich ersteres relevant (vgl. [88]).

Die bereits aus den vorherigen Abschnitten bekannte OpenCV-Bibliothek stellt eine effiziente Implementierung der beschriebenen Transformation be-

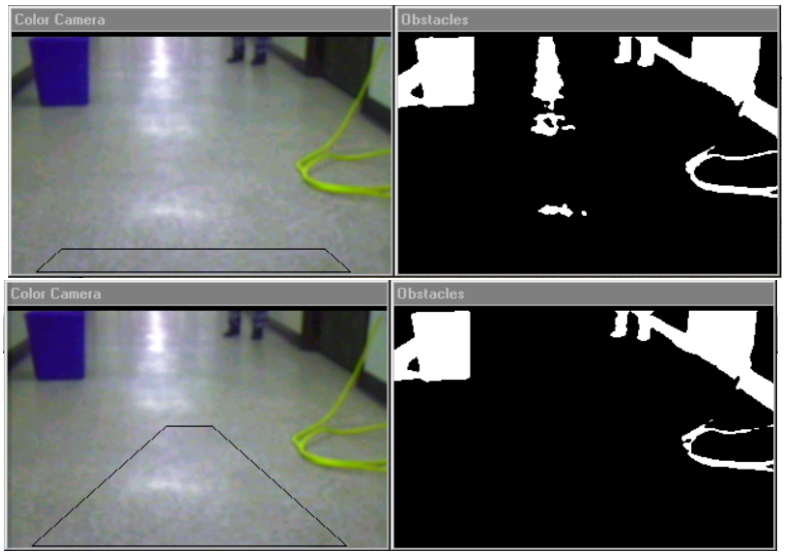


Abbildung 7.40: Trapezförmige Referenzhistogramme (aus [112])

reit. Eine ausführliche Beschreibung der verwendeten `dft()`-Methode ist in [88] zu finden. Aufbauend auf den Inhalten der zu Grunde liegenden Dokumentation wird die Funktionsweise des Verfahrens anhand eines beispielhaften Anwendungsfalls erläutert und der aktuelle Entwicklungsstand beschrieben.

Grundprinzip und Funktionsweise Vor dem Beginn der Analyse wird das zu verarbeitende Ausgangsbild in eine Graustufen-Repräsentation überführt. Um die eingelesenen Bilddaten anhand ihrer charakteristischen Eigenschaften in divergente Bildsegmente unterteilen und mögliche Kollisionsrisiken zuverlässig erkennen zu können, werden die Rohdaten in n gleichgroße, dem Seitenverhältnis des Ausgangsbildes entsprechende Kacheln zerlegt. Durch einen Vergleich der Datensätze können so schrittweise Abhängigkeiten und Unterschiede zwischen den Bildbereichen aus einer Häufung von Signalanteilen auffällig hoher oder niedriger Frequenzen abgeleitet werden.

Eine erste Idee für die Umsetzung der dargestellten Vorgehensweise beschreibt der nachfolgend aufgeführte Algorithmus, der als Konzept für die prototypische Implementierung dient und dem aktuellen Entwicklungsstand der frequenzbasierten Objekterkennung entspricht:

1. Einlesen und Vorverarbeiten des Bildes
2. Unterteilung des Bildes in n gleichgroße Kacheln
3. Bestimmen des Frequenzspektrums für jede Kachel

4. Berechnen und Vergleichen verschiedener Informationen unterschiedlicher Kacheln und Kachelformationen, um Hindernisse zu detektieren

5. Iterative Verfeinerung der Analyse durch schrittweise Verkleinerung der Kachelgröße, Wiederholung des beschriebenen Ablaufs

In Abb. 7.41 ist das Kamerabild einer typischen Ausgangssituation dargestellt, auf dem neben einem beispielhaften Hindernis auch der Uferbereich des Tweelbäker Sees zu erkennen ist. Auffällig ist in diesem Fall insbesondere die durch die Oberflächenstruktur des Wassers verborgene Wurfleine, die das Kollisionsrisiko unter realen Einsatzbedingungen maßgeblich erhöhen würde.

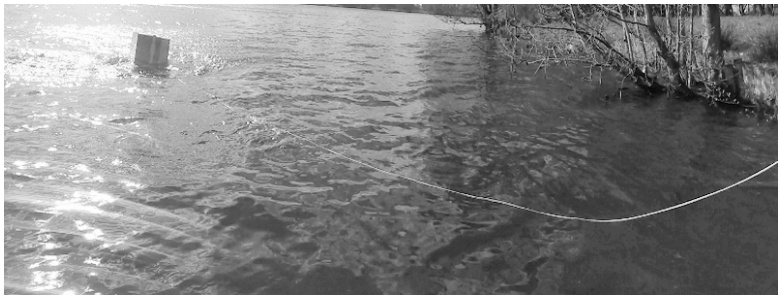


Abbildung 7.41: Ausgangsbild mit beispielhaftem Hindernis

Wie bereits zu Beginn dieses Abschnitts beschrieben, bedarf die Anwendung des frequenzbasierten Verfahrens zur Kollisionserkennung einiger Vorbereitungen, die in Abb. 7.41 bereits vorgenommen wurden. Neben dem Verwerfen sämtlicher Farbinformationen, dem Erkennen des Horizonts und der Extraktion der Wasseroberfläche sowie der Skalierung des Bildmaterials auf eine einheitliche Größe werden mögliche Reflektionen im Sichtbereich mithilfe des zuvor beschriebenen Verfahrens entfernt (vgl. Kapitel 7.3.1).

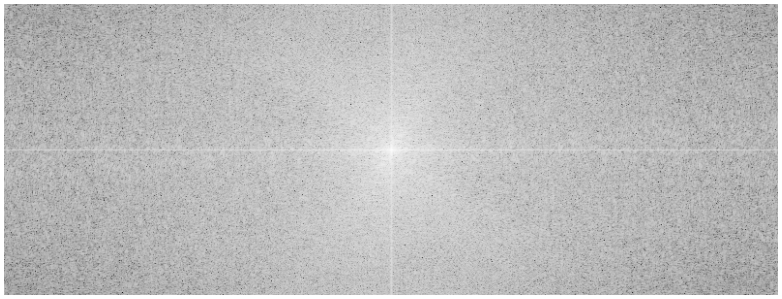


Abbildung 7.42: Frequenzspektrum des Ausgangsbildes

In Abb. 7.42 ist das in den Frequenzbereich überführte Ausgangsbild dargestellt. Während die Detektion von Hindernissen in der gewählten Visualisierung mithilfe des menschlichen Auges beinahe unmöglich erscheint, erlaubt die durchgeführte Transformation die Anwendung spezieller mathematischer Verfahren und minimiert den Aufwand für einen Einsatz der Standard-Filterkerne. Während die Überführung eines Bildes aus dem Zeit- und in den Frequenzbereich bereits zahlreiche Vorteile bei der Analyse mit sich bringt, eröffnen sich zudem neue Möglichkeiten für den Vergleich zweier Grafiken.

Um diesen Vorteil für das beschriebene Verfahren nutzbar zu machen, wird die Bildfläche des Ausgangsbildes in $25 \cdot 25 = 625$ Bildkacheln zerteilt, die durch die wiederholte Anwendung der `dft()`-Transformation iterativ in den Frequenzbereich überführt werden. Das Phasenspektrum verwerfend, ergeben sich so insgesamt 625 Amplitudenspektren, die in Abb. 7.43 an den ursprünglichen Positionen der Bildkacheln im Zeitbereich dargestellt sind.

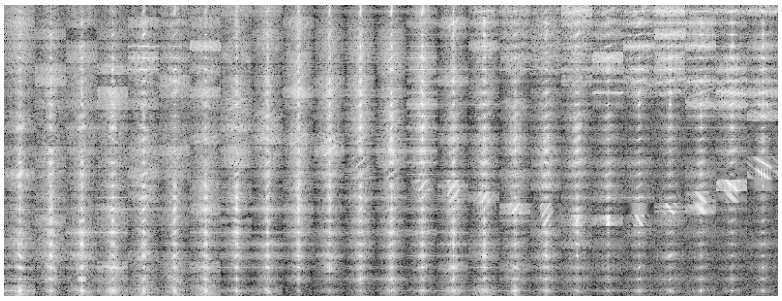


Abbildung 7.43: Frequenzspektren der Bildkacheln

Durch die Berechnung und den Vergleich der Mittelwerte, der minimalen und maximalen Amplituden sowie ausgewählter weiterer Metriken verschiedener Kacheln und Kachelformationen wird die ganzheitliche Analyse des zu Grunde liegenden Ausgangsbildes schrittweise vorangetrieben. Eine höhere Auflösung in fraglichen Bildbereichen anstrebend, kann zudem eine Verfeinerung der Analyse durch eine schrittweise Reduzierung der Kachelgrößen oder eine Verkleinerung des Bildausschnitts erreicht werden.

Aktueller Entwicklungsstand Trotz der guten Skalierbarkeit und der geringen Laufzeitkomplexität des Verfahrens wurde die softwaretechnische Umsetzung bereits zu einem frühen Zeitpunkt der Entwicklung eingestellt. Die Gründe für das Verwerfen dieses Ansatz gleichen denen für die Ablehnung der histogrammbasierten Objekterkennung. Die zeitliche Überschneidung mit den ersten Erfolgen des Canny Edge-Ansatzes sowie die Verlagerung personeller Kapazitäten führten dazu, dass die Arbeiten an der Entwicklung des beschriebenen Verfahrens frühzeitig abgeschlossen werden mussten.

7.3.2 Laserbasierte Kollisionserkennung

Während der Ausarbeitung verschiedener Ansätze zur Kollisionserkennung wurden ebenfalls Tests mit einem Laserscanner vorgenommen. Laserscanner funktionieren ähnlich wie RADAR-Systeme, setzen jedoch anstelle von Radiowellen Laserstrahlen ein, um eine optische Abstandsmessung vorzunehmen. Ein Laserstrahl wird ausgesandt und an einem Objekt reflektiert. Das reflektierte Licht kann detektiert werden und anhand der Laufzeit des Signals die Entfernung zum Objekt bestimmt werden.

Verwendete Hardware

Die ersten Tests wurden mit einem Laserscanner der Firma SICK durchgeführt. Das zur Verfügung gestellte Modell trägt die Bezeichnung *LMS151* und nutzt eine Wellenlänge von 905 nm (Infrarot). Es besitzt einen Öffnungswinkel von 270° und eine Auflösung von 0.5° bei einer Abtastfrequenz von 50 Hz. Der erfassbare Arbeitsbereich ist angegeben von 0.5 bis 50 m.

Der Scanner wird mittels einer Ethernet-Schnittstelle über das TCP/IP-Protokoll angesteuert. Über ein Socket wird eine Verbindung zum Scanner aufgebaut. Wurde die Verbindung akzeptiert, können über den Socket hexadezimale Befehle geschickt werden, die üblicherweise aus einen *Startcode*, mehrere *Befehl* und einen *Endcode* enthalten. Start- und Endcode werden mittels des Codes `0x02` (`STX`, *Start of Text*) und als `0x03` (`ETX`, *End of Text*) an die Sensoreinheit übertragen.

Zwischen der Ausführung der Start- und Endcodes wird zunächst die drei Bytes lange Zeichenfolge `sMN` verschickt, die die Art des auszuführenden Befehls kennzeichnet. Zum Starten einer Messung wird weiterhin das Kommando `LMCstartmeas` gesendet. Zusammengesetzt sieht die übertragene Zeichenfolge wie folgt aus: `0x02 sMN LMCstartmeas 0x03`. Nach dem erfolgreichen Start der angeforderten Messung wird vom Scanner die Bestätigung `0x02 sAN LMCstartmeas 0 0x03` zurückgemeldet, die dem `ACK`-Befehl des technischen Sicherheitskonzepts aus Abb. 6.7 entspricht.

Aufnahme der Messwerte

Nachdem eine Messung gestartet wurde, können die aktuellen Messwerte vom Scanner abgefragt werden. Die Scandaten enthalten neben den Distanz- und Remissionswerten auch den Gerätestatus des Sensorsystems. Letzterer gibt an, ob das Gerät betriebsbereit ist, eine Messung durchführt oder sich im Standby-Modus befindet. Der Messbereich des Scanners kann nun durch die Anzahl vorhandener Messwerte dividiert und für jeden Messwert so ein zugehöriger Messwinkel abgeleitet werden.

Auswertung der Abstandsinformationen

Die Extraktion von Hindernissen aus den zuvor aufgenommenen Messwerten erfolgt im Rahmen dieses Projektvorhabens mithilfe des sogenannten *Occupancy Grid*-Ansatzes. Nach dem Auslesen der Abstandsinformationen werden die abgeleiteten Distanzdaten durch ein iteratives Verfahren in eine zellenbasierte Darstellung überführt, die eine zuverlässige und effiziente Auswertung der aufgezeichneten Datensätze ermöglicht.

Um die Verarbeitung mithilfe des genannten Verfahrens zu ermöglichen und das spätere Testen der Implementierung zu vereinfachen, wird der Sichtbereich des LIDAR-Systems hierzu in einem ersten Schritt auf 180° begrenzt. Die ersten und letzten 180 Messpunkte (0° bis 44.75°, 225.25° bis 270°) des Datensatzes verwerfend, verbleiben damit 721 Messpunkte, die auf den neuen Sichtbereich von 0° bis 180° projiziert und für die Detektion möglicher Hindernisse herangezogen werden.

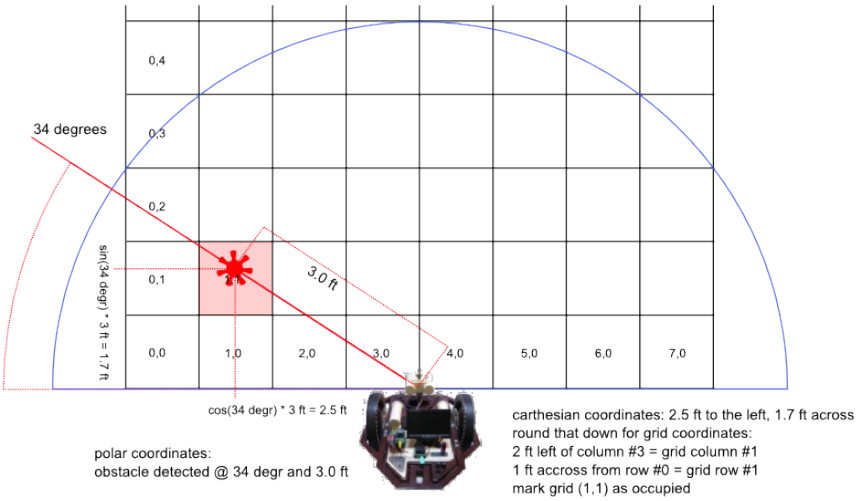


Abbildung 7.44: Grundprinzip des *Occupancy Grid*-Ansatzes (aus [96])

In Abb. 7.44 ist das Grundprinzip des *Occupancy Grid*-Ansatzes graphisch dargestellt. Während die verbleibenden Abstandsinformationen mit einem Abtastintervall von 0.25° vorliegen, können die Messwinkel α_i der Distanzdaten d_i unter Einbezug des Schleifenindizies $i \in \{0, ...720\}$ mit

$$\alpha_i = i \cdot 0.25$$

bestimmt werden. In einem zweiten Schritt können die zuvor erhaltenen Messwinkel weiterhin mithilfe der bekannten Formel

$$r_i = \alpha_i \cdot \frac{\pi}{180}$$

in die Längen r_i der korrespondierenden Kreisbögen im Einheitskreis transformiert und für die Berechnung der Koordinaten x_i und y_i des zu generierenden *Occupancy Grids* herangezogen werden:

$$x_i = \lfloor \cos(r_i) \cdot d_i \rfloor$$

$$y_i = \lfloor \sin(r_i) \cdot d_i \rfloor$$

Durch die Anwendung der vorgestellten Rechenschritte und Transformationen auf die zu Grunde liegenden Distanzdaten $D = \{d_1, ..., d_{721}\}$ ergibt sich eine Menge von Koordinatenpaaren $G_{\text{occupied}} = \{(x_k, y_k) \,|\, k \in \{1, ..., l\} \wedge l \leq i\}$ der *Occupancy Grid*-Zellen, die jeweils durch einen oder mehrere Messpunkten belegt sind. Aus der Anzahl der enthaltenen Messpunkte kann für jede Zelle anschließend ein individuelles Kollisionsrisiko bestimmt werden, das die Grundlage für die spätere Kollisionsverhütung bildet.

Um aus einzelnen oder mehreren Zellen des erzeugten *Occupancy Grids* zuverlässige Informationen über die Beschaffenheit der Umgebung gewinnen zu können, müssen sie schrittweise zu konkreten Hindernissen zusammengeführt werden. In einem ersten Schritt werden dazu zunächst alle Zellen (x_k, y_k) entfernt, deren Distanzdatum über den Wert $d_k = 0$ verfügt.

Einen einfachen Schwellwertfilter $d_k < c_{\text{cutoff}}$ mit $c_{\text{cutoff}} = 2$ anwendend, wird weiterhin die Verarbeitung fehlerhafter Messwerte, die durch Kratzer oder Wassertropfen im Sichtbereich des Sensors hervorgerufen werden können, unterbunden. Durch die Zusammenfassung sämtlicher verbleibender Zellen (x_i, y_i) und (x_j, y_j) mit $i \neq j$, für die $|\frac{x_i}{x_j}| \leq 20$ und $|\frac{y_i}{y_j}| \leq 20$ gilt, kann das *Occupancy Grid* in seine endgültige Form überführt werden.

Attribut	*Beschreibung*
`occupancyCount`	Anzahl der enthaltenen Messpunkte
`minAngle`	Kleinster enthaltener Messwinkel $\alpha_{\min}$
`maxAngle`	Größter enthaltener Messwinkel $\alpha_{\max}$
`minDistance`	Kleinste enthaltene Distanz $d_{\min}$
`maxDistance`	Größte enthaltene Distanz $d_{\max}$
`minAngleDistance`	Distanz am kleinsten Messwinkel $d(\alpha_{\min})$
`maxAngleDistance`	Distanz am größten Messwinkel $d(\alpha_{\max})$
`mergeCount`	Anzahl der zusammengefassten Zellen

Tabelle 7.2: Attribute der *Occupancy Grid*-Zellen

Da die in Tab. 7.2 dargestellten, für die Kollisionsverhütung relevanten Informationen während des gesamten Vorgangs an die jeweiligen Nachfolgezellen vererbt werden, können die Attribute der zu generierenden Hindernis-Objekte im Anschluss aus den noch verbleibenden Zellen extrahiert werden.

Nach dem Berechnen des Hindernis-Messwinkels

$$\alpha_{\text{obstacle}} = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})$$

und dem Ablesen der Entfernung $d_{\text{obstacle}} = d_{min}$ kann die Größe des detektierten Hindernisses mithilfe des Kosinussatzes über

$$s_{\text{obstacle}} = \sqrt{a^2 + b^2 - (2 \cdot a \cdot b \cdot \cos(\alpha_{\max} - \alpha_{\min}))}$$

geschätzt und das Hindernis-Objekt an die nachgelagerten Komponenten der Kollisionsverhütung weitergeleitet werden.

7.4 Kollisionsverhütung

Im Folgenden wird die Realisierung der Kollisionsverhütung beschrieben. Die für die Implementierung der Komponenten notwendigen Modifikationen wurden bereits in Abschnitt 7.2.4 beschrieben.

7.4.1 Ablauf der Kollisionsverhütung

Da die Hindernisse nur Positionsinformationen besitzen, die relativ zur aktuellen Position des Bootes interpretiert werden können, müssen diese in absolute Werte konvertiert werden. Dies ist mithilfe der aktuellen GPS-Position sowie des Headings des Bootes möglich. Zusammen mit der Distanz, dem Winkel zum Hindernis und dessen geschätzte Größe wird eine quadratische Gefahrenzone erstellt. Sobald die Position der Eckpunkte zweifelsfrei bestimmt wurden, können sie in den A^*-Graphen integriert und der `MissionController` in den `Found Obstacle`-Zustand gebracht werden. Von diesem aus wird die Wegplanung neu angestoßen. Dabei wird der `PathPlanner` in den Zustand `Computing` gebracht und der `MissionController` in den Zustand `Wait_for_Waypoints`. Währenddessen hält das Boot die Position, um eine direkte Kollision zu vermeiden. Nach der Wegplanung wird die Fahrt fortgesetzt und dem detektierten Hindernis mit einem angemessenen Sicherheitsabstand ausgewichen.

7.4.2 Simulation der Kollisionsverhütung

Um das entwickelte System zur Kollisionsverhütung zu testen wurde eine einfache Simulation entwickelt. Diese baut auf der bereits vorhandenen Simulation der Plattform da, die eine Missionsdurchführung ermöglicht.

Zusätzlich wurde ein TCP-Server eingerichtet, über den per Konsoleneingabe Hindernisse an die Clients verschickt werden können (siehe Abschnitt 7.2.5 für weitere Details). Verbinden sich nun die simulierte Missionsplanung und Onshore-Software mit dem Server können Missionen geplant und durchgeführt werden. Während der Durchführung können nun über den Server Hindernisse abgeschickt werden.

In der Simulation des Bootes auf einem Windows Rechner werden die verhütenden Maßnahmen rechtzeitig eingeleitet und das simulierte Boot weicht dem Hindernis aus. Unter reellen Bedingungen scheitert die Verhütung daran, dass die Rechenleistung des Raspberry Pis nicht genügt um die Bahnplanung in ausreichender Zeit durchzuführen. Dadurch hält das Boot mehrere Sekunden die Position, in dieser Zeit kann sich das Hindernis jedoch weiter verschoben haben oder das Boot wurde durch Strömungen verschoben.

7.5 Regelung

Um eine autonome Anfahrt von Missionspunkten gewährleisten zu können, ist es essentiell, dass das Boot ein möglichst direktes und stabiles Fahrverhalten aufweist. Eine Voraussetzung hierfür ist eine korrekt eingestellte Regelung.

7.5.1 Bisherige Reglerstruktur

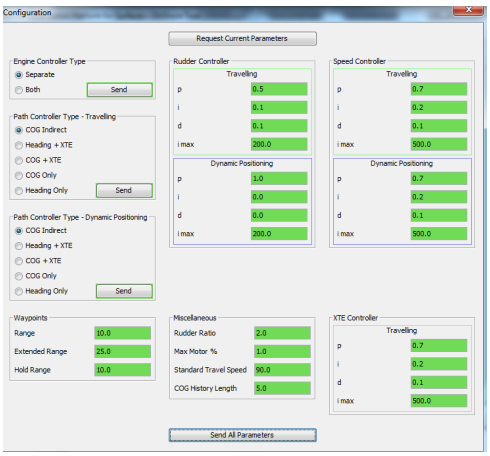


Abbildung 7.45: Onshore-Software mit Reglerparametern

Da es keine Dokumentation über die bisher eingesetzte Reglerstruktur gibt, ist die bisher eingesetzte Reglerstruktur nicht ohne weiteres nachvollziehbar. Es ist lediglich bekannt, dass fünf Regler vorhanden sind. Diese werden in der Tabelle 7.3 aufgezeigt. Abbildung 7.45 zeigt die bisher eingestellten

Parameter in der Onshore-Software. Der Verwendungszweck und die Einsatzweise können anhand der Namensgebung nur vermutet werden.

Rudder Controller	Travelling
Rudder Controller	Dynamic Positioning
Speed Controller	Travelling
Speed Controller	Dynamic Positioning
XTE Controller	Travelling

Tabelle 7.3: Übersicht über die bisher eingesetzten Regler

Um die bisherige Reglerstruktur kennenzulernen, wurde der Quellcode der Vorgängergruppe Reverse Engineered. Hierbei lag das Augenmerk auf dem Package Pathcontrol. In Pathcontrol gibt es verschiedene Klassen die in der Abbildung 7.46 gezeigt werden. Diese Klassen implementieren verschiedene Regelungsvarianten, wie:

- Die Regelung des Headings, bei der die Differenz zwischen Heading und Bearing berechnet wird und als Regelungsfehler an einen PID-Regler weitergegeben wird.
- Die direkte Regelung des COG, bei der die Hundekurven vermieden oder verringert werden.
- Die indirekte Regelung des COG, bei der ein Vorhaltewinkel antizipiert werden kann.
- Die Regelung des XTE, bei der die Cross Track Error (XTE) begrenzt wird.

Die Parameter P, I und D werden im PIDController erzeugt und in den Funktionen und Objekten wie COGRudder, rudderController, speedController, xteController, PathControllerHeadingonly, PathControllerCOGXTE, PathControllerHeadingXTE verwendet.

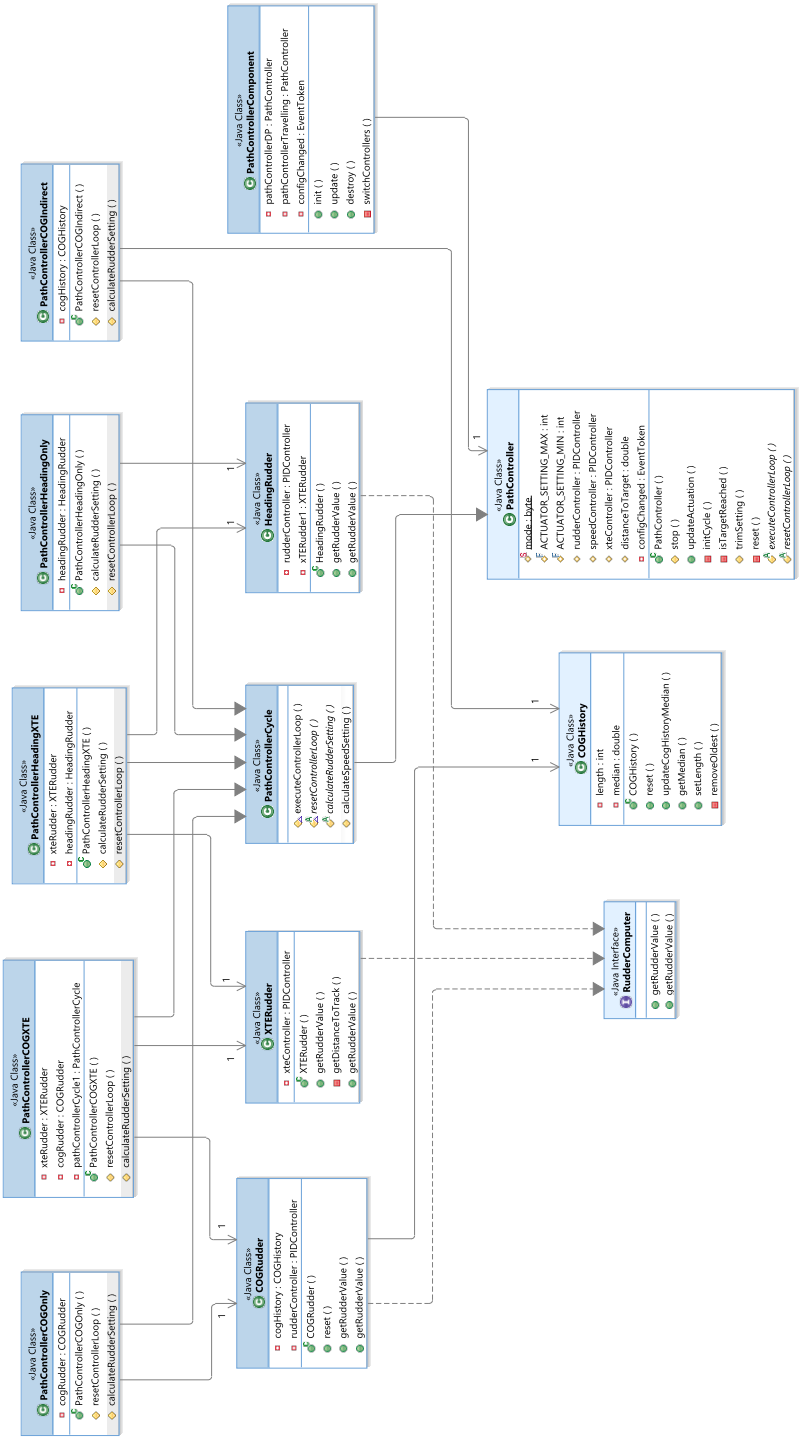


Abbildung 7.46: Klassendiagramm „Regelung“

In den durchgeführten Praxistests zeigte sich, dass die bisherige Regelung ein instabiles Verhalten hervorrufen kann. Dies zeigte sich in stark ausgeprägten Schlingerkursfahrten, die das Boot weit von den vorgegebenen Wegpunkten entfernen lässt. Hieraus lässt sich schließen, dass die bisherige Reglerstruktur oder die eingesetzten Parametersätze nicht optimal gewählt wurden. Um eine kontrollierte und stabile Fahrweise des Wasserfahrzeugs zu gewährleisten, gilt es dieses zu verbessern.

7.5.2 Schema des neuen Regelkreises

Im Folgenden wird der Aufbau des neuen Reglerkonstrukts thematisiert. Das Augenmerk liegt hier auf der Bestimmung der Sollwerte und Parameter. Des Weiteren liegen die Entscheidungen bzgl. der Struktur im Fokus.

Sollwertberechnungen

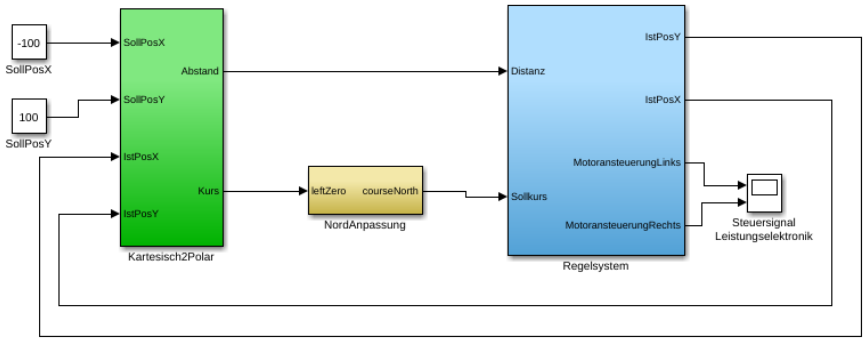


Abbildung 7.47: Oberste Ebene des Regelsystems

Eine grundlegende Fähigkeit der Plattform soll es sein, eine gewünschte Position mit Unterstützung der Regelung anfahren zu können. Hierfür gilt es, zuerst die Differenz zwischen Soll- und Ist-Position zu bestimmen. Da in diesem Anwendungsfall die Koordinaten via GPS ermittelt werden, war die Nutzung eines kartesischen Koordinatensystems naheliegend. Werden nun die Positions- und Sollkoordinaten als Ortsvektoren betrachtet, können diese von einander subtrahiert und aus dem Ergebnis ein Richtungsvektor $\overrightarrow{r}$ gebildet werden. Die Formel kann in der Abbildung 7.48 betrachtet werden. Aus dem Richtungsvektor $\overrightarrow{r}$ wird ein Betrag und eine Richtung gebildet. Diese werden dann als Entfernung oder Abstand und Kurs interpretiert. Abbildung 7.47 zeigt linksseitig die Aufnahme der Werte für die Formel 7.1 und die Weiterleitung von Kurs und Abstand im Simulinkmodell.

$$\overrightarrow{r} = \overrightarrow{p_{\text{soll}}} - \overrightarrow{p_{\text{ist}}} = \begin{bmatrix} p_{\text{soll x}} \\ p_{\text{soll y}} \end{bmatrix} - \begin{bmatrix} p_{\text{ist x}} \\ p_{\text{ist y}} \end{bmatrix} \tag{7.1}$$

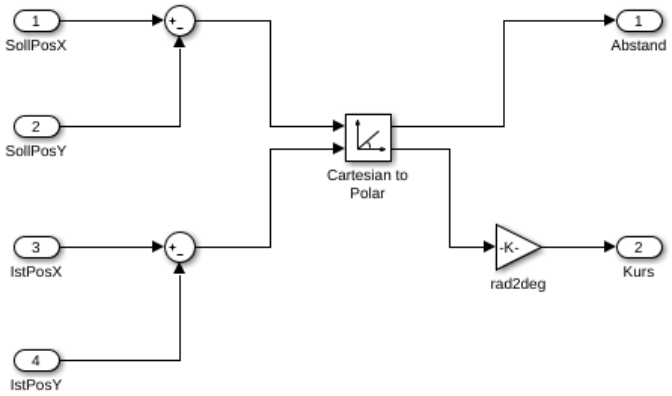


Abbildung 7.48: Umrechnung kartesischer in Polarkoordinaten

Wird das innere des Subsystems *Kartesisch2Polar* aus der Abbildung 7.47 betrachtet, ist in der Abbildung 7.48 linksseitig die Subtraktion aus der Formel 7.1 zu sehen. Der Funktionsblock *Cartesian to Polar* bildet aus der Differenz der Koordinaten den Abstand und den Kurs. Rechtsseitig bildet der Funktionsblock *rad2deg* die Winkelwerte aus dem Radianten. Der nun zur Verfügung stehende Kurs, ist zur X-Achse des kartesischen Koordinatensystems ausgerichtet. Ein Kurs von 0° würde also parallel zur X-Achse verlaufen. Um einer eindeutigeren, beziehungsweise einer intuitiven Definition zu folgen, sollte die 0°-Ausrichtung des Kurses parallel zur Y-Achse zeigen. Daher muss der Kurs um 90° verschoben werden. Wird also ein Kurs von 90° berechnet, soll dieser als 0° interpretiert werden. Deswegen muss von jedem Kurs ein Winkel von 90° subtrahiert werden. Werden die Begrifflichkeiten aus der Abbildung 7.47 genutzt, wird vom Wert *leftZero* die Subtraktion durchgeführt, um den korrigierten Wert *courseNorth* zu erhalten. Die Formel 7.2 stellt dies dar und wird im Subsystem *Nordanpassung* realisiert. Die Abbildung 7.49 zeigt das Innere des Subsystems. Der korrigierte Kurswert wird dann dem Subsystem *Regelung und Strecke* zur Verfügung gestellt.

$$courseNorth = leftZero - 90° \tag{7.2}$$

Regelung und Strecke

Die Differenz der Koordinaten kann als Vektor interpretiert werden, welcher den Kurs und die Entfernung darstellt. Hieraus lässt sich ableiten, dass mindestens zwei Regler vorhanden sein müssen. Einen für den Kurs und einen

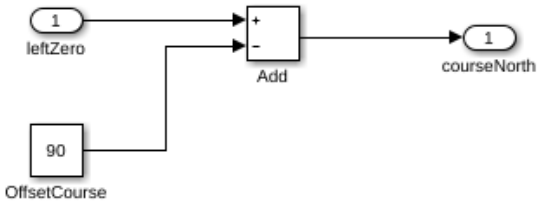


Abbildung 7.49: Nordanpassung

Regler welcher die Entfernung gegen Null regelt. Diese Regler und die Regelstrecke werden im Folgenden thematisiert.

Regelung der Geschwindigkeit Ohne in erster Betrachtung auf Extremfälle bzgl. der Störgrößen einzugehen, kann die Entfernung auch als Produkt aus Geschwindigkeit und Zeit interpretiert werden. Die Geschwindigkeit soll bei großen Entfernungen hoch sein, um diese möglichst schnell zu überbrücken. Bei kürzeren Entfernungen soll eine kleinere Geschwindigkeit gewählt werden. Sollte nun durch den Einfluss von Störgrößen, z.B. eine kleinere Entfernung nicht in einer annehmbaren Zeit überbrückt werden können, gilt es die Geschwindigkeit zu erhöhen. Der Einfluss der Zeit wird erst in zweiter Betrachtung hinzugefügt. Vorerst muss das Funktionsprinzip des Entfernungsreglers betrachtet werden, um die Geschwindigkeit validieren zu können. Um aber schon in erster Betrachtung den Einfluss der Zeit vorbereiten zu können, ist es notwendig hierfür ein Reglerkonstrukt zu wählen. Sollte die Wahl auf einen klassischen PID-Regler fallen, müssten die Übertragungsfunktionen mit großem Aufwand angepasst werden. Aus diesem Grund bietet sich ein Fuzzy-Regler an. Hier müssten nur die Regeln zur Verarbeitung angepasst werden, was ist mit geringerem Aufwand möglich ist. Daher wird dieser als Regler für die Entfernung eingesetzt.

Abbildung 7.50 zeigt den grundliegenden Aufbau einer Fuzzy-Logik in Simulink. Im linksseitigen Block wird ein Entfernungswert als Vorgabe für den Block *EntfGeschw Regler* generiert. Der Reglerblock setzt den Entfernungswert nun über die Fuzzy-Regel in eine Geschwindigkeit um.

Abbildung 7.51 zeigt im oberen Diagramm eine sinkende Entfernung und im Unteren die proportional sinkende Geschwindigkeit. Der genaue Verlauf der Kurven kann über den Verlauf der Memberfunktionen eingestellt werden. In Abbildung 7.51 wird lediglich das Funktionsprinzip erläutert. Durch den Verlauf und die Verknüpfung der Memberfunktionen, wird der Verlauf der Ausgangsgröße bestimmt. Die Abbildungen 7.52 und 7.53 zeigen die Memberfunktionen über die der Outputverlauf eingestellt werden kann.

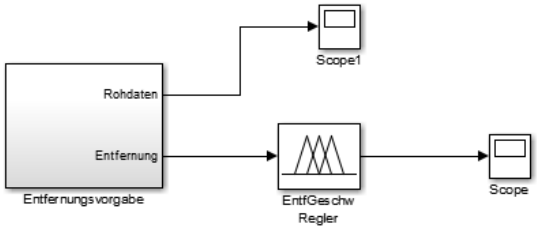


Abbildung 7.50: Entfernungs- und Geschwindigkeitsumsetzung

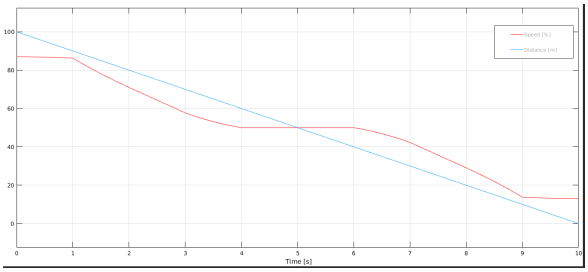


Abbildung 7.51: Entferung und Geschwindigkeit aus der Fuzzy-Logik

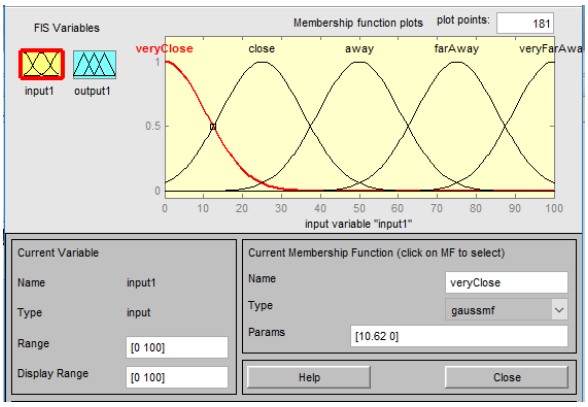


Abbildung 7.52: Memberfunktionen für die Inputvariable

Über den Wizard welcher in den Abbildungen 7.52 und 7.53 gezeigt wird, kann die Kurvenform und die Position festgelegt werden. Des Weiteren kann der Zahlenbereich der Funktionen festgelegt werden. Um nun die zeitliche Komponente mit in den Fuzzy-Regler einzubringen, müsste eine zweite Input-

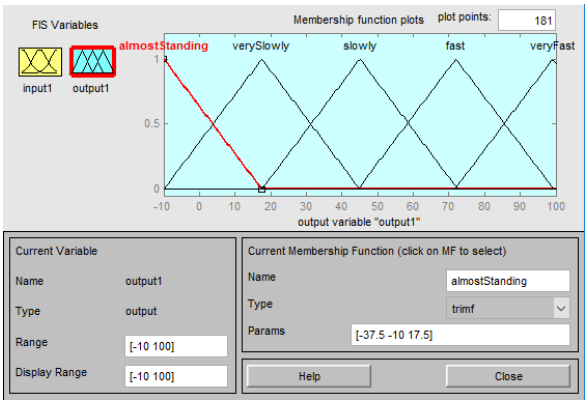


Abbildung 7.53: Memberfunktionen für die Outputvariable

Memberfunktion angelegt werden. Über die zeitliche Komponente könnte überprüft werden ob für eine Wegstrecke zu viel Fahrzeit benötigt wird und dann die Geschwindigkeit erhöhen.

Regelung des Kurses Um den Kurs zu regeln, bietet sich ein klassischer PID-Regler an, da bei der Kursreglung nach erprobten Einstellmaßnahmen vorgegangen werden kann und diese nach erfolgreicher Einstellung als gesetzt angesehen werden können.

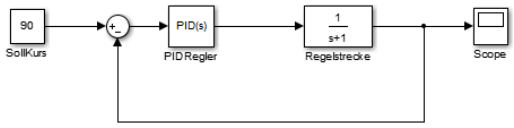


Abbildung 7.54: Grundsätzlicher Aufbau des PID-Regelkreises

Abbildung 7.54 zeigt einen minimalen PID-Regelkreis. Linksseitig vor dem Additionspunkt wird ein Sollkurs angelegt. Nachdem die Rückführung des Regelkreises von der Sollgröße subtrahiert wurde, verarbeiten der PID-Regler und die Übertragungsfunktion der Regelstrecke (hier beispielhaft dargestellt) die Führungsgröße. Diese wird zum Additionspunkt zurückgeführt. Abbildung 7.55 zeigt das Verhalten der Regelgröße. Hier wird der Kurs von 0° auf 90° in fünf Sekunden ausgeregelt. In den folgenden Schritten ist es nun notwendig, die Parameter für die Regelstrecke und den Regler so einzustellen, dass es der Realität entspricht bzw. ein stabiles Systemverhalten gezeigt werden kann.

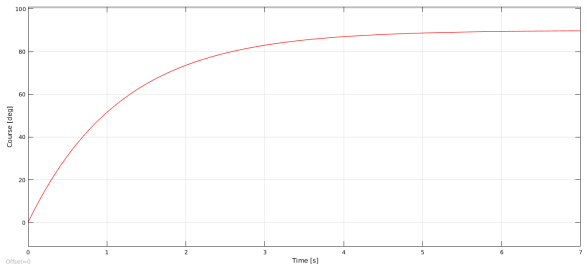


Abbildung 7.55: Regelgröße bei Kurswechsel allgemein

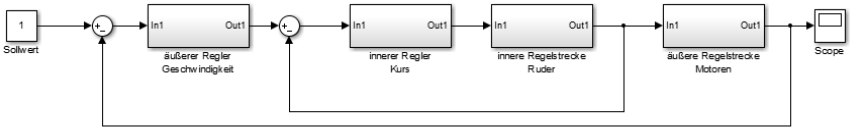


Abbildung 7.56: Kaskadenregelung

Zusammenführen der Regler Nun gilt es festzulegen, welche Struktur der spätere Regelkreis haben soll. Klassischerweise würde eine Kaskadenregelung für diesen Anwendungsfall gewählt werden. Abbildung 7.56 zeigt einen beispielhaften kaskadierten Regelkreis. Hierbei müsste die schneller agierende Regelung, in den Regelkreis des langsamer agierenden Systems eingefügt werden. In Abbildung 7.56 ist der vermeintlich schneller agierende Regler, der Kursregler und der langsamere Regler der Geschwindigkeitsregler. Es ist in diesem Anwendungsfall anzunehmen, dass eine Kursänderung schneller erfolgen muss als der Einfluss der Entfernung. Bei falsch geregeltem Kurs wirkt sich dieses auch auf die Entfernung negativ aus. Um aber die Regler ineinander zu schalten, müssten die physikalischen Größen der Regler aufeinander abgestimmt sein. Es wird also aus dem überlagertem Entfernungsregler eine Drehmomentenanforderung für die Leistungselektronik herausgegeben. Der Eingang des Kursreglers erwartet aber als Eingangsgröße eine Kursdifferenz. Das heißt, es müsste der Einfluss der Drehmomentenanforderung durch ein mathematisches Modell umgerechnet werden, um daraus den Sollkurs zur Bestimmung der Kursdifferenz zu erhalten. Da sowohl die Kursinformation, als auch die Entfernung beziehungsweise die Geschwindigkeit durch die Positionsüberwachung via GPS vorhanden sind, steht die Überlegung an, beide Regler getrennt voneinander zu betreiben und die Ausgänge auf einen Summenpunkt zu legen, bevor die Regelstrecke gespeist wird. So kann ein Differenzsignal für die Motoren erzeugt und sowohl als Geschwindigkeit als auch Kurs dargestellt werden. Abbildung 6.16 zeigt dieses Beispielhaft. Für das Geschwindigkeits-

signal ist hierzu keinerlei Anpassung notwendig. Für das Kurssignal muss das Ausgangssignal des Reglers differenziert werden und auf einen Wert für den linken und rechten Motor umgerechnet werden.

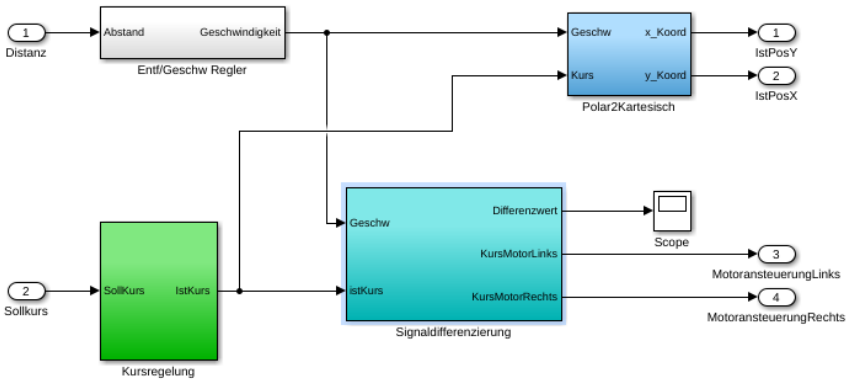


Abbildung 7.57: Zusammenfassen der Ansteuersignale beider Motoren

Abbildung 7.57 zeigt dass die Ausgänge der beiden Regler in einem Subsystem zusammengefasst werden. Das Subsystem Signaldifferenzierung berechnet die Sollwerte für die Leistungselektronik der beiden Motoren.

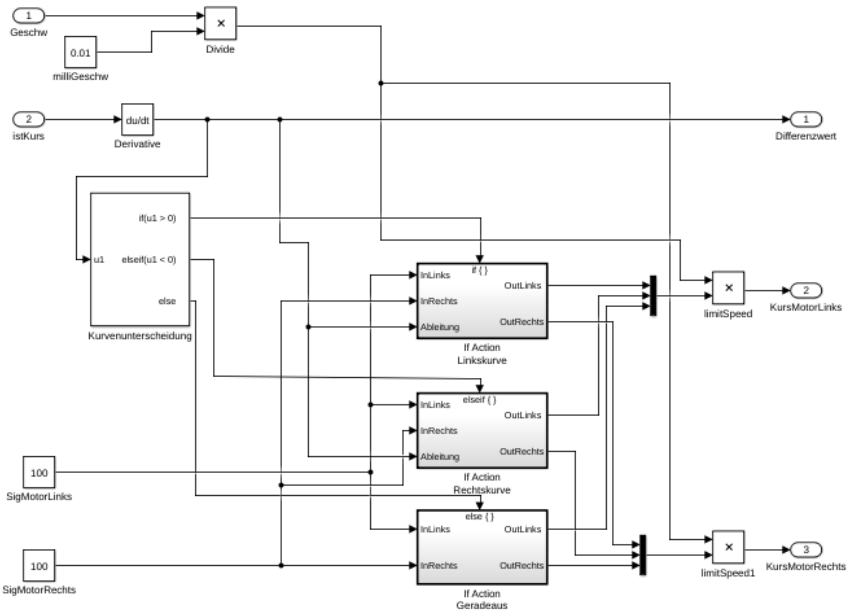


Abbildung 7.58: Einfluss der mathematischen Ableitung auf den Kurs

In Abbildung 7.58 wird gezeigt, dass der *IstKurs* durch den Funktionsblock *Derivate* (links oben in der Abbildung) eine mathematische Ableitung nach der Zeit erfährt. Durch das Vorzeichen des Ergebnisses der Ableitung kann entschieden werden, ob es sich um eine Kurvenfahrt nach Links oder nach Rechts handelt. Sollte die Ableitung den Wert 0 haben, handelt es sich um eine Geradeausfahrt. Abbildung 7.59 zeigt die Gradänderung bei den Kurswechseln. Der Kurswechsel in die jeweilige Richtung erzeugt das gleiche Vorzeichen bei der Ableitung.

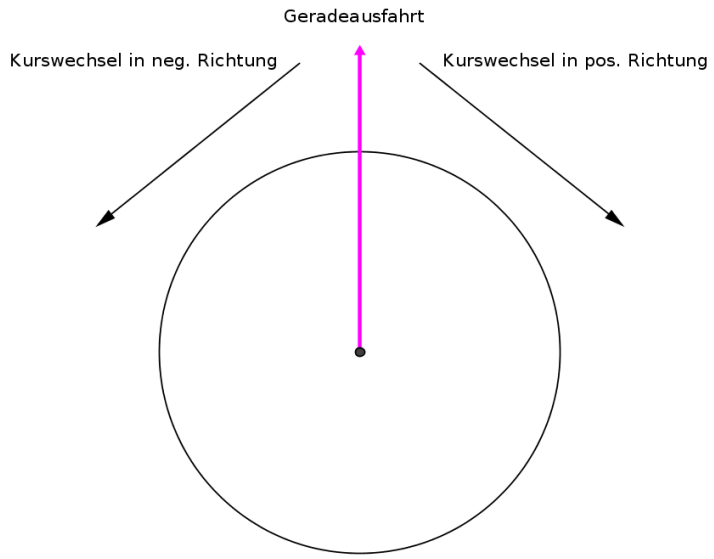


Abbildung 7.59: Richtungswechsel im Kurskreisel

Abbildung 7.60 zeigt das Steuerverhalten, wenn eine Linkskurve um 90° durchgeführt wird. Hier wird der Sollkurs von 90° angestrebt und nach ca. 5 Sekunden gehalten. Der rechte Motor steuert über den kompletten Zeitraum mit 100 % Ausgangsleitung. Proportional zur Kurvenfahrt steuert der linke Motor im Verlauf des Messintervalls von 0 bis 100 % auf.

Nun gilt es die beiden Ausgänge der Regelkreise zusammenzulegen und zu verarbeiten. Das Geschwindigkeitssignal stellt die max. Aussteuerung des Ausgangssignals dar. Der Faktor des Kurvensignals beeinflusst das Signal der jeweilige Motorseite. Im oberen Teil der Abbildung 7.58 wird das Geschwindigkeitssignal durch hundert dividiert und mit dem berechneten Ansteuersignal multipliziert. Hierdurch wird die max. Ansteuerung je nach geforderter Geschwindigkeit begrenzt.

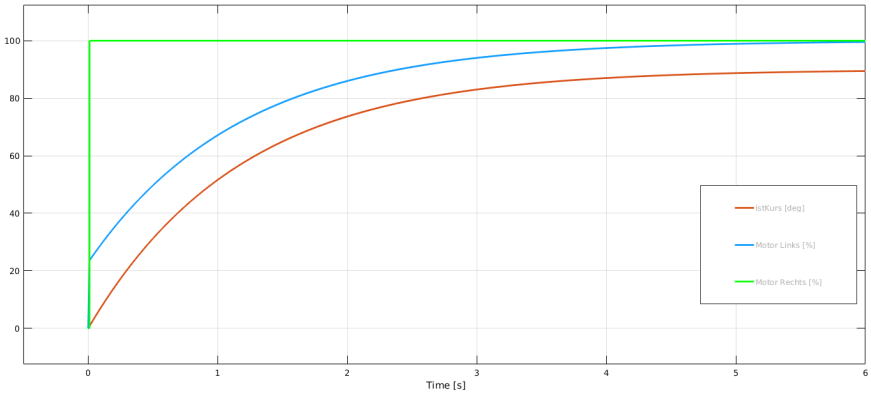


Abbildung 7.60: Steuerverhalten bei einer Linkskurve

7.5.3 Parameterbestimmung

Um eine stabile und funktionierende Regelung entwerfen zu können, ist es notwendig die Systemparameter zu bestimmen. Sollten die Parameter nicht passend gewählt werden, kann entweder ein Missionsziel nicht angefahren werden, da der Missionspunkt durch ungenaue Anfahrt nicht erreicht wird oder das System reagiert instabil und zeigt massive Schlingerfahrten. Um die Werte für die Regelparameter zu bestimmen, wird die Sprungantwort aufgenommen (siehe Kapitel A.10 - Verfahren zur Parameteridentifikation). Hieraus kann ein Geschwindigkeitsdiagramm erstellt werden aus dem die Regelparameter bestimmt werden können. Die Geschwingkeit wird durch den Einsatz von Beschleunigungssensoren und das Integrieren des Ausgangssignals der Beschleunigungssensoren geregelt. Siehe Formel 7.3.

$$\overrightarrow{v}(t) = \int \overrightarrow{a}(t)\ \delta t \tag{7.3}$$

Die Beschleunigungssensoren die hier eingesetzt wurden, sind Bestandteil eines Smartphones und eines Tablets. Dies hatte den Vorteil, dass die Sensorik ein autonomes System darstellt und die Fähigkeit besitzt, die Beschleunigungswerte zu speichern. Um die Sprungantwort des Systems zu erlangen wurde das Wasserfahrzeug aus dem Stillstand beschleunigt, bis die maximale Geschwindigkeit erreicht wurde. Dieser Vorgang wurde mehrmals wiederholt um eventuell aufgetretene Störgrößen zu ermitteln. Als Störgrößen können beispielsweise Strömungen oder Wind auftreten.

Szenarien

Die Regelparameter wurden via Sprungantwortverfahren während der Testfahrten ermittelt. Hierzu wurden Szenarien gefahren in denen die Plattform

aus dem Stillstand maximal beschleunigt oder aus einer Geradeausfahrt einen drastischen Kurswechsel erfährt. Während der Szenarien werden aus den Werten für `Latitude`, `GPS-Course-Over-Ground`, `GPS-Speed` und `NextWayPoint` ein Logfile erzeugt. Parallel dazu wurde ein Beschleunigungssensor aus einem Smartphone bzw. einem Tablet genutzt um Beschleunigungsdaten aufzuzeichnen. Dieses wurde während drei Testfahrten durchgeführt. Die Daten der Testfahrten wurden gesammelt, gemittelt und dienen als Grundlage zur Analyse des Beschleunigungs- und Kurvenverhaltens des autonomen Wasserfahrzeugs.

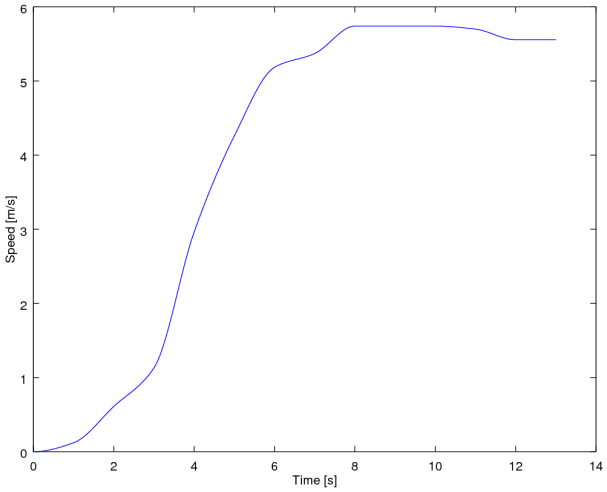


Abbildung 7.61: Geschwindigkeitsdiagramm

Geschwindigkeit Abbildung 7.61 zeigt das Geschwindigkeitsdiagramm des Bootes. Es ist zusehen, dass vom Stillstand auf eine maximale Geschwindigkeit beschleunigt wird. Ein minimales Überschwingen ist zu erkennen.

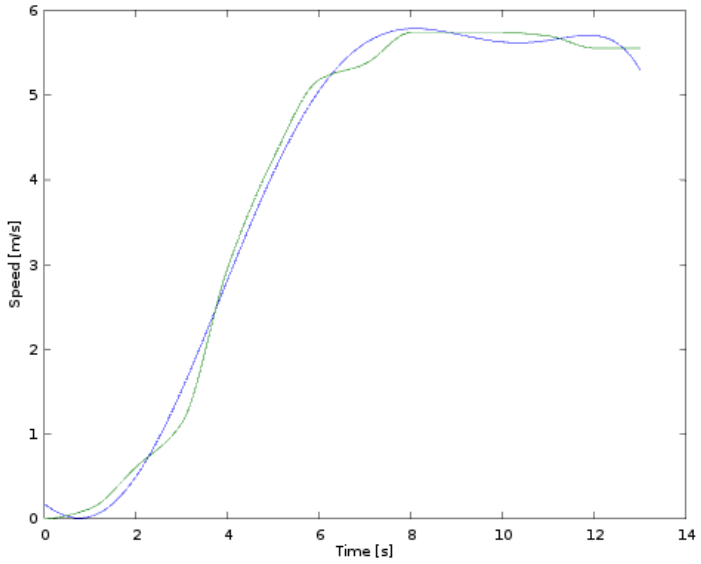


Abbildung 7.62: Interpoliertes Geschwindigkeitsdiagramm

Um die Einstellregeln für des Wendetangentenverfahren anwenden zu können, müssen die Daten interpoliert werden. Abbildung 7.62 zeigt die interpolierte Funktion in blau auf dem Datensatz. Die interpolierte Funktion dient nun als Vorbild der Regelstrecke für den Entfernungs-/Geschwindigkeitsregler und wird in Näherung in Simulink nachgebildet.

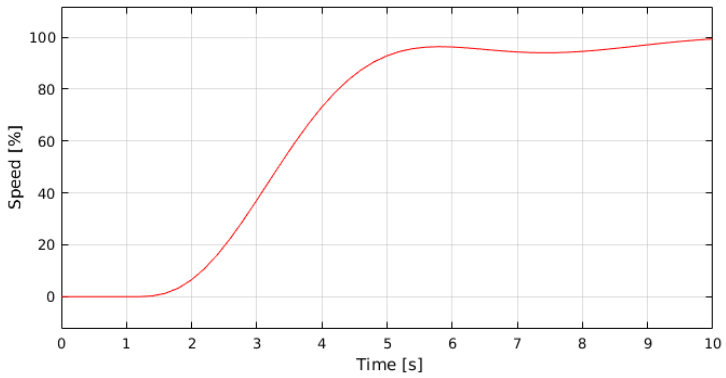


Abbildung 7.63: Angenäherte Funktion der Geschwindigkeits-Regelstrecke

Abbildung 7.63 zeigt die angenäherte Funktion der Regelstrecke für die Geschwindigkeit. Formel 7.4 zeigt die genutzte Transferfunktion $G(s)$. Diese wird an den Fuzzy-Regler angefügt.

$$G(s) = \frac{1}{1,3s^3 + 1,8s^2 + 2,5s + 1} \tag{7.4}$$

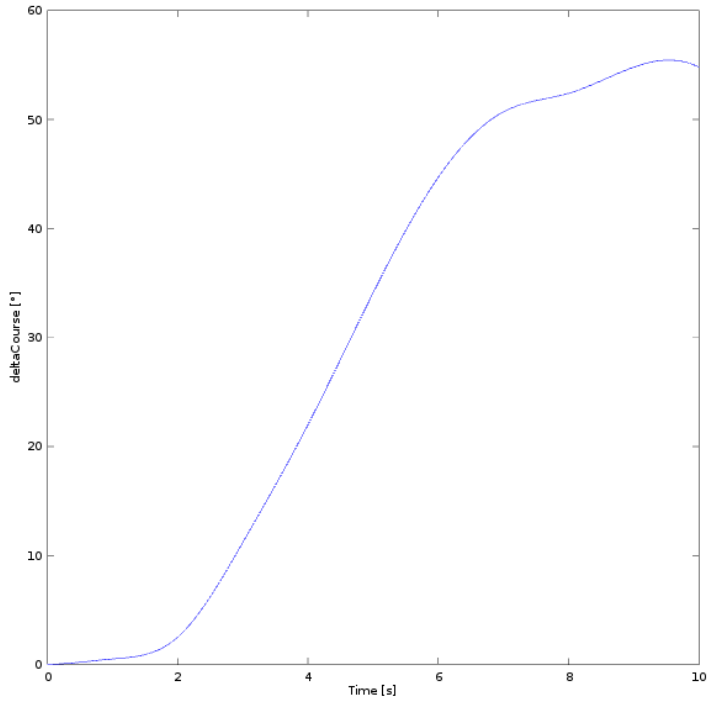


Abbildung 7.64: Kursdiagramm

Kurs Bezüglich des Kurvenverhaltens, wurde während der Testfahrten das in Abbildung 7.64 gezeigte Fahrprofil geloggt. Hierbei wurde eine Kursänderung von 54.8° beschrieben. Um diesen Datensatz für das Simulinkmodell nutzen zu können, muss ein Polynom über die Kurve gelegt werden.

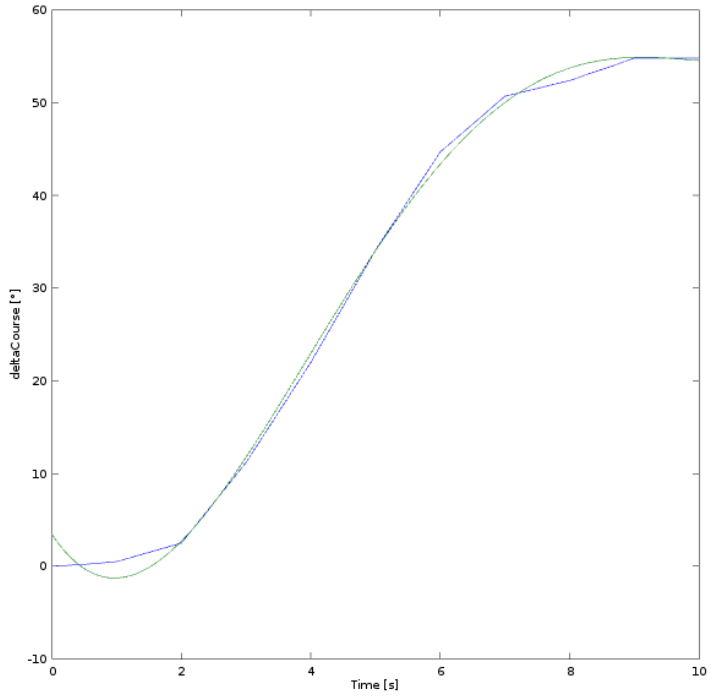


Abbildung 7.65: Angenäherte Polynomfunktion des Kursdiagramms

Abbildung 7.65 zeigt die angenäherte Polynomfunktion als grüne Kurve. Diese gilt es nun als Laplacefunktion in Simulink nachzubilden um die Regelstrecke für den Kursregler zu finden.

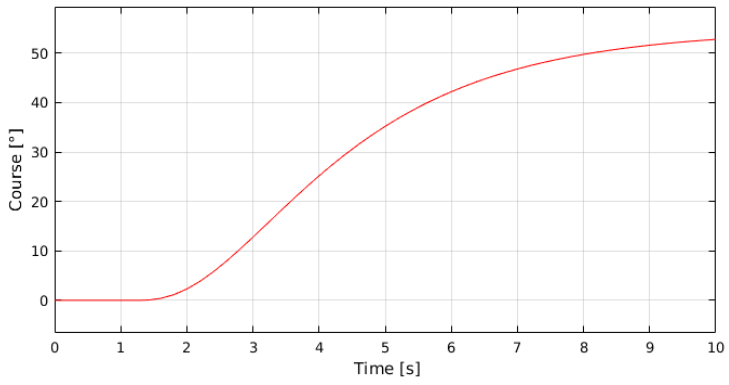


Abbildung 7.66: Angenäherte Regelstrecke für den Kurs

Abbildung 7.66 zeigt die angenäherte Funktion der Regelstrecke für das Simulinkmodell. Die dazugehörige Übertragungsfunktion $G(s)$ wird in Formel 7.5 gezeigt.

$$G(s) = \frac{1}{1,3s^3 \ + 3,5s^2 \ + 3,5s \ + 1} \tag{7.5}$$

Um die Werte für den PID-Regler zu identifizieren muss eine Wendetangente in die Polynomfunktion der Sprungantwort gelegt werden.

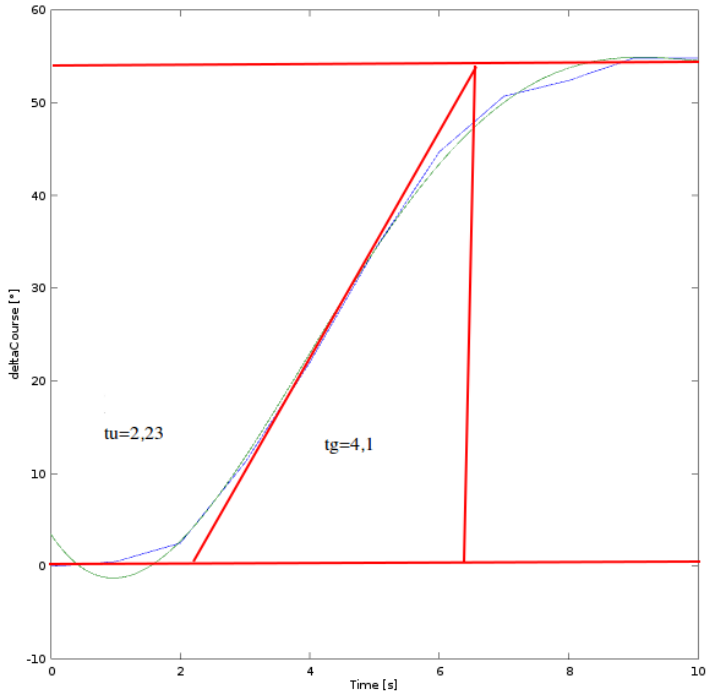


Abbildung 7.67: Wendetangente Kursdiagramm

Abbildung 7.67 zeigt die angelegte Wendetangente im Kursdiagramm. Hier wurden die Parameter $t_u = 2.23s$ und $t_g = 4.1s$ identifiziert. Laut Tabelle mit den Einstellregeln von Ziegler und Nichols (Tabelle A.7) werden folgende Formeln genutzt um die Parameter P,I und D zu bestimmen. Die Variable K_s gibt das Verstärkungsverhältnis zwischen eingeschwungener Ausgangsgröße und Sollwertgröße an. Da die eingeschwungene Ausgangsgröße gleich der Sollgröße ist, kann der Wert für K_s gleich eins gesetzt werden.

$$K_P = \frac{1,2 \cdot t_g}{t_u \cdot K_s} = \frac{1,2 \cdot 4,1s}{2,23s \cdot 1} = 2,2 \tag{7.6}$$

$$T_N = 2 \cdot t_u = 2 \cdot 2,23s = 4,46s \tag{7.7}$$

$$T_V = 0,5 \cdot t_u = 0,5 \cdot 2,23s = 1,115s \tag{7.8}$$

Diese Werte werden nun in die Reglergleichung $R(s)$ aus Formel 7.9 übernommen.

$$R(s) = K_P \cdot \left(1 + \frac{1}{T_N \cdot s} + T_V \cdot s\right) \tag{7.9}$$

7.5.4 Simulation der Regelung

Um die Antwort des Systems zu erhalten wird ein Kurswechsel um 90° und eine Entfernung von 100 m eingestellt.

Simulation des Kurses

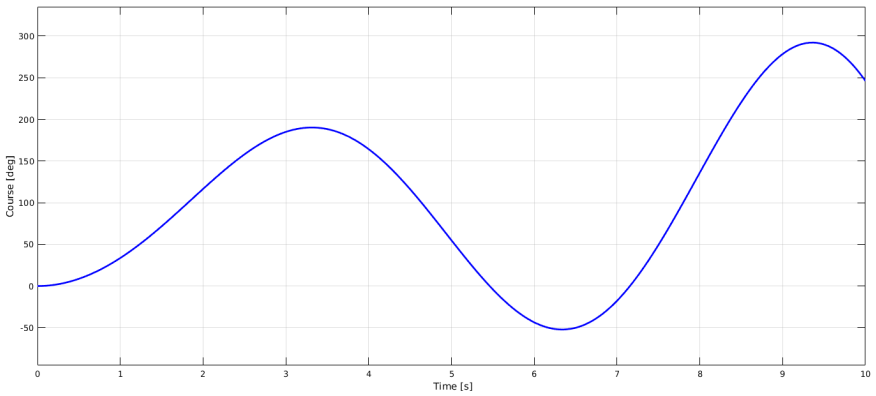


Abbildung 7.68: Simulierter Kursverlauf

In der Abbildung 7.68 wird der Verlauf des Kurses gezeigt wenn eine 90°-Wende erfolgen soll. Aufgrund der immer größer werdenden Schwingungen kann das Verhalten als instabil bezeichnet werden. Der Grund hierfür ist in den verwendeten Reglergleichungen zu finden. Im bisherigen Vorgehen ist die Reglergleichung nach Formel 7.9 verwendet worden. Zur Simulation des Systems wurde die Software MATLAB/SIMULINK verwendet. Die hier genutzte Reglergleichung $R_M(s)$ ist in Formel 7.10 zu sehen.

$$R_M(s) = P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}} \tag{7.10}$$

Aus den Formeln 7.9 und 7.10 geht hervor, das der Wert K_P in die Reglerteile I und D mit einfließt. Des Weiteren wird T_N zu I im Kehrwert verwendet.

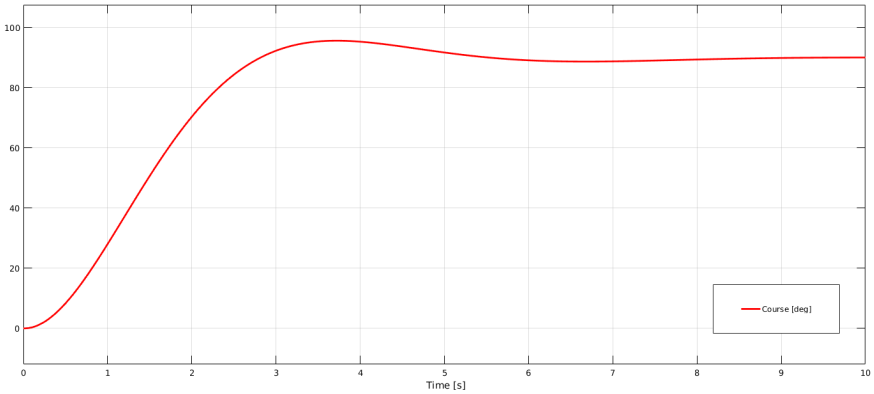


Abbildung 7.69: Simulierter Kursverlauf mit angepassten Parametern

Der Parameter N stellt einen Filterkoeffizienten dar, welcher bei dieser Verwendung zu vernachlässigen ist.

Abbildung 7.69 zeigt nun den Kursverlauf mit angepassten Parametern. Hier ist ein stabiles Systemverhalten zu sehen. Der Endwert wird nach einem leichten Überschwingen schnell eingeregelt. Die angepassten Reglerparameter sind in Tabelle 7.4 aufgeführt.

Reglerparameter	Wert
P	2,2
I	0,5
D	1,15

Tabelle 7.4: Angepasste Kursreglerparameter

Simulation der Geschwindigkeit

Um das Verhalten der Fuzzy-Logik zu simulieren wird, eine Streckenänderung von 100 m auf 0 m in den Regler eingespeist.

Abbildung 7.70 zeigt die zuerst ansteigende Geschwindigkeit im Zusammenhang mit der Regelstrecke und dem Ausgang des Reglers. Mit fallender Entfernung nimmt auch die Geschwindigkeitsvorgabe ab. Die Regelstrecke reagiert entsprechend und verliert ebenfalls Geschwindigkeit.

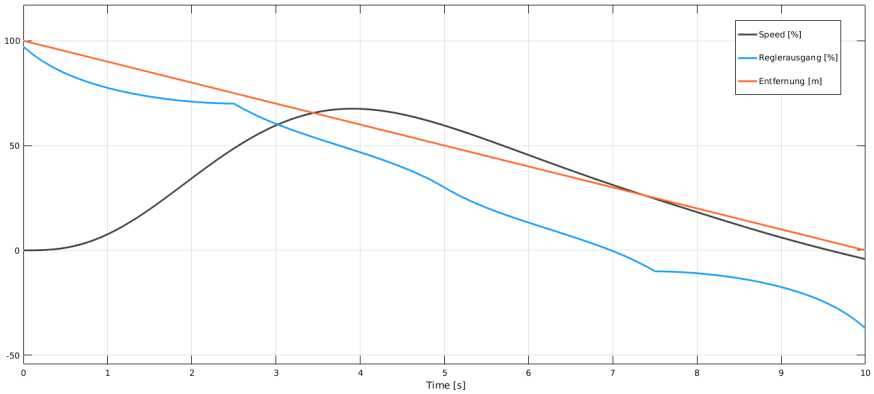


Abbildung 7.70: Simulierter Geschwindigkeitsverlauf

7.5.5 Vergleich der Parametersätze

Die Parameter der neuen Regelung wurden in das bisherige System eingegeben. Die entworfene Regelungsstruktur wurde aufgrund von Zeitmangel nicht übernommen. Daher gilt es zu beweisen, dass die Parameter eine Verbesserung des Fahrverhaltens darstellen. Um das Fahrverhalten vergleichen zu können, wird eine Mission mit vier Wegpunkten geplant, durchgeführt und aufgezeichnet. Durch das Aufzeichnen der Fahrt mit den verschiedenen Parametersätzen unter ähnlichen Umweltbedingungen, kann ein Vergleich durchgeführt werden.

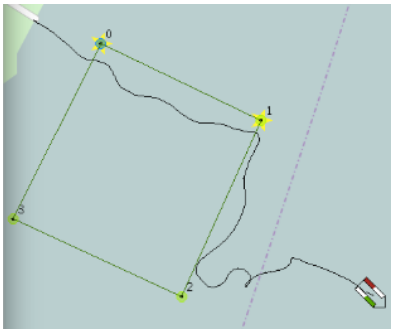


Abbildung 7.71: Fahrverhalten mit dem alten Parametersatz

Abbildung 7.71 zeigt das Fahrverhalten mit dem bisherigen Parametersatz. Hier ist eindeutig zu sehen, dass nach dem Anfahren des vorletzten Wegpunktes die Regelung instabil wird und sich das Wasserfahrzeug weit aus dem Missionsbereich entfernt, sowie nicht wieder in diesen eintritt. Mit diesem Parametersatz konnte die Mission nicht erfolgreich beendet werden.

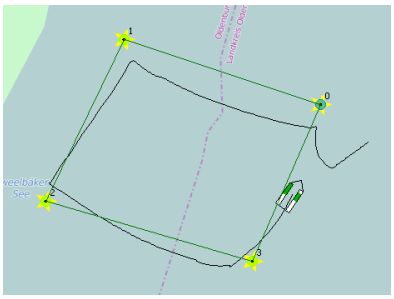


Abbildung 7.72: Simuliertes Fahrverhalten mit dem neuen Parametersatz

Das simulierte Fahrverhalten mit den hier ermittelten Parametern wird in Abbildung 7.72 gezeigt. Hier ist zu sehen, dass ohne Schlingerfahrten alle Missionspunkte angefahren werden. Des Weiteren wird die Mission erfolgreich beendet. In einem Praxistest wurde dieser Parametersatz überprüft. Leider konnte das in Abbildung 7.72 nicht rekonstruiert werden, da die Umwelteinflüsse ein inkorrektes Fahrverhalten hervorriefen. Um unter den gegebenen Umweltbedingungen ein autonomes System nachweisen zu können, wurde ein empirisches Einstellverfahren angewendet. Hierbei wurde zuerst der P-Parameter erhöht bis ein stabiles Fahrverhalten eintritt.

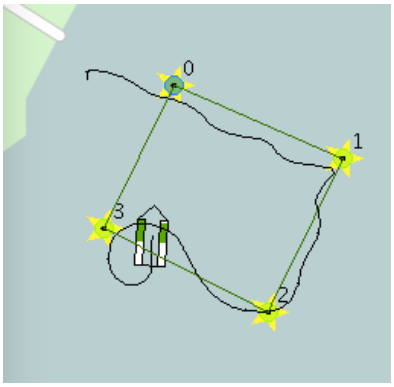


Abbildung 7.73: Fahrverhalten mit stabilem P-Parameter

Abbildung 7.73 zeigt das Kursverhalten mit einem reinen P-Regler. Der Parameter P wurde von Null bis auf 1.0 erhöht.

Anschließend wurde der D-Parameter erhöht bis ein optimales Fahrverhalten einstellt. Dieses war bei $D = 0.25$ der Fall. Abbildung 7.74 zeigt dieses. Abschließend galt es den I-Parameter zu ermitteln.

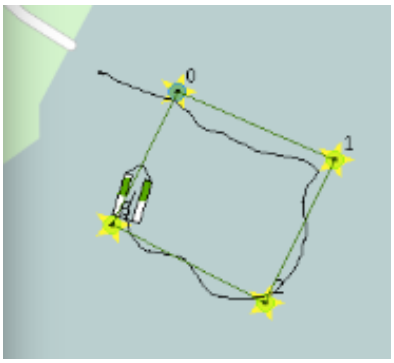


Abbildung 7.74: Fahrverhalten mit stabilem D-Parameter

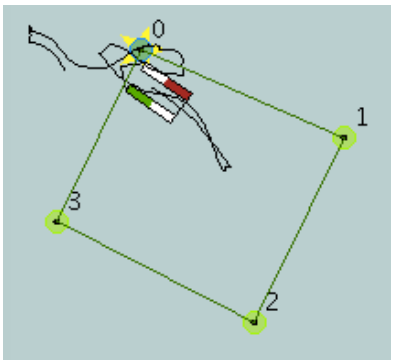


Abbildung 7.75: Fahrverhalten mit I-Parameter

Abbildung 7.75 zeigt die Auswirkungen eines I-Anteils im Reglerparametersatz. Daher wurde ein PD-Regler eingesetzt um unter den gegebenen Bedingungen Testfahrten durchführen zu können. In Abbildung 7.74 wird gezeigt, dass mit dem neuen Parametersatz alle Missionspunkte angefahren werden und das Boot nach Erreichen des letzten Missionspunkt wieder Kurs auf den ersten Missionspunkt nimmt. Hier wird gezeigt, dass die Regelung während des gesamten Missionszeitraums alle aufgetretenen Störgrößen stabil ausregeln konnte und die Mission erfolgreich beendet wurde. Das Einstellverfahren nach Ziegler und Nichols konnte nicht erfolgreich umgesetzt werden, da ein PD-Regler erforderlich war. Die Einstellregeln boten diese Möglichkeit nicht. Daher ist die bisherige Arbeit mit dem Einstellverfahren dazu zu sehen, die Parametergrößen grob abschätzen zu können. Dieses konnte erfolgreich umgesetzt werden.

Kapitel 8

Evaluierung

Das Kapitel Evaluierung befasst sich mit vier zentralen Themen: Zunächst werden die Kamera und der Laser bzgl. der Tauglichkeit für eine Kollisionserkennung gegenübergestellt. Der zweite Abschnitt geht auf das Laufzeitverhalten des Systems ein. Hier werden verschiedene Softwarekomponenten und ihre Laufzeit betrachtet, um eine Aussage über die Tauglichkeit in einem zeitkritischen Kollisionserkennungssystem zu treffen. Der dritte Abschnitt erläutert technische Grenzfälle. Speziell werden Probleme erläutert, die sich während der Praxistests zeigten und in Zukunft eliminiert werden sollten. Die letzten beiden Kapitel gehen auf die Überprüfung der Anforderungen bzw. des Meilensteinplans ein.

8.1 Vergleich von Kamera und Laser

Tabelle 8.2 zeigt eine Übersicht über die Vor- und Nachteile einer Kamera oder eines Laserscanners für den Einsatz innerhalb einer Kollisionserkennung. Im Vergleich zum Lidar ist eine Kamera in der Anschaffung deutlich günstiger und es besteht die Möglichkeit Farbinformationen aus den Bildern zu verwenden. Dies kann zum Beispiel für die Seezeichenerkennung interessant sein. Das Lidar hingegen erlaubt eine sehr schnelle Berechnung von Hindernissen, was für eine Echtzeitanwendung von enormer Bedeutung ist. Die bereitgestellten Messwerte sind sehr genau, sodass ein Arbeiten im Millimeterbereich ohne Weiteres möglich ist. Darüber hinaus ist die Objekterkennung in der Umsetzung deutlich einfacherer, da keine komplizierten Bildverarbeitungsalgorithmen verstanden und implementiert werden müssen.

Nachteilig bei der Verwendung der Kamera ist, dass keine Tiefeninformationen vorhanden sind oder nicht ohne großen Aufwand verwendet werden können. Dies ist insbesondere bei der Entfernungsabschätzung eines Hindernisses zum Wasserfahrzeug problematisch. Ein weiteres ernst zu nehmendes Probleme ist die lange Berechnungszeit der Bildverarbeitungsalgorithmen, die auf dem Raspberry Pi im Sekundenbereich liegt. Dies macht eine Echtzeitan-

Kamera	Lidar
Vorteile	
• Kostengünstige Anschaffung (74.89 Euro, vgl. [33]) • Liefert auch Farbinformationen	• Berechnung von Objekte sehr schnell möglich • Messwerte sehr genau • Objekterkennung relativ einfach
Nachteile	
• keine Tiefeninformationen • Algorithmen sehr rechenintensiv	• Messwerte nur in einer Ebene • Kostspielige Anschaffung (ca. 3700 Euro, [104]) • Reflexion problematisch

Tabelle 8.2: Vergleich von Lidar und Kamera

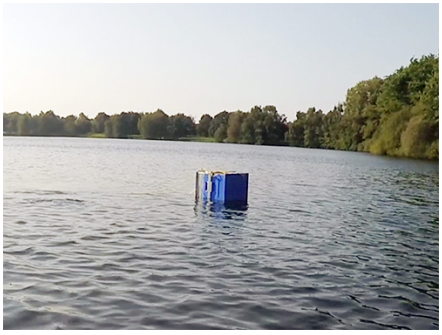
wendung unmöglich. Die Implementierung eines stabilen Objekterkennungsverfahren, dass auf Bilddaten basiert, ist bisher noch nicht gelungen, da die verschiedenen Beleuchtungssituationen und immer wechselnden Reflexionserscheinungen (sind Wolken am Himmel oder nicht) nur sehr schwer in einem universal einsetzbaren Verfahren abbildbar sind. Dennoch ist eine solide Basis gelegt worden, die für verschiedene Szenarien sehr gute Ergebnisse liefert.

Der Laserscanner weist mit einem Anschaffungspreis von ca. 4500 Euro einen hohen finanziellen Aufwand auf. Ein Nachteil dieses System besteht darin, dass die Messwerte nur in einer Ebene ermittelt werden können, sodass der Scanner relativ nah an der Wasseroberfläche angebracht werden muss, um kleine Hindernisse erkennen zu können. Des Weiteren liefern spiegelnde Flächen, die stark reflektieren, keine verwertbaren Messergebnisse, sodass hier ein gewisses Risikopotential vorhanden ist.

Um einen Einblick und Vergleich der beiden Systeme im Bezug auf die Genauigkeit zu erhalten, folgt ein qualitativer Vergleich der beiden Systeme. Hierfür werden verschiedene Hindernisse vor den Systemen positioniert und die Erkennungsqualität verglichen.

Aus ersten Tests und dabei aufgenommenem Videomaterial wurden Szenarien extrahiert, die für den Einsatz auf dem See für die Erfüllung der Anforderungen relevant schienen. Diese Szenarien waren insbesondere das *Fahren mit Gegenlicht* und das *Fahren ohne Gegenlicht*.

Mit dem *Fahren mit Gegenlicht* waren primär Szenarien gemeint, in denen mindestens eine größere Reflexion auf der Wasseroberfläche zu sehen ist. Die Stärke der Reflexion hat während der Tests unvermeidlich variiert. *Fahren ohne Gegenlicht* wurde bei bedecktem Himmel getestet. Die für die Evaluierung ausgewählten Testszenarien sind in Abb. 8.1 graphisch dargestellt.



(a) Szenario „*Blaue Kiste*“

(b) Szenario „*Rotes Boot*“

(c) Szenario „*Steg mit Personen*“

(d) Szenario „*Ufer*“

Abbildung 8.1: Ausgewählte Testszenarien

Die Ergebnisse der ersten Tests sind rein richtungsweisender Natur und sollten keinesfalls als statistische Untersuchung und Auswertung angesehen werden. An dem Einsatzort identische Testbedingungen herzustellen ist nahezu unmöglich, sodass bei den Tests innerhalb der Gruppe auf Basis der vorliegenden Video- und Logdateien entschieden wurde, ob das Hindernis als erkannt oder nicht erkannt gewertet wird. In Tab. 8.4 sind die Testergebnisse der kamerabasierten Kollisionserkennung zusammengefasst.

Gerade das Fahren bei Gegenlicht hatte einen starken Einfluss auf die Erkennung mit Hilfe der Kamera. Durch Reflexionseffekte auf der Wasseroberfläche konnte keine zuverlässige Erkennung von Hindernissen durchgeführt werden. Bei leichtem Gegenlicht, konnten die Objekte erkannt werden. Hierbei spielte es jedoch auch eine Rolle, in welchem Winkel die Objekte aufgenom-

Umweltbedingungen	*Szenario*	*Kamera*	*LIDAR*
Fahren mit Gegenlicht	Blaue Kiste	Teilw.	Ja
	Rotes Boot	Teilw.	Ja
	Steg mit Personen	Teilw.	Ja
	Ufer	Nein	Ja
Fahren ohne Gegenlicht	Blaue Kiste	Ja	Ja
	Rotes Boot	Ja	Ja
	Steg mit Personen	Ja	Ja
	Ufer	Nein	Ja

Tabelle 8.4: Testergebnisse der kamerabasierten Kollisionserkennung

men wurden und welche Distanz zwischen dem Boot und dem Objekt lag. Der Laserscanner hatte mit Gegenlicht keine Probleme.

Beide Systeme haben ohne irritierende Lichtverhältnisse zuverlässig gearbeitet. Bei der kamerabasierten Objekterkennung, konnte es teilweise dazu kommen, dass das Objekt für 1-2 Sekunden nicht mehr erkannt wurde, danach jedoch erneut als Hindernis vom System wahrgenommen wurde. Der implementierte Ansatz ist im nächsten Schritt zu optimieren und mehr auf die Umwelteinflüsse anzupassen.

Nach einem Praxistest mit welligem Wasser wurde deutlich, dass der Laserscanner mit Wellengang teilweise Probleme zu haben schien. Durch das Auf- und Abschwanken des Bootes hat der Laser teilweise zu hoch gemessen, sodass es bei einem niedrigen Hindernis, wie der blauen Tonne, zu Fehlern kam. Auch der flache Steg wurde nicht zuverlässig erkannt, jedoch aber die Personen, die auf dem Steg standen. Uferbereiche wurden zuverlässig erkannt. Eine möglichst dichte Positionierung an der Wasseroberfläche hat die Ergebnisse dahingehend verbessert. Um den Laser nicht unnötigen Gefahren auszusetzen, wurde jedoch auf eine Anbringung, die über den Bootsrand hinaus geht verzichtet. Das Umdrehen des Lasers brachte die besten Ergebnisse, da der Laser somit bis zur Oberkante des Bootes abgesenkt werden konnte, wodurch eine tiefe Scanebene erzeugt werden konnte.

Abschließend ergibt sich für die Projektgruppe folgendes Fazit: Die Implementierung der laserbasierten Kollisionserkennung war nicht nur in der Umsetzung deutlich unproblematischer, sondern lieferte auch aufgrund der hohen Genauigkeit des Sensors zuverlässige Ergebnisse, die eine für die Praxis taugliche Hinderniserkennung ermöglichen. Für eine in Zukunft nötige Verkehrsschilderkennung ist der kamerabasierte Ansatz dennoch von beson-

derem Interesse und kann für nachfolgende Projektgruppen, die ihre Fokus auf ein solches Themengebiet legen, ein geeigneter Einstiegspunkt sein.

8.2 Laufzeitverhalten des Gesamtsystems

Verschiedene Softwarekomponenten des entwickelten Systems unterliegen zeitkritischen Grenzen. Das folgende Kapitel gibt einen Eindruck von den Berechnungszeiten der unterschiedlichen Komponenten. Insbesondere werden drei Teilbereiche betrachtet:

1. Die Routenplanung ist eine zentrale Komponente, die für die Kollisionsvermeidung enorme Bedeutung hat. Jedes Mal, wenn ein neues Hindernis detektiert wird, ist eine Überprüfung der Route und gegebenenfalls eine Neuberechnung nötig. Idealerweise findet die Berechnung in sehr kurzer Zeit statt, sodass der Kurs direkt korrigiert werden kann. Die Routenplanung ist demnach ein sehr zeitkritisches Ereignis.

2. Die kamerabasierte Kollisionserkennung ist ebenfalls auf eine sehr kurze Berechnungszeit angewiesen, um eine drohende Kollision zu verhindern. Hier werden Zeitbetrachtungen für die Horizonterkennung, die Reflexionsreduktion und das Canny-Edge-Verfahren zur Objekterkennung durchgeführt.

3. Die Objekterkennung per Lidar unterliegt ebenfalls zeitkritischen Anforderungen, daher wird auch für dieses Verfahren eine Zeitbetrachtung angegeben.

Die Berechnungen wurden für einen Vergleich auf verschiedenen Rechnern durchgeführt, um die Hardwareabhängigkeit zu verdeutlichen. Die Ergebnisse sind in Tabelle 8.6 dargestellt.

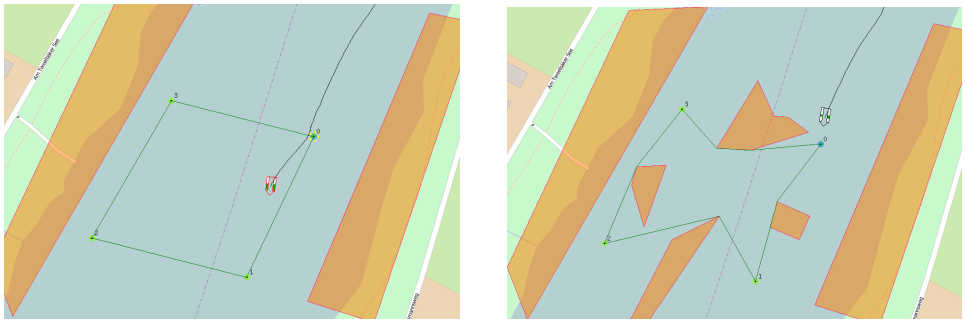
Für die Berechnung der zu fahrenden Route wird ein Graphalgorithmus eingesetzt, der je nach Komplexität der Route und zugrunde liegendem Rechner unterschiedliche Laufzeiteigenschaften aufweist. Abbildung 8.2(a) zeigt eine relativ einfache Mission, die aus vier Wegpunkten und zwei Gefahrenzonen besteht. In Tabelle 8.6 Abschnitt eins ist die Berechnungszeit für diese Mission dargestellt. Hierbei zeigt sich, dass für das Erzeugen des Graphen auf einem Macbook 25 ms benötigt werden, wohingegen der Raspberry 430 ms mit der Berechnung verbringt. Ähnlich sieht das Bild bei der Berechnung der Route aus, hier benötigt das Macbook 71 ms und der Raspberry Pi 774 ms. Offenbar ist aufgrund der schwächeren Hardware des Raspberry Pis eine deutlich höhere Berechnungzeit notwendig, was für eine echtzeitfähige Kollisionsvermeidung ein kritisches Maß einnimmt.

In Abb. 8.2(b) ist eine etwas kompliziertere und realistischere Mission dargestellt. Hierbei können die vier Gefahrenzonen in der Mitte als mögliche

	Berechnungszeit [ms]	
	Macbook	Raspberry
Grapherzeugung	25	430
Routenplanung	71	774
Grapherzeugung kom.	810	1422
Routenplanung kom.	878	9433
Horizonterkennung	9,1	1189
Reflexionsreduktion	42,4	5431
Canny-Edge-Verfahren	6,9	67
Oberjekterkennung (Lidar)	1,8	10

Tabelle 8.6: Laufzeitverhalten auf verschiedenen Rechnern

dynamische Hindernisse aufgefasst werden. Die Berechnungzeiten für Graph und Route (vgl. Tabelle 8.6, zweiter Abschnitt) steigen bei der leicht erhöhten Komplexität enorm an. Während das Macbook bei Zeiten von 810 ms und 878 ms liegt, ist der Raspberry Pi bereits bei Zeiten von nahe 10 s und 1.5 s.



(a) Mission mit zwei Gefahrenzonen (b) Mission mit sechs Gefahrenzonen

Abbildung 8.2: Missionen unterschiedlicher Laufzeitkomplexität

Aus diesen Ergebnissen lässt sich folgern, dass die eingesetzte Hardware für einen Realeinsatz zu schwach ist, da die Berechnungszeiten nicht mehr akzeptabel sind, um einem dynamischen Hindernis in kürzester Zeit auszuweichen. An dieser Stelle kann zum einen der Hardwareeinsatz erhöht werden, zum anderen aber auch der eingesetzte Graphalgorithmus analysiert werden. Dieser wurde vollständig von der vorangegangenen Projektgruppe übernommen und nicht auf seine Performanz überprüft.

Für die Berechnung von Hindernissen auf Kamerabasis wurden ebenfalls Laufzeiteigenschaften untersucht, die in Tabelle 8.6 Abschnitt drei dargestellt sind. Von besonderem Interesse sind das Berechnen des Horizonts, die Reflexionsreduktion und das Canny-Edge-Verfahren zur Objekterkennung. Auch hier zeichnet sich ein sehr hardwareabhängiges Ergebnis ab. Während das Macbook Rechnenzeiten von 6.9 bis 42.4 ms benötigt, liegt der Raspberry Pi bereits bei Zeiten zwischen 67 bis 1189 ms. Auch hier wird die hohe Hardwarerelevanz deutlich.

Die Objekterkennung per Laser ist in Tabelle 8.6 dargestellt. Hier liegen die Berechnungszeiten bei 1.8 ms bzw. 10 ms. Es ist zwar ein deutlicher Unterschied zu erkennen, allerdings liegen die Zeiten in einem akzeptablen Bereich für die Weiterverwendung, sodass der Laserscanner als eine praxistaugliche Variante für die Kollisionserkennung angesehen werden kann.

Die verwendeten Rechner unterscheiden sich in allen Hardwarebauteilen, daher ist in Tabelle 8.8 eine Übersicht der Details dargestellt.

	Eingesetzter Rechner	
Merkmal	Macbook	Raspberry
Anzahl Kerne	4	4
Leistung pro Kern	1,7 GHz	1,2 GHz
Grafikkarte	Intel HD Graphics 5000	Dual Core VideoCore
Arbeitsspeicher	4 GB	1 GB
Betriebssystem	OS X El Captain	Raspbian

Tabelle 8.8: Details zu verwendeten Rechnern

8.3 Technische Grenzen

Während der Testphasen am Tweelbäker See sind einige technische Schwachstellen des Wasserfahrzeugs zum Vorschein gekommen, welche bei einer Weiterführung des Projektes beachtet und nach Möglichkeit eliminiert werden sollten.

Bei mehreren Testfahrten wurde durch starken Wind auf der Wasseroberfläche des Sees eine deutlich spürbare Oberflächenströmung erzeugt. Diese hatte einen spürbaren Einfluss auf das Fahrverhalten des Wasserfahrzeugs. Stellenweise wurde selbst bei voller Drehzahl der Motoren nur noch ein minimaler Vorschub erzeugt. Dies führte dazu, dass die Regelung des Bootes kaum verwertbare Ergebnisse lieferte und ein autonomes Fortbewegen un-

möglich machte. Auch mithilfe der Fernbedienung gelang es nur mit Mühe, das Wasserfahrzeug zurück zum Steg zu navigieren.

Ein Weiteres Problem findet sich in dem Verwenden von Otterboxen. Da diese auf dem gesamten Boot verteilt sind, müssen häufig neue Kabel zwischen den einzelnen Boxen verlegt werden. Darüber hinaus ist der Verschluss über Gummi-Schnallen realisiert. Diese müssen oft mühsam geöffnet und geschlossen werden, was auf dem Wasser eine nicht ungefährliche Tätigkeit ist, da die meisten Gerät sehr wasseranfällig sind. Die Abdichtung der Otterboxen ist ebenfalls sehr fraglich, da die Dichtungen in den Deckeln häufig verrutschen und herausfallen. Eine Alternative zu den Otterboxen kann ein einfacher Schaltschrank darstellen. Dieser ist in beliebigen Größen erhältlich und ebenso in verschiedenen Schutzklassen (z.B.: IP68). In der Regel können Schaltschränke leicht geöffnet werden und bei geeigneter Größe finden alle Komponenten darin Platz.

Das Verwenden von XBee als Datenübertragungsstandard zur Onshore-Software hat sich als nicht ideal erwiesen, denn beim Übertragen größerer Datenmenge, wie z.B. dynamischen Hindernisse, gab es immer wieder Verbindungsausfälle. Hier ist möglicherweise ein Umdenken nötig. Vielleicht sollte einem LTE-Modul der Vorzug gegeben werden. Dies bedarf allerdings einer intensiven Analyse und Überprüfung.

Wie Abschnitt 8.2 belegt, ist das Laufzeitverhalten des Raspberry Pi bzgl. der Bildverarbeitung und Routenberechnung kritisch. Ein Echtzeitverhalten kann nicht erreicht werden, sodass die Hardware an Bord des Wasserfahrzeugs aufgestockt werden muss. Hierzu eignen sich insbesondere leistungsstarke Rechensysteme, die sich auch in einem Schaltschrank verbauen lassen.

8.4 Revision der Anforderungen

Während der anfänglichen Projektplanung wurden Anforderungen definiert, die die zu erfüllenden Eigenschaften des Systems beschreiben. Im Rahmen der Evaluation können diese Anforderungen zur Bewertung des Systems herangezogen werden.

Im Verlauf des Projektes wurde der planmäßige Fortschritt mehrfach von unvorhergesehenen Ereignissen ausgebremst. Durch zielorientiertes Planen und Handeln wurden die Anforderungen stets im Blick behalten und unterschiedlich priorisiert. Das primäre Ziel der Projektgruppe war es, eine funktionierende Kollisionserkennung und -vermeidung zu realisieren. Anforderungen, die diese Kernkompetenz ermöglichen, wurden für die Bearbeitung höher priorisiert. Ergänzende Anforderungen, die das Einsatzgebiet zwar erweitert, jedoch nicht das primäre Ziel der Projektgruppe verfolgt hätten entsprechend niedriger priorisiert.

Anpassungen der zeitlichen Planung wurden auch immer in Hinblick auf die Kompetenzen der einzelnen beteiligten Mitglieder der Gruppe vorgenom-

men, sodass es teilweise zu Überschneidungen kam, da beispielsweise Mitglieder der Gruppe „Regelung“ benötigt wurden, um die aufgetretenen elektrischen Probleme des bestehenden Systems zu untersuchen und zu beseitigen.

Im folgenden Abschnitt werden die definierten Anforderungen auf ihren Grad an Erfüllung im Rahmen der Projektgruppe analysiert. Hierbei kann den Anforderungen der Status *Erfüllt*, *Teilweise erfüllt* oder *Nicht erfüllt* zugeordnet werden. Sollte eine Anforderung nicht, oder nur teilweise erfüllt worden sein, so werden die Gründe für die Unvollständigkeit erläutert. Der jeweilige Status wird mit entsprechenden Piktogrammen visualisiert (*Erfüllt*: Kreis und Haken, *Teilweise erfüllt*: Gepunkteter Kreis und Haken, *Nicht erfüllt*: Gepunkteter Kreis und Kreuz).

8.4.1 Software

Anforderung	Status	Erläuterung
Das System muss die Umgebung kontinuierlich beobachten	✓	
Das Planen einer Mission soll durch die Darstellung des mathematischen Modells vereinfacht werden.	×	Aufgrund der Probleme des bestehenden elektrischen Systems fokussierte sich die Gruppe zunächst auf den Neuaufbau der Bootskomponenten, die in ihrer Vollständigkeit die Basis für das mathematische Modell bilden.
Das Planen einer Mission soll durch die automatische Kennzeichnung von Gefahrenzonen vereinfacht werden.	✓	
Eine erstellte Mission muss vom Nutzer gespeichert werden können.	✓	
Eine Mission muss auf einen externen Datenträger exportiert werden können.	✓	

Anforderung	**Status**	**Erläuterung**
Eine gespeicherte Mission muss geladen werden können.	✓	
Eine exportierte Mission muss importiert werden können.	✓	
Eine zu startende Mission muss valide sein.	✓	
Zum Starten einer Mission muss eine Verbindung zum Boot bestehen.	✓	
Eine gestartete Mission muss abgebrochen werden können.	✓	

Anforderung	**Status**	**Erläuterung**
Nach dem Abbruch einer Mission muss das Boot zur Basis zurückkehren.	✓	
Der Pfad zur Basis muss valide sein.	✓	
Nach dem Abschluss einer Mission sollen die aufgenommenen Sensordaten dem Nutzer zum Export bereitstehen.	×	Nach anfänglichen Tests stellten sich die Sensoren als nicht betriebsbereit heraus, und da der Fokus auf der Implementierung einer Kollisionserkennung und -erhütung lag wurde die Funktion nicht realisiert.
Der aktuelle Zustand des Bootes muss durch den Nutzer überwacht werden können.	✓	
Der Soll-Kurs muss überwacht werden können.	✓	

Anforderung	**Status**	**Erläuterung**
Der Ist-Kurs muss überwacht werden können.	✓	
Der Missionsfortschritt muss überwacht werden können.	✓	
Die Geschwindigkeit muss überwacht werden können.	✓	
Der verbleibende Kapazität der Batterie muss überwacht werden können.	✗	Die personellen Ressourcen waren aufgrund der Verzögerung durch den Systemumbau nicht verfügbar.
Die verbleibende Laufzeit der Batterie muss überwacht werden können.	✗	s.o.

Anforderung	Status	Erläuterung
Die abgeschätzte Zeit für den Weg zurück zur Basis muss überwacht werden können.	✗	s.o.
Die Aktivität des Bootes muss überwacht werden können.	✓	
Die Wetterdaten am Ort des Bootes müssen überwacht werden können.	✗	Schon frühzeitig wurde der Fokus auf die Kollisionserkennung und -verhütung verschoben, weswegen die Anforderungen nicht realisiert wurde.
Mögliche Fehlerzustände müssen überwacht werden können.	✓	
Die Signalstärke der XBee-Verbindung muss überwacht werden können.	✗	Im Laufe des Projektes zeigte sich, dass XBee eine zu geringe Bandbreite aufwies, weswegen als Alternative eine WLAN-Übertragung implementiert wurde.

Anforderung	**Status**	**Erläuterung**
Die Software soll die Positionen erkannter Hindernisse approximiert darstellen.	✓	
Nach einer Kollision soll der Nutzer benachrichtigt werden.	✗	Mit den derzeitig vorhanden Kommunikationsschnittstellen nicht möglich. Hierfür ist beispielsweise ein LTE-Modul denkbar.
Das System soll eine Schnittstelle bereitstellen, die es dem Entwickler ermöglicht, ausgewählte Systemparameter zu verändern.	✓	
Zu Testzwecken muss die Software über eine Möglichkeit verfügen, die Hardware des Bootes auf einem Rechner zu simulieren.	✓	

8.4.2 Kollisionserkennung

Anforderung	Status	Erläuterung
Das Sytstem muss die Umgebung kontinuierlich beobachten	✓	
Das System muss den Bereich vor dem Boot beobachten.	✓	
Das System muss die Bereiche neben dem Boot beobachten.	×	Zur Erprobung der Kollisionserkennungstechnologien wurde sich auf den vorderen Bereich des Bootes beschränkt. Durch zusätzliche Scanner und Kameras kann der Radius erhöht werden.
Das System muss den Bereich hinter dem Boot beobachten.	×	s.o.
Das System muss Objekte erkennen können.	✓	

Anforderung	**Status**	**Erläuterung**
Das System muss Objekte identifizieren können.	✗	Der Kamerabasierte Ansatz, der für die Identifikation notwendig gewesen wäre, wurde im Laufe des Projektes durch den Einsatz des Laserscanners nicht weiter ausgebaut.
Das System muss Seezeichen identifizieren können.	✗	Da die Kollissionserkennung einen Großteil der personellen Ressourcen in Anspruch nahm, wurde das Identifizieren einzelner Seezeichen mit niedrigerer Priorität bewertet.
Das System muss statische Hindernisse identifizieren können.	✓	
Das System muss dynamische Hindernisse identifizieren können.	✓	
Das System soll Domänen-unspezifisch agieren.	✓	

Anforderung	**Status**	**Erläuterung**
Das System muss objektspezifische Parameter erkennen.	✓	
Das System muss die Größe von Objekten erkennen.	✓	
Das System muss die Entfernung zu Objekten erkennen.	✓	
Das System soll die Geschwindigkeit eines Objekts bestimmen.	×	Aufgrund mangelnder personeller Ressourcen, wurde auf die Umsetzung weiterer objektspezifischer Parameter verzichtet.
Das System soll den Kurs eines Objekts abschätzen.	×	s.o.

Anforderung	**Status**	**Erläuterung**
Das System muss abwägen, ob eine Kollision wahrscheinlich ist.	✓	
Bei der Erkennung eines Seezeichens soll das System einen Hinweis an die Bahnplanung weiterleiten.	×	s.o.
Bei dem Aktivieren der Fernsteuerung muss die Kollisionserkennung deaktiviert werden.	✓	

8.4.3 Kursberechnung und Bahnplanung

Anforderung	Status	Erläuterung
Der berechnete Pfad muss valide sein.	✓	
Der berechnete Pfad muss gespeichert werden.	✓	
Nach einem Neustart des Systems muss der gespeicherte Pfad erneut aufgerufen werden können.	✓	
Der Algorithmus für die Bahnplanung muss Hindernisse einbeziehen.	✓	
Der Algorithmus muss statische Hindernisse einbeziehen.	✓	

Anforderung	**Status**	**Erläuterung**
Der Algorithmus muss dynamische Hindernisse einbeziehen.	✓	
Das Boot soll die Position halten, wenn durch den Bahnplanungsalgorithmus kein valider Pfad gefunden werden kann.	✓	
Der Algorithmus für die Bahnplanung muss vollständig simuliert werden können.	✓	
Das System muss den Kurs beibehalten, wenn es zu keiner Kollision kommen wird.	✓	
Das System muss selbstständig kollisionsverhütende Maßnahmen einleiten können, wenn es zu einer Kollision kommen würde.	✓	

8.4.4 Erkennen von physikalischen Grenzfällen

Anforderung	Status	Erläuterung
Die Plausibilität des GPS-Signals soll anhand bereits vorhandener Sensoren überprüft werden.	⊗	Aufgrund mangelnder personeller Ressourcen, wurde auf die Erkennung physikalischer Grenzfälle verzichtet. Primär wurde ein stabiles Gesamtsystem angestrebt.
Der Bahnplanungsalgorithmus soll die Richtung und die Geschwindigkeit der Strömung berücksichtigen können.	⊗	s.o.
Der Bahnplanungsalgorithmus soll die Richtung und die Geschwindigkeit des Windes berücksichtigen können.	⊗	s.o.

8.4.5 Regelung und mathematisches Modell

Anforderung	Status	Erläuterung
Das System muss über geeignete Regler verfügen.	✓	
Der Regler muss über Parametersätze verfügen, der unter Berücksichtigung möglicher Störgrößen ein stabiles System garantiert.	✓	
Das mathematische Schiffsmodell soll die physikalischen Eigenschaften des Bootes modellieren.	✓	Die Realisierung eines vollständigen mathematischen Systems wurde aufgrund des ungeplanten Systemumbaus zunächst geringer priorisiert.

8.5 Abgleich des Meilensteinplans

Die ursprüngliche Projektplanung sah zwölf Sprints mit einer Länge von jeweils drei Wochen vor (vgl. 2.2).

8.5.1 Meilenstein 1

In dem ersten Meilenstein wurden theoretische Grundlagen und Konzepte erarbeitet, die als solide Basis für die weiterführenden Arbeiten in Richtung Kollisionserkennung und Kollisionsverhütung dienten. Durch eine umfangreiche IST-Analyse und erste Tests, wurden diverse Problemstellen identifiziert, die zu einer Abweichung in Teilgebieten vom ursprünglichen Meilensteinplan führten. Im nachfolgenden Teil werden die einzelnen Gruppen zusammenfassend in ihrer Arbeit betrachtet und die aufgetretenen Abweichungen erläutert. Die Sprints 4 bis 6 wurden von dem ersten Meilenstein beendet.

Software I

Die Gruppe *G1 - Software I* war dafür zuständig, den IST-Zustand der Software zu analysieren, das Build System von Ant auf Maven umzustellen, Git als Versionierungssystem einzuführen und die Kartendaten zu parsen, um Gefahrenzonen automatisiert erkennen zu können. Weiterhin wurde geplant, die Hardware-Simulation um einige Parameter zu erweitern, wie beispielsweise Wind oder Strömung. Der letzte Punkt wurde im weiteren Verlauf der Phase innerhalb der Gruppe genauer betrachtet und anschließend verworfen. Eine Parametrisierung der Simulation wurde im Rahmen des Projektes als nicht zielorientiert erachtet. Der Fokus der Arbeit wurde nach Rücksprache mit den Stakeholdern klar auf die Erkennung von Hindernissen sowie die Kollisionsverhütung gerichtet, sodass eine weitere Erfassung von Umweltparametern wie Windstärke oder Strömung als irrelevant für dieses Projektziel gewertet wurde.

Hardware I

Durch frühe Untersuchungen der Gruppe *G2 - Hardware I* zeigte sich, dass die bestehende Bootsarchitektur verändert werden musste. Der aktuelle Aufbau hatte zahlreiche Fehlfunktionen und Verbindungsabbrüche sowie Systemneustarts zur Folge. Statt den Fokus weiterhin auf die Anbringung neuer Sensorik oder der Vervollständigung der Handbücher zu richten, wurde die Analyse des IST-Zustands dafür genutzt, ein durchdachtes Konzept zur Neuordnung der einzelnen Komponenten im Boot auszuarbeiten. Diese Entscheidung hatte eine Abweichung vom ursprünglichen Plan zur Folge, da für die angedachte Regelung zunächst das neue Layout realisiert werden musste. Die Verantwortlichen aus der Gruppe *G4 - Regelung I* haben sich mit der Gruppe *G2 - Hardware I* zusammengeschlossen, um dieses Problem gemeinsam zu lösen.

Das Konzept zur Wasserdichtigkeit wurde wie geplant erarbeitet. Die Pläne und Kosten dafür wurden dem Betreuer vorgelegt und gemeinsam Vor- und Nachteile diskutiert. Durch die ungewisse weitere Entwicklung hinsichtlich des Bootsaufbaus, der Sensorik und der Anbringung wurde die Umsetzung für diesen Moment zurückgestellt. Kleine Änderungen im Bootsaufbau hätten die Wasserdichtigkeit gefährden können, sodass sich angesichts der hohen möglichen Folgekosten durch notwendige Änderungen gegen die weitere Implementierung entschieden wurde.

Kollisionserkennung I

Die Arbeit der Gruppe *G3 - Kollisionserkennung I* war wie geplant primär theoretischer Natur. Die Systemumgebung für die Erkennung von Hindernissen wurde vom restlichen System abgegrenzt. Die Unterschiede der Systeme zur Erkennung und Verhütung wurden herausgestellt. Auf Basis der ersten Konzepte wurde eine funktionale Architektur erarbeitet, aus der notwendige Schnittstellen zwischen den neuen Systemen zur Erkennung und Verhütung und dem bestehenden System abgeleitet werden konnten. Weiterhin wurde versucht, Informationen über Kamera- oder Sensorsysteme aus verwandten Arbeiten zu extrahieren, um diese auf das Projekt anzuwenden. Diese Arbeit gestaltete sich schwieriger als zuvor angenommen, da die verwandten Arbeiten primär statische Systeme einsetzen. Somit konnte keine eindeutige Entscheidung darüber getroffen werden, welches System letztlich zum Einsatz kommen wird. Jedoch wurden bereits Konzepte zur Anbringung der Sensoren überdacht und skizziert. Die Sicherheitsanforderungen wurden aufgrund des anstehenden Systemumbaus zunächst auf einen späteren Zeitpunkt verschoben.

Regelung I

Während des ersten Meilenstein wurde die Grundlagenarbeit bzgl. der Regelung durchgeführt. Neben der Literaturrecherche wurde auch nach Projekten gesucht welche ein Ähnliches Implementierungsvorhaben aufweisen. Es wurde die Verwendungsmöglichkeit der verschiedenen Reglertypen auf unseren Anwendungfall geprüft und bewertet. Durch die Modellierungsmöglichkeiten mit MATLAB/SIMULINK wurde beispielhaft der jeweilige Reglertyp in eine angenäherte Umgebung eingefügt und mit geschätzten Parametern betrieben. Dieses stellte die Grundlage zur Bewertung der Reglertypen dar. Aufgrund von einer Nutzen/Aufwandseinschätzung wurde, für die jeweiligen Anwendungen eine erste Auswahl der Reglertypen vorgenommen. Wichtiger als die erste Auswahl vorzunehmen war es die Reglertypen zu Identifizieren welche ausgeschlossen werden können. Auf diese Weise konnten die Neuronale Netze als Regler aufgrund des hohen Implementierungsaufwands ausgeschlossen werden. Abschließend wurde eine eher untypische Regler als erster Struktur-

entwurf gewählt. Hierbei galt die Idee die Regler für Kurs und Geschwindigkeit parallel zu schalten und über Summenpunkte jeweils ein Ansteuersignal für die Motoren zu generieren. Dieses Ansteuerverfahren würde durch die im Projekt verwendete Leistungselektronik unterstützt. Des Weiteren wurde die mathematische Modellbildung mittels des Nomoto first Order Ansatzes betrachtet. Sollte sich im Laufe des Projekts herausstellen, dass evtl. Beobachterstrukturen in die Regelung mit einfließen müssen, sollte dieses Modell die Grundlage bilden.

8.5.2 Meilenstein 2

Durch die identifizierten Probleme gab es erste Abweichungen, die sich auf den gesamten Projektverlauf auswirkten. Im Zusammenschluss mit dem Projektgruppenbetreuer wurde die Abweichung besprochen und die nachfolgenden Projektziele priorisiert, um den Zeitverlust durch die Probleme zu kompensieren. Primär wurden verfügbare Kräfte im Bereich *G4 - Regelung II* mobilisiert, um die Restrukturierung des Gesamtsystems zu beschleunigen. Der zweite Meilenstein bildete das Ende der Sprints 7 bis 9.

Software II

Auf die Implementierung des mathematischen Modells wurde verzichtet. Aufgrund des Systemumbaus kam es im Bereich *G4 - Regelung II* zu einem Engpass in Hinsicht auf die verfügbaren Kapazitäten. Trotz dessen konnte die Gruppe *G2 - Software II* wichtige Kommunikationskonzepte überarbeiten. Bereits zu Beginn der IST-Analyse gab es Fehlverhalten, das auf das genutzte Kommunikationskonzept zurückzuführen war. Die Implementation von XBee wurde verbessert, um das System zu stabilisieren und die notwendigen Voraussetzungen für den Einsatz der Kamera und des Laserscanners zu schaffen. Das geplante Heartbeat-Monitoring wurde auf konzeptioneller Ebene erarbeitet, jedoch ebenfalls zurückgestellt, da die Erweiterung des Kommunikationskonzeptes mehr Zeit in Anspruch nahm, als zunächst geplant war. Die Software wurde dahingehend erweitert, als dass fortan die Zustände der einzelnen Komponenten (Motordaten, GPS, IMU-Board,..) in Echtzeit überwacht werden können. Diese Implementierung war für die Testfahrten am See relevant, um das Verhalten des Bootes bei der Erkennung von Hindernissen vom Ufer mitverfolgen zu können.

Kollisionserkennung II

Der zweite Meilenstein sah vor, die notwendigen Schnittstellen für die Kollisionserkennung zu implementieren, geeignete Hardware zu testen und evaluieren und die Kollisionsverhütung vorzubereiten. Notwendige Schnittstellen für ein Kamerasystem wurden in diesem Abschnitt implementiert. Nach anfänglichen Tests, bekam die Gruppe zudem die Möglichkeit, einen Laserscanner ein-

zusetzen. Dieser wurde umfassend auf das erforderliche Szenario untersucht. Hierbei wurde festgestellt, dass der ursprünglich geplante Laserscanner untauglich war und ein weiterer Laserscanner getestet, der den Anforderungen des Projektes entsprach und letzten Endes implementiert wurde. Die Vorbereitung der Anbindung an die Kollisionsverhütung wurde auf einen späteren Zeitpunkt verschoben, da der unerwartete Einsatz eines Laserscanners eine Erweiterung der Systemstruktur mit sich brachte, die zunächst gelöst werden musste.

Hardware II

Eine umfangreiche Änderung des Plans betraf die Gruppe *G3 - Hardware II*. Sowohl die finale Umsetzung des Konzeptes zur Wasserdichtigkeit, als auch die Integration des Heartbeat-Monitoring Systems mussten dem Neuaufbau des Gesamtsystems weichen. Die sensible Elektronik im Boot wurde beispielsweise durch die unvorteilhafte Verlegung der stromführenden Leitungen gestört, sodass es zu Abstürzen und falschen Daten kam. Auch die eingesetzten Konverter, um die Spannung von 12 V auf 5 V zu regulieren waren untauglich und wurden durch neue Komponenten ersetzt. Der langwierige Bestellprozess bremste die Arbeit am Gesamtsystem zudem spürbar, sodass die verantwortlichen Gruppenmitglieder zeitweise in den anderen Gruppen mithalfen, während auf neue Bauteile gewartet wurde.

Kollisionsverhütung und Regelung I

Wie auch die Gruppe *G3 - Hardware II* betraf der Systemumbau den Fortschritt der zusammengelegten Gruppe *G4/G5 - Kollisionsverhütung und Regelung I*. Das Boot war zu dem Zeitpunkt final konstruiert, sodass eine Aufzeichnung von Regelungsparametern und die Kalibrierung der Regelung zu untauglichen Daten geführt hätten. Auch wurden die Kapazitäten der Gruppe *G2 - Kollisionserkennung II* durch den Einsatz des Laserscanners halbiert, da fortan zwei unterschiedliche Systeme implementiert wurden. Da die Priorität auf der Kollisionserkennung und diese für die Kollisionsverhütung ein hinreichendes Kriterium darstellt, wurden die verfügbaren Kapazitäten zunächst von der Regelung auf die Gruppen *G2 - Kollisionserkennung II* und *G3 - Hardware II* disponiert.

8.5.3 Meilenstein 3

Durch die Änderungen zur Erreichung des zweiten Meilensteins wurden einige Ziele in den Prozess zur Erreichung des dritten Meilensteins verschoben. Ursprünglich war der dritte Meilenstein mit dem primären Ziel der Kollisionsverhütung geplant. Neben ersten Ansätzen zur Kollisionsverhütung wurde jedoch weiterhin stark an der Kollisionserkennung gearbeitet. Der dritte Meilenstein umfasste Sprint 10 bis 12.

Software III

Begleitend unterstützte die Software Gruppe die weiteren Gruppen und implementierte relevante Softwareteile für den Einsatz der Sensoren. Die neue Hardware konnte an die Software angebunden werden, eigene Testprogramme für die Sensoren geschrieben werden und auch die relevanten Schnittstellen zur Kommunikation untereinander wurden realisiert und erweitert. Durch den ungeplanten Einsatz des Laserscanners mussten zudem weitere Änderungen im Kommunikationsdesign vorgenommen werden. Der Datenaustausch zwischen den einzelnen Komponenten wurden über TCP/IP realisiert.

Hardware III

Die Integration der neuen Sensoren in das Gesamtsystem konnte wie geplant umgesetzt werden. Nachdem der Systemumbau abgeschlossen war, bot das Gesamtsystem eine solide Basis für die weitere Integration der Komponenten für die Kollisionserkennung.

Kollisionserkennung III

In Zusammenarbeit mit den anderen Gruppen wurden wie geplant die Erkennungssysteme abschließend und umfassend in das Gesamtsystem integriert. Durch mehrfache Überprüfung in Realtests konnten die möglichen Grenzen analysiert und Testszenarien erstellt werden. In dieser Phase hat die Gruppe eng mit der Gruppe *G5 - Kollisionsverhütung und Regelung II* zusammengearbeitet, um die Ziele zur Erreichung des dritten Meilensteins zu erreichen.

Kollisionsverhütung und Regelung II

Die Kollisionsverhütung war Bestandteil der Arbeit aller Gruppen, da die Integration, die fortlaufenden Tests sowie die Anpassungen viel Kapazität in Anspruch nahm und die Zusammenarbeit der gesamten Projektgruppe erforderte. Die Regelung wurde bis zu diesem Zeitpunkt seitens der Struktur mittels MATLAB/SIMULINK validiert. Um dieses aber durchführen zu können, galt es die Systemparameter zu bestimmen. Dieses wurde durch zahlreiche Tests sichergestellt. Während dieser Tests wurden mittels Beschleunigungssensoren Kurswechsel durchgeführt und Beschleunigungsfahrten absolviert. Hieraus konnten in erster Näherung Systemparameter bestimmt werden. Diese wurden dann in Testmissionen optimiert bis der Kollisionsverhütung ein verbessertes Fahrverhalten zur Verfügung stand.

8.5.4 Meilenstein 4

Ursprünglich waren für den vierten Meilenstein weitere Systemtests, Projektabschluss und das Anfertigen der Dokumentation geplant. Aufgrund der

vorangegangenen Anpassungen wurde fortlaufend in Sprints darauf geachtet, möglichst viele Aufgaben zur Erfüllung des vierten Meilensteins bereits Begleitend zu erledigen, sofern Kapazitäten innerhalb der Gruppe ungenutzt waren. So wurde die Dokumentation bereits während der Implementierung ergänzt und auch die einzelnen Komponenten wurden zur Optimierung fortlaufend getestet. Der vierte Meilenstein umfasste Sprint 13 bis 15.

Kapitel 9

Projektabschluss

Das nachfolgende Kapitel fasst die Ergebnisse der Projektgruppe in einem Fazit zusammen. Außerdem wird ein Ausblick für verschiedenen Projektaspekte gegeben, der eine mögliche Gliederung für die Weiterentwicklung des Systems aufzeigt und zukünftige Optimierungsziele identifiziert.

9.1 Fazit

Das Ziel des ersten Semesters war es, ein grundlegendes Verständnis in der Thematik des autonomen Wasserfahrzeugs zu erlangen, den aktuellen Entwicklungsstand nachzuvollziehen, mögliche Verbesserungspotentiale zu analysieren sowie entsprechende Anforderungen zu erheben. Aufbauend darauf sollten Konzepte zur Realisierung für die nachfolgende Praxisphase erarbeitet werden. Um die vielschichtigen Aufgaben im Rahmen des Projektvorhabens effektiv bearbeiten zu können, wurde zudem jedem Gruppenmitglied eine ausgewählte Rolle zugewiesen, für die er tiefere Kenntnisse erlangte und als Ansprechpartner innerhalb der Gruppe fungierte.

Im Anschluss an die Erarbeitung der Seminarausarbeitungen wurde auf Basis der erlangten Informationen eine Projektvision erstellt, die erste Projektvorstellungen und Ziele beinhaltete. Im Fokus der anschließenden Bearbeitungsphase standen die Themengebiete *Kollisionserkennung*, *Kollisionsverhütung*, *Regelung*, *Bootsaufbau und Seetauglichkeit* sowie die *Ist-Analyse des aktuell eingesetzten Systems*, die aus den individuellen Interessen der Gruppenmitglieder und den in der Projektvision identifizierten Problemen in Abhängigkeit von den verfügbaren Kapazitäten abgeleitet wurden.

Mit der Abgabe des Zwischenberichts konnte die Phase der Anforderungserhebung abgeschlossen werden. Fehleinschätzungen bei der Planung traten im Rahmen der Klausurenphase auf, da der aus der Vorbereitung resultierende Arbeitsaufwand unterschätzt wurde. Hier konnten die Sprint-Ziele nicht wie geplant erreicht werden, sodass einzelne Aufgaben in die nachfolgenden Sprints verschoben werden mussten.

Um den weiteren Projektverlauf planen und einen optimalen Einsatz der zur Verfügung stehenen Ressourcen gewährleisten zu können, wurde zu Beginn des zweiten Semesters eine ausführliche Meilensteinplanung vorgenommen. Durch die Formulierung eines Sicherheitskonzepts konnten zudem die notwendigen Maßnahmen zur Sicherstellung der funktionalen Sicherheit identifiziert und die systematische Herleitung der funktionalen und technischen Systemarchitekturen vorangetrieben werden.

Die nachfolgende Projektphase befasste sich mit der technischen Umsetzung der erarbeiteten Konzepte. Neben den Kernfeldern der Kollisionserkennung und -verhütung konnte die Benutzeroberfläche der Onshore-Software grundlegend überarbeitet und eine Validierung der bisherigen Regelstruktur durchgeführt werden.

Die stetige Weiterentwicklung des Onboard-Systems über mehrere Projektgruppen hinweg hatte die eingesetzten Komponenten an die Grenzen ihrer Leistungsfähigkeit gebracht und erschwerte die Erweiterbarkeit zunehmend. Insbesondere die Instabilität der Spannungsversorgung und die Beeinflussung der Sensorsysteme durch elektrisch induzierte Felder bedingten die Dringlichkeit eines umfassenden Systemumbaus. Ein neues Leitungskonzept, die Überarbeitung der Anordnung der einbezogenen Teilsysteme, der Einsatz von Kondensatoren sowie die Erhöhung der Betriebssicherheit durch die Anbringung eines Verpolungschutzes und die Integration von An-Aus- und Notaus-Schaltern stellten die Stabilität des Gesamtsystems langfristig sicher.

Infolge der notwendig gewordenen Restrukturierung des elektrischen Systems mussten umfassende Änderungen an der zuvor erstellten Meilensteinplanung vorgenommen werden. Um den unvorhergesehenen Mehraufwand ausgleichen zu können, mussten eine strategische Neuausrichtung der Projektgruppe vorgenommen und niederpriorisierte Anforderungen in spätere Arbeitsphasen vorschoben werden. Die *Erkennung physikalischer Grenzfälle*, wie beispielsweise die Detektion starken Wellengangs oder natürlicher Umwelteinflüsse, musste im Zuge der Anpassungen verworfen werden.

Die Ziele in Hinblick auf die *Kollisionserkennung* wurden von der Projektgruppe erfüllt. Die Detektion ortsfester und beweglicher Hindernisse erfolgt unter Verwendung des LIDAR-Systems und unter realen Einsatzbedingungen äußerst zuverlässig. Die Funktionsfähigkeit der kamerabasierten Kollisionserkennung konnte unter idealen Umweltbedingungen, d.h. ohne das Auftreten starker Reflexionen auf der Wasseroberfläche, demonstriert werden. Die implementierten Algorithmen erkennen selbst komplexe Zusammenhänge zwischen einer Vielzahl heterogener Messpunkte und leiten die extrahierten Hindernis-Objekte an das Teilsystem der Kollisionsverhütung weiter.

Die Kollisionsverhütung kann – basierend auf zuvor aufgezeichneten Hindernisinformationen – vollständig in der Onshore-Software simuliert werden. Aufgrund der begrenzten Leistungsfähigkeit der eingesetzten Recheneinheiten und den resultierenden Verzögerungen bei der Berechnung der Ausweichrou-

ten konnte die gewählte Implementierung lediglich unter idealen Einsatzbedingungen ein sicheres Umfahren der erkannten Hindernisse gewährleisten.

Die Benutzerfreundlichkeit der Onshore-Software konnte nachhaltig verbessert werden. Das Parsen ausgewählter Kartendaten und die automatische Detektion relevanter Gefahrenzonen verringern den Aufwand für Erstellung eines validen Missionsplans und erleichtern die Arbeit fachfremder Spezialisten. Durch die Darstellung fortlaufend aktualisierter Informationen über die aktuellen Systemparameter können das Wasserfahrzeug und der Fortschritt während der Durchführung von Missionen engmaschig überwacht werden.

Abschließend lassen sich die besonders hohe Eigenständigkeit und Einsatzbereitschaft innerhalb der Projektgruppe hervorheben. In enger Zusammenarbeit wurden zahlreiche, größtenteils tägliche Arbeitstreffen abgehalten, in denen der Projektfortschritt neben den wöchentlichen Gruppensitzungen vorangetrieben wurde. Die offene Kommunikation auftretender Probleme und die zielorienterte Bearbeitung fachlicher Herausforderungen berücksichtigend, lässt sich das vergangene Jahr als produktiver Arbeitsprozess beschreiben, der insbesondere durch das freundschaftliche Miteinander der Gruppenmitglieder geprägt wurde. Die persönlichen Interessen mit den vorgegebenen Projektzielen vereinend, werden uns die gemeinsamen Testfahrten am Tweelbäker See sicherlich auch in Zukunft als motivierende Erlebnisse in Erinnerung bleiben.

9.2 Ausblick

Im Verlauf des Projektes haben sich zahlreiche Ansätze für weitere Forschungsfragen und zukünftige Arbeitsaufträge herausgebildet. Im nachfolgenden Abschnitt werden daher zukünftige Entwicklungsschwerpunkte für die Hauptbestandteile des von uns realisierten Gesamtsystems beschrieben.

Der Systemumbau hat grundlegende Veränderungen mit sich gebracht. Die neue Gewichtsverteilung des Bootes erfordert eine erneute Analyse und Implementation geeigneter Reglerstrukturen. Erste Untersuchungen dazu wurden bereits im Rahmen der Projektgruppe unternommen. Weitere Parameter müssen durch Praxistests erfasst und ausgewertet werden. Insbesondere infolge einer Überarbeitung des Antriebskonzepts müssten die Reglerparameter erneut identifiziert werden, um die Stabilität des Systems zu gewährleisten.

Obwohl sich die kamerabasierte Kollisionserkennung unter realen Einsatzbedingungen als ungeeignet herausgestellt hat, könnte sie in einer nachfolgenden Projektgruppe für die Identifikation von Seezeichen oder die Aufnahme ergänzender Hindernisinformationen verwendet werden. Der nächste Schritt würde eine Erkennung mittels geeigneter Algorithmen vorsehen, in dem Bilder von Seezeichen angelernt werden, um eine zukünftige Identifikation dieser Zeichen zu ermöglichen. Auch die Logik, die diese Schilder implizieren, würde in das System implementiert werden.

Neben der Verbesserung der Objekterkennung mittels der Kamera könnte eine Weiterentwicklung die Integration einer zweiten Kamera vorsehen, um einem stereoskopischen Bild Tiefeninformationen zu entnehmen. Hier ergeben sich viele Forschungsfragen, wie beispielsweise die richtige Auswahl und Anordnung der Kameras, die genutzten Algorithmen und die Umwelteinflüsse auf das Kamerabild und die Berechnung der Tiefeninformationen. Außerdem könnte durch das zusätzliche Anbringen von Kameras das Sichtfeld auf 360 ° erweitert werden. Analog könnte mit dem Laserscanner verfahren werden.

Die entwickelte Lösung der Projektgruppe setzt auf der Implementation der vorherigen Projektgruppen auf. Der entwickelte `MissionController` ist im Hinblick auf eine Kollisionsvermeidung schlecht erweiterbar, sodass eine Neuimplementation dringend nötig wäre.

Vorgesehen ist weiterhin, die XBee-Schnittstelle vollständig durch eine geeignete WLAN- und LTE-Schnittstelle zu ersetzen. Erste Entwicklungen dahingehend wurden im Rahmen der Projektgruppe mit Erfolg abgeschlossen, erfordern aber weitere Untersuchungen.

Bei realen Einsätzen auf dem Wasser erwies sich die genutzte Hardware als ungeeignet. Die Rechenkapazitäten reichen für eine Kollisionserkennung mehrerer Hindernisse nicht aus, und eine Echtzeitfähigkeit des System kann nicht realisiert werden. Die Betrachtung der Laufzeiteigenschaften hat außerdem gezeigt, dass eine Neuberechnung der Route bei einer Kollisionserkennung sehr zeitintensiv ist. Die Evaluation des Gesamtsystems (Siehe Kapitel 8.2) hat jedoch ergeben, dass die Berechnungen durch schnellere Hardware enorm beschleunigt werden können. Für die weitere Entwicklung wäre eine Portierung des Systems auf geeignete Hardware daher notwendig.

Für Anwendungen des Wasserfahrzeugs auf einem Kanal oder auf der See sind folgende Faktoren zu verbessern: Die Motoren haben sich in den Tests als zu schwach erwiesen. Besonders bei aufkommender Strömung oder ungünstigen Windverhältnissen machte sich dies bemerkbar. Darüber hinaus fehlt dem Boot die gesetzlich vorgeschriebene Lichterführung. Hierfür müssen Seitenlichter und ein Rundumlicht angebracht werden.

Um das Boot weitestgehend wasserdicht zu halten, ist eine zugeschnittene Plane geeignet. Der Zugang zur Hardware muss flexibel bleiben, weshalb eine Dachkonstruktion nicht infrage kommt. Bei einem Fachgeschäft für den Boots- und Schiffsbau haben wir erfahren, dass es für den Optimisten bereits vorgefertigte Planen gibt. Diese könnte dann entsprechend zugeschnitten werden und ließe sich einfach am Boot befestigen. Außerdem gibt es die Möglichkeit das Zuschneiden einer Plane bei einem entsprechendem Fachbetrieb zu beauftragen.

Anhang A

Seminarausarbeitungen

Die nachfolgenden Abschnitte beinhalten die schriftlichen Zusammenfassungen der Vorträge, die im Rahmen der Seminarphase während unserer wöchentlichen Gruppensitzungen gehalten wurden.

A.1 Software des bestehenden Systems

Von Dennis Pilny, vorgetragen am 20. Oktober 2015.

Hauptbestandteil der Software des aktuell eingesetzten Systems sind zwei Java-Applikationen. Zum einem die Onboard-Software, die auf dem Raspberry Pi läuft, welcher sich auf dem Boot befindet. Dessen Aufgabe ist es, Missionen entgegenzunehmen, zu planen und durchzuführen. Zum anderen die Onshore-Software. Diese bildet die Mensch-Maschinen-Schnittstelle und ermöglicht die Konfiguration des Bootes, das Planen und Starten einer Mission. Außerdem stellt sie den aktuellen Zustand des Bootes dar. Neben den beiden Java Applikationen werden zur Sensormessung und Ansteuerung der Motoren Arduino Boards verwendet, welche C++-Code ausführen. Auf letztere wird in diesem Beitrag nicht näher eingegangen.

Software-Architektur

Im Folgenden wird die Software Architektur von Onboard- und Onshore-Software erklärt, daraufhin folgt eine Beschreibung der Kommunikation zwischen den Applikationen mittels XBee. Abschließend wird eine kurze Bewertung über die Architektur gegeben und ein möglicher Ausblick zur Weiterentwicklung der Software gegeben.

Onboard-Software

Die Onboard-Software der Plattform wird auf einem Raspberry Pi ausgeführt. Folgend werden die einzelnen Teilaufgaben der Software beschrieben:

- Kommunikation mit der Onshore-Software
- Berechnung der Motorsteuerwerte
- Missionsverwaltung
- Ansteuerung der Motoren
- Auslesen von CPU-Temperatur
- Auslesen von Sensordaten
- Auslesen von GPS-Koordinaten
- Planen von Missionsrouten beim Start einer Mission
- Persitierung des aktuellen Zustands

Komponenten Um eine spezifische Teilaufgabe zu erfüllen, wurden diese in einzelne Komponenten gekapselt, welche indirekt miteinander kommunizieren. Der Aufbau einer Komponente ist dabei vorgeschrieben. Wird eine Komponente eingehängt so wird `init()` einmalig aufgerufen. In dieser Methode werden nötige Initialisierungen vorgenommen. Nach dieser wird in regelmäßigen Abständen die `update()`-Methode aufgerufen. In dieser führt die Komponente ihre Aufgabe. In der `destroy()`-Methode kann eine Komponente zum Ende seiner Laufzeit Ressourcen freigeben. Dieser Lebenszyklus wird in Abb. A.1 graphisch dargestellt.

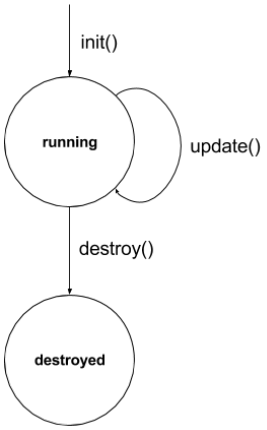


Abbildung A.1: Visualisierung des Lebenszyklus

Generell wird unterschieden zwischen IO-Komponenten und sequentiellen Komponenten. Letztere werden vom `ComponentManager` verwaltet und hinter-

einander auf dem Main-Thread ausgeführt. Dementsprechend dürfen sequentielle Komponenten keine lang andauernden Berechnungen in der `update()`-Methode ausführen, da dies sonst die Ausführung der restlichen Komponenten behindern würde. Die Reihenfolge des Ausführen der einzelnen Komponenten entspricht dabei der Reihenfolge des Hinzufügens zum `ComponentManager`. Abbildung A.2 skizziert den sich wiederholenden Aufruf der `update()`-Methode einer sequentiellen Komponente im linken Teil der Abbildung.

Langwierige Aufgaben werden als IO-Komponente implementiert, diese führen ihre Aufgaben in separaten Threads durch. Die Verwaltung dieser erfolgt im `IOManager`. Eine IO-Komponente ruft seine `update()`-Methode in einem selbst definierten Abstand – das Minimum beträgt hierbei mindestens zehn Millisekunden.

Die indirekte Kommunikation zwischen den Komponenten erfolgt über die Klasse `Data`. Diese enthält Felder mit Daten welche synchronisiert gelesen und geschrieben werden können. So kann beispielsweise die GPS-IO-Komponente die aktuellen GPS-Koordinaten speichern und die sequentielle Komponente `PathPlaner` kann diese dann lesen.

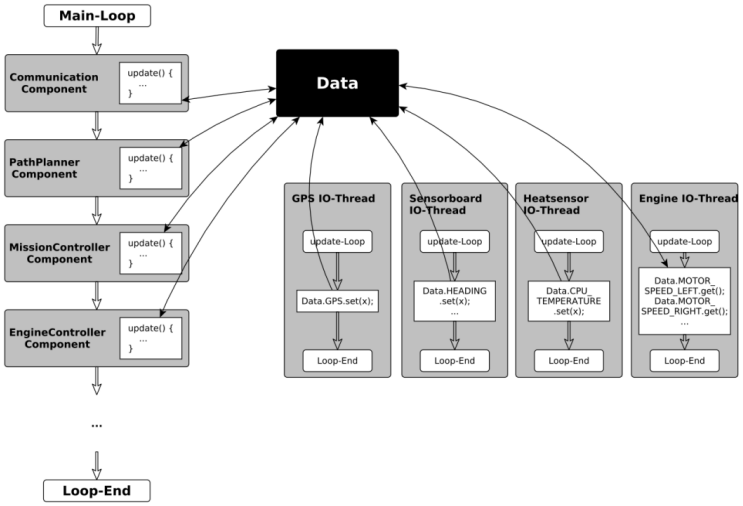


Abbildung A.2: Architektur der Onboard-Software (aus [31, S. 23])

Planung und Steuerung Zur Veranschaulichung der Funktionsweise der Komponenten wird nun ein detaillierter Blick auf die Missionsplanung und Durchführung auf der Plattform gerichtet.

Abbildung A.3 stellt die verwendeten Komponenten für die Durchführung und Planung einer Mission graphisch dar. Die gestrichelten Linien visualisie-

ren eine indirekte Kommunikation zwischen den Komponenten und die durchgezogenen den Zugriff auf Felder des Data Objekts.

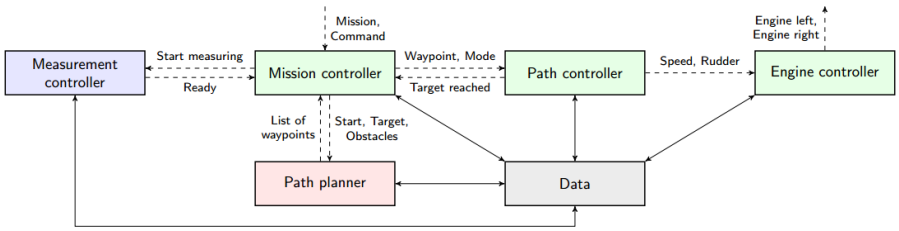


Abbildung A.3: Kommunikation für die Missionsplanung (aus [31, S. 51])

Die Durchführung einer Mission wird koordiniert von der `Mission Controller` Komponente. Sie implementiert die Funktionalität des Aktivitätsdiagramms aus Abb. A.4. Wird eine Mission an die Onboard-Software übertragen und gestartet, so startet der `Mission Controller` den Plan Modus. Da das Planen der Mission eine längere Zeit beansprucht ist diese in der IO-Komponente `Path Planer` ausgelagert. Diese nimmt die aktuelle Position als Startkoordinate und den nächsten Messpunkt aus dem Missionsplan und berechnet eine Liste an Wegpunkt zu dem Messpunkt hin. Sobald die Berechnung abgeschlossen ist, weist der`Mission Controller` dem `Path Controller` einen `Waypoint` zu der angefahren werden soll. Dessen Aufgabe ist den Kurs des Bootes auf der berechneten Bahn zum Messpunkt zu halten. Dies wird realisiert über eine PID-Regelung. Aktuell gibt es fünf verschiedene Implementierung dieser, die in Abhängigkeit mehrerer Parameter (aktueller Ort, Ausrichtung des Bootes, Distanz zum Messpunkt, ...) abstrakte Ruder und Geschwindigkeitswerte berechnen. Die abstrakten Ruder und Geschwindigkeitswerte werden von der `Engine Controller` Komponente zu konkreten Werten zur Ansteuerung der Motoren umgerechnet. Die Abstraktion der Motorsteuerwerte ermöglicht eine einfache Implementierung anderer Motor und Steuerungskonzepte.

Onshore-Software

Die Onshore-Software dient zur Missionsplanung und Konfiguration des Bootes. Zusätzlich bietet sie einen Überblick über den aktuellen Status des Bootes. Parameter die konfiguriert werden können sind die Art des Path-Controllings, die Parameter für die PID-Regelung und die Kalibrierung des Kompasses. Abbildung A.5 zeigt die GUI der Software.

Der Verbindungsaufbau zu dem Boot erfolgt automatisch im Hintergrund beim Start der Software, wobei die Verbindungsparameter hartkodiert im Sourcecode stehen.

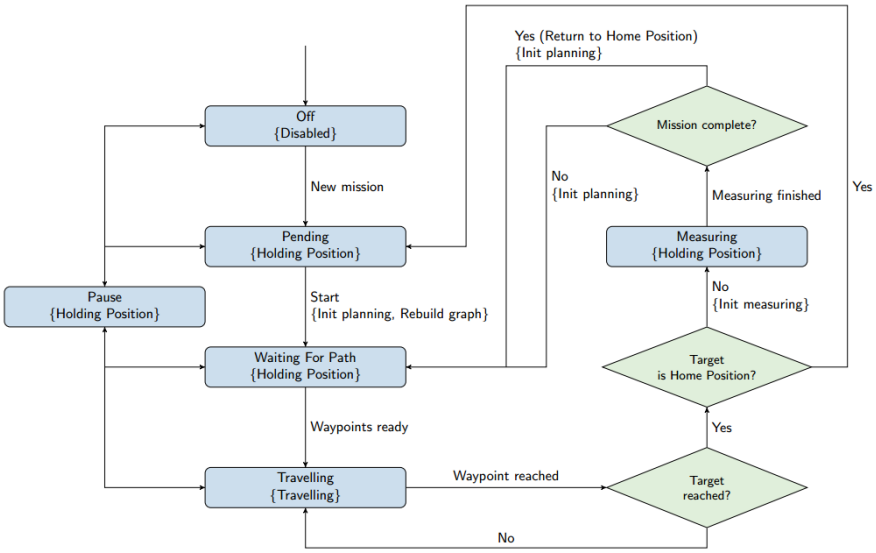


Abbildung A.4: Durchführung einer Mission (aus [31, S. 66])

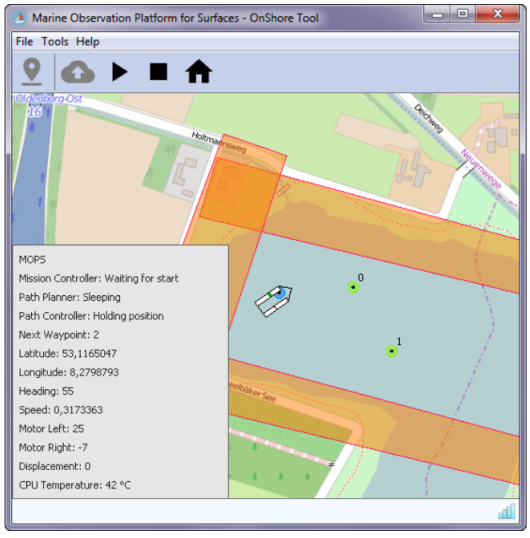


Abbildung A.5: Benutzeroberfläche der Onshore-Software (aus [31, S. 16])

Hauptbestandteil der Oberfläche bildet eine Karte, die das Boot darstellt. Das Kartenmaterial wird über die OpenSeaMap API angefordert und die Darstellung erfolgt über den `JXMapViewer`, welche sich problemlos in eine Swing GUI einbinden lässt. Innerhalb der Karte kann mit einem Linksklick

+ STRG-Taste Messpunkte für eine Mission gesetzt werden. Sperrzonen, die vom Boot umfahren werden sollen, können erstellt werden, indem man die Shift-Taste hält und mit Linksklick Eckpunkte setzt. Sobald man die Shift-Taste loslässt wird die Sperrzone dargestellt. Klickt man das Boot an, so werden in einem Kasten in der unteren linken Ecke Allgemeine Informationen zu dem Boot angezeigt.

Kommunikation

Die Kommunikation zwischen Onshore- und Onboard-Software erfolgt per Funk über XBee Module. Zur Übertragung von Daten wurde ein eigenes Protokoll implementiert. Eine Nachricht wird dabei byteweise aufgeteilt, im Folgenden wird das Format einer Nachricht erläutert.

- 1. Byte: Länge der Nachricht
- 2. Byte: Kommando Id
- 3. ... N-2. Byte: Inhalt der Nachricht
- N-1. und N. Byte: CRC16 Prüfsumme

Der Kommunikationsablauf ist dem Request-Response Modell nachempfunden. Es sendet also immer nur eine Seite Anfragen an die andere und erhält dann von dieser Antworten. Das Stellen der Anfragen wird von der Onshore-Software ausgeführt. Um die Funkverbindung stetig zu Überprüfen, sendet die Onshore-Software eine HeartBeat Anfrage. Die Onboard-Antwort auf eine `HeartBeat`-Anfrage enthält allgemeine Informationen zu dem Boot.

Wenn größere Datenpakete übertragen sollen, so besteht die Möglichkeit die Daten auf mehrere Nachrichten aufzuteilen. Der Start einer Nachricht wird über eine `DataTransferBegin` signalisiert. Darauf folgen `DataTransferPacket` Nachrichten, welche den Teilinhalt des Gesamtpakets enthalten. Über eine`DataTransferFailed` Nachricht wird das fehlschlagen der Übertragung signalisiert. Eine `DataTransferEnd` Nachricht signalisiert das Ende der Gesamtnachricht.

Empfangen Nachrichten werden auf die CRC16 Prüfsumme überprüft. Je nach Wichtigkeit des Kommandos wird ein Ereignis geworfen, welches das erneute Senden der Nachricht ermöglicht. Die folgende Liste enthält die aktuelle implementierten Kommandos, sowie eine kurze Beschreibung.

StatusOK Inhaltslose Antwort, die das erfolgreiche Empfangen einer Nachricht signalisiert

StatusErrorAntwort die Fehler signalisiert (Anmerkung: im Code nicht verwendet)

Parameter Enthält spezifische Parameterinformationen zur Kalibrierung oder Konfiguration

DataTransferBegin Start Nachricht eines größeren Datenpakets (z.B. Übertragung einer Mission)

DataTransferPacket Datenpaket mit Teilinhalt eines größeren Datenpakets

DataTransferEnd End Nachricht, die das Ende eines größeren Datenpakets signalisiert

DataTransferFailed Fehler Nachricht, die die Fehlerhafte Übertragung eines größeren Datenpakets signalisiert

HeartBeatRequest HeartBeat Anfrage an die Onboard-Software

HeartBeatResponse HeartBeat Antwort an die Onshore-Software, enthält Informationen zum aktuellen Zustand des Bootes

Pause Nachricht zum Pausieren der aktuellen Mission

Start Nachricht zum Starten der aktuellen Mission

ReturnHome Nachricht zum Abbruch der Mission und Zurückkehren zum Home Punkt

Stop Nachricht zum Abbruch der aktuellen Mission

ParameterInfoRequest Anfrage an die Onboard-Software für die aktuell verwendeten Parameter

ParameterInfoResponse Antwort an die Onshore-Software mit den aktuelle verwendeten Parametern

Entwicklung und Test

Dieser Abschnitt beschreibt den aktuellen Zustand des Sourcecodes, wie an ihm entwickelt wurde und auf welche Weise man diesen Testen kann.

Projektstruktur

Onboard sowie Onshore-Software sind wie bereits erwähnt Java Applikation. Die Strukturierung der Projekte erfolgt in Eclipse Projekten. Dies bietet die Möglichkeit Sourcecode, der in beiden Applikationen verwendet wird aufzuteilen. Insgesamt teilt sich die Java-Software in sechs Teilprojekte auf.

- Communication: Enthält Klassen zur Kommunikation mittels XBee zwischen Onboard und Onshore Applikation

- Model: Enthält Klassen zur Beschreibung von Missionen und dem Zustand des Bootes
- Onboard: Enthält die Onboard Applikation, welche auf dem Raspberry Pi ausgeführt wird
- Onshore: Enthält die Onshore Applikation, welche auf einem Rechner ausgeführt wird
- PathPlaning: Enthält den Algorithmus zur Planung einer Route zu einem Messpunkt
- Util: Enthält Allgemeine Hilfsklassen

Alle Projekte werden mit Java 8 gebaut. Zum Kompilieren der Onboard und Onshore Applikationen weden ANT-Skripte eingesetzt, die ausführbare Jars erstellen. Da es sich um Eclipse Projekte handel, können diese ohne größere Komplikationen nur mit der Eclipse IDE weiterentwickelt werden.

Testen

Um die Funktionalität der Software zu testen ohne an einen See zu fahren gibt es die Möglichkeit Komponenten zu simulieren. Über boolsche Flags kann im SourceCode der Simulationsmodus aktiviert werden. Ist dieser aktiv, so werden GPS-Daten sowie die Daten des Sensorboards simuliert. Im Simulationsmodus kann sowohl die Onshore- als auch die Onboard-Software auf dem Entwicklungsrechner ausgeführt werden. Dabei kommunizieren die beiden Applikationen über UDP statt über das XBee Modul.

Beurteilung und Ausblick

Der Sourcecode ist übersichtlich aufgeteilt, es sind jedoch nur wenige Klassen ausreichend Dokumentiert. Außerdem gibt es für die Teilprojekte keine Versionierung. Es würde sich anbieten das Build System auf Maven oder Gradle umzustellen. Dies würde es ermöglichen versionierte Artefakte der Teilprojekte zu bauen. Zusätzlich wäre das Entwickeln nicht mehr zwingend an die Entwicklung mit der Eclipse IDE gebunden.

Der Betrieb der Onshore-Software funktioniert einwandfrei im Simulationsmodus zusammen mit der Onboard-Software. Jedoch ist die Bedienung nicht sehr intuitiv, und im Hintergrund werden Exceptions geworfen, die jedoch keinen negativen Einfluss auf den Betrieb der Software hat.

A.2 Hardware des bestehenden Systems

Von Michael Beering, vorgetragen am 20. Oktober 2015.

Die Projektgruppen Mops I-III haben ein autonomes Wasserfahrzeug entwickelt, welches in der Lage ist verschiedene Wasserparameter zu bestimmen. Das folgende Kapitel beschreibt die Hardwarekomponenten des Wasserfahrzeuges. Dieses wurde in der Zeit vom Wintersemester 2014 bis Sommersemester 2015 im Rahmen der Projektgruppe Mops III entwickelt. Zwei weitere Projektgruppen beschäftigten sich bereits in den vorangegangen zwei Jahren mit entsprechenden früheren Versionen der Plattform.

Überblick

Grundlage der Plattform ist das in Abb. A.6 dargestellte Boot vom Typ Optimist. Die Bootsklasse Optimist beschreibt einen Bootstyp, der üblicherweise im Bereich des Segelns für Jugendliche bis 15 Jahren eingesetzt wird. Im Rahmen der Projektgruppe wurde das Boot zu einem autonomen Wasserfahrzeug umgebaut. Die dafür nötigen Bauteile werden im Folgenden in einem kurzen Überblick benannt.

Abbildung A.6: Übersicht der Plattform von Projektgruppe Mops III (aus [31, S. 77])

Alurahmen

Eine wichtige Voraussetzung für die Montage von Bauteilen ist ein solider Grundrahmen innerhalb des Bootes. Dieser wurde durch vorherige Projektgruppen realisiert. Eingesetzt wurden hierfür Boschprofile, in die Otterboxen eingesetzt werden können.

Der Rahmen unterteilt sich in zwei Bereiche. Der hintere Bereich (siehe Abb. A.6 Markierung 1) beinhaltet die Otterboxen für die Rechner und die Elektronik. Außerdem werden hier die Auftriebskörper angebracht. Der vordere Teil (siehe Abb. A.6 Markierung 2) bietet Platz für zwei weitere Otterboxen, in denen jeweils eine PKW-Batterie untergebracht ist.

Außerdem ist an dem Grundrahmen der Mast für die Datenübertragung per XBee montiert worden.

Otterboxen

Um eine Hochseetauglichkeit zu gewährleisten ist es essentiell, dass sämtliche Elektronikkomponenten gegen Wasser geschützt werden. Hierfür werden Otterboxen eingesetzt. Diese sind mit Kabeldurchführungen ausgestattet, um zum Beispiel Strom von den Batterien zu erhalten oder die Motoren anzusteuern. Neben dem Schutz vor Wasser bieten die Otterboxen auch die Möglichkeit die Elektronik fest zu verbauen, sodass diese gegen Vibration und Stöße geschützt wird.

Auftriebskörper

Um ein Sinken des Bootes bei einem möglichen Wassereintritt zu verhindern, sind an der Rahmenkonstruktion Auftriebskörper befestigt. Jeder Auftriebskörper umfasst ein Volumen von 55 l, sodass genug Auftrieb erzeugt werden kann, um die Plattform bei Wassereintritt an der Oberfläche zu halten.

Sensorkäfig

Da das Wasserfahrzeug unter anderem das Ziel verfolgt Wassermesswerte zu bestimmen, existiert ein Sensorkäfig, in dem vier Sensoren untergebracht werden können. Gefertigt ist dieser aus Boschprofilen, die eine leichte Montage des Käfigs ermöglicht haben. Eine Halterung, um den Aufbau kontrolliert ins Wasser ablassen zu können, existiert bisher noch nicht. Abbildung A.7 zeigt den Sensorkäfig.

Energieversorgung

Die Energieversorgung an Bord wird durch zwei 12 V PKW-Batterien realisiert. Diese sind wasserdicht in zwei Otterboxen verbaut und werden über zwei Krokodil-Klemmen mit der Elektronik verbunden. Neben der Stromversorgung des Raspberry Pi, des Arudino und diverser Sensoren versorgen

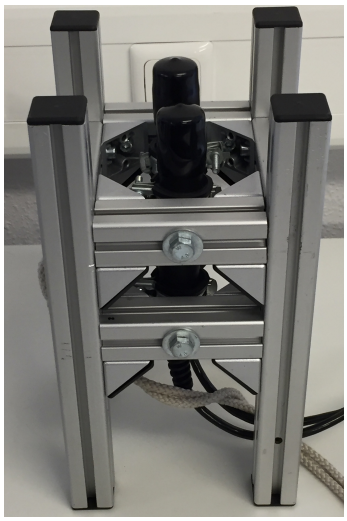


Abbildung A.7: Prototypischer Sensorkäfig

die Batterien auch die Außenbordmotoren mit Strom. Zur Spannungsregelung sind Spannungsregler verbaut worden, um z.B. Ausgangsspannungen von 5 V zu erzeugen. Abbildung A.8 zeigt eine eingebaute Batterie. Um die Verpolungsgefahr zu reduzieren, wurden für den Plus- und Minuspol deutliche Markierungen angebracht.

Abbildung A.8: In eine Otterbox eingebaute Batterie

Raspberry Pi und Arduino

Der Raspberry Pi ist ein kleiner und preiswerter Einplatinen-Computer, der die zentrale Recheneinheit der Plattform darstellt. Auf ihm läuft die Onboard-Software, die zum Ausführen von Szenarien und der Pfadplanung eingesetzt wird. Darüber hinaus steuert der Raspberry Pi einige Geräte an und liest Sensoren aus. Die Kommunikation zum Festland per XBee wird ebenfalls durch ein mit dem Rechner verbundenes Modul von dem Raspberry Pi durchgeführt.

Eine zentrale Aufgabe des Arduino besteht darin, die Steuerung der Außenbordmotoren zu übernehmen. Näheres hierzu findet sich in Abschnitt A.2. Des Weiteren wird der Arduino zum Auslesen des Salzgehalts- und Trübheitssensors verwendet.

Sowohl Raspberry Pi als auch Arduino sind in den Otterboxen im hinteren Teil des Optimists untergebracht (siehe Abb. A.9) und werden von hier mit Strom der PKW-Batterien versorgt.

(a) Raspberry Pi in Otterbox (b) Arduino in Otterbox

Abbildung A.9: Eingebauter Raspberry Pi und Arduino (aus [31, S. 81 f.])

Motorsteuerung

Die wichtigsten Elemente um das Wasserfahrzeug zu navigieren sind die zwei Außenbordmotoren. Hierbei handelt es sich um zwei Rhino-VX-34 Motoren.

Diese sind am Heck des Bootes befestigt und von dort mit spritzwassergeschützten Steckern mit den Batterien verbunden. Die folgende Abb. zeigt einen der beiden Motoren.

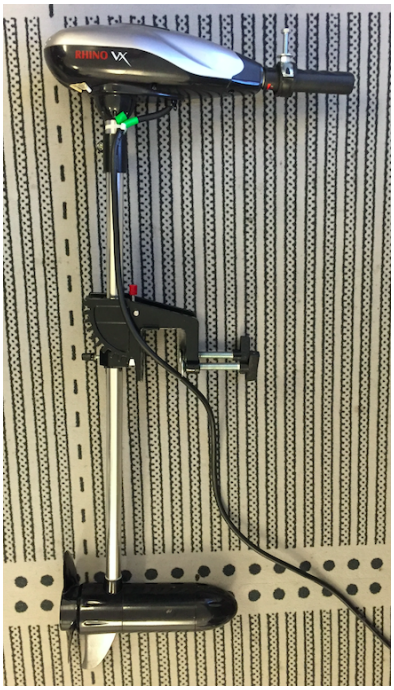


Abbildung A.10: Motor Rhino-VX-34

Steuerung per Software

Die Steuerung der Motoren per Software funktioniert nach dem in Abb. A.11 dargestellten Kommandopfad. Zunächst werden mit der Steuerungssoftware auf dem Raspberry Pi Motoreinstellungen für einen bestimmten Kurs berechnet. Diese werden an den Arduino übertragen. Der Arduino generiert pulsweitenmodulierte Signale, welche der Motor erwartet und in entsprechende Rotationen der Rotoren übersetzt. Da die vom Arduino generierten Signale nicht stark genug sind, werden diese auf dem Weg zum Motor von einer Leistungselektronik verstärkt.

Steuerung per Fernbedienung

Aus Sicherheitsgründen lässt sich das Boot auch per RC-Fernbedienung steuern. Dies hat den Vorteil, dass in Situationen wie dem Ablegen vom Ufer per Fernbedienung eingegriffen werden kann. Darüber hinaus haben die Steuersignale der Fernbedienung immer Vorrang gegenüber den Signalen, die von

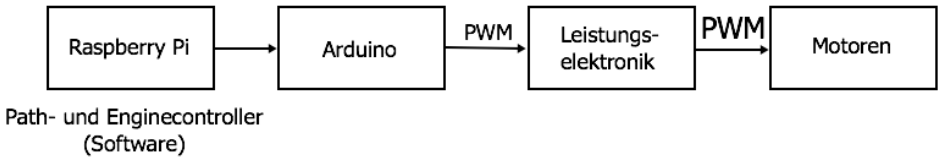


Abbildung A.11: Kommando-Pfad der Motorsteuerung per Software

der Software berechnet werden. Dies ermöglicht ein direktes Eingreifen per Fernbedienung, sodass der Mops in kürzester Zeit in einen sicheren Bereich navigiert werden kann.

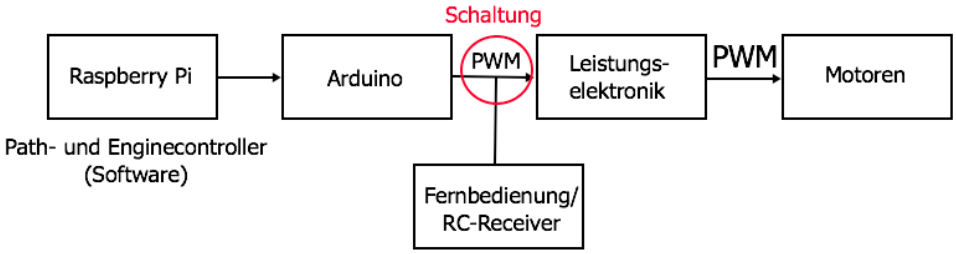


Abbildung A.12: Kommando-Pfad der Motorsteuerung per Fernbedienung

Die technische Realisierung und der entsprechende Kommandofluss sind in Abb. A.12 dargestellt. Zwischen der Übertragung der PWM-Signale vom Arduino zur Leistungselektronik wurde eine Schaltung ergänzt. Diese gibt den Signalen der RC-Fernbedienung immer Vorrang. Sendet diese keine Signale, so werden die Signale der Softwaresteuerung an den Motor weitergegeben. Die Schaltung realisiert außerdem, dass die Fernbedienung bei Systemstart einmal benutzt werden muss, bevor die Softwaresteuerung übernehmen darf. Dies hat den Grund, dass häufig Probleme beim Ablegen vom Ufer aufgetreten sind, sodass an dieser Stelle ein manuelles Ablegen notwendig ist.

Sensoren

Dieses Kapitel gibt einen kurzen Überblick über die verbauten Sensoren. Im ersten Teil wird die zur Navigation nötige Sensorik beschrieben. Der zweite Teil geht auf die Sensoren zur Bestimmung von Wasserparametern ein.

GPS

Die Plattform ist mit verschiedenen Sensoren ausgestattet. Für die Missionsplanung und die Navigation ist die Position des Wasserfahrzeugs im Meer (oder auf einem See) von besonderem Interesse. Als GPS-Empfänger wird ein GM-65 eingesetzt. Dieser ist per Magnet am Boot befestigt. Es wurde darauf geachtet, dass der Empfänger nie von anderen Bauteilen überdeckt wird, damit es dadurch nicht zum Signalverlust kommt. Verbunden wird der GM-65 mit dem Raspberry Pi, sodass dieser die entsprechenden Daten auslesen kann. Für die Datenübertragung wird das NMEA-0813-Protokoll verwendet. Vorteil dieses Protokolls ist es, dass die Datensätze in einem standardisierten Format vorliegen und von diversen Navigationsprogrammen gelesen werden können.

IMU-Board

Das IMU-Board ist mit drei verschiedenen Sensoren ausgestattet, mit denen es möglich ist, die Ausrichtung der Plattform zu bestimmen. Der erste Sensor ist ein 3-Achsen-Magnetfeld-Sensor, aus dessen Messwerten die Ausrichtung berechnet werden kann. Bei dem zweiten Sensor handelt es sich um einen Beschleunigungssensor. Der dritte Sensor ist ein Gyroskop, welches das Drehverhalten bestimmt. Aus diesem kann ein Rückschluss auf den Wellengang gezogen werden. Dies hat den Hintergrund, dass das Gefährt bei zu starkem Wellengang abschalten oder Sicherheitsmaßnahmen einleiten soll.

C4E Conductivity

Der C4E Conductivity Sensor misst den Salzgehalt und die Leitfähigkeit des Wassers. Ausgewertet werden die Messwerte über einen Arduino. Mit entsprechenden Bibliotheken werden die Daten an den Raspberry Pi weitergegeben. Näheres zu diesem Sensor wird in der Seminarausarbeitung Umweltsensoren (vgl. Abschnitt A.10) beschrieben.

Nephelometric Turbidity

Dieser Sensor misst die Trübheit des Wassers. Das Auslesen der Messwerte erfolgt analog zu dem vorangegangenen Sensor. Auch hier gibt es weiterführende Informationen in der Seminarausarbeitung Umweltsensoren (vgl. Abschnitt A.10).

Diverses

Der folgende Abschnitt beschreibt den Kommunikationsmast, die Debug LED zur Hinweisausgabe und den Arduino Watchdog, welcher zum Starten des Raspberry Pi nach einem Systemaufall eingesetzt wird.

Funkmast

Ein Bild des Funkmastes des Wasserfahrzeugs ist in Abb. A.13 dargestellt. Der Mast besteht aus einem einfachen PVC-Rohr, in welches die XBee-Antenne integriert wurde. Hiermit ist die Datenübertragung mit einem am Raspberry Pi verbauten XBee-Modul möglich.

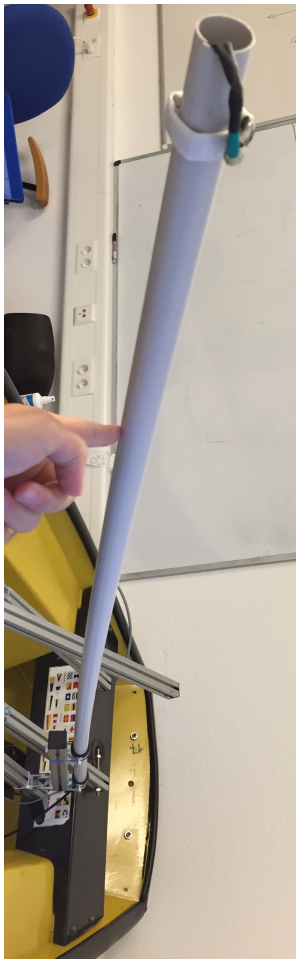


Abbildung A.13: Funkmast mit Debug-LED

Debug-LED

Der Funkmast enthält neben der Antenne auch eine Debug-LED. Diese dient dazu den am Ufer bedindlichen Personen eine Information über den Zustand der Plattform zu geben. Beispielsweise bedeutet die Farbe grün: Arduino empfängt korrekte Motorwerte oder gelb: Raspberry Pi hat XBee-Signal verloren.

Arduino-Watchdog

Der Arduino-Watchdog erfüllt die Aufgabe den Raspberry Pi neu zu starten, wenn es zu einem Systemabsturz gekommen ist. Sendet der Raspberry Pi keine Signale mehr an den Arduino (kein Heartbeat zu hören), so führt dieser einen Neustart des Systems durch. Eine entsprechende Anzeige an der Debug-LED wird außerdem angezeigt.

A.3 Rechtliche Rahmenbedingungen

Von Henning Lawatsch, vorgetragen am 20. Oktober 2015.

Diese Seminararbeit befasst sich mit den rechtlichen Merkmalen, die bei dem Bau eines autonomen Forschungsbootes MOPS IV beachtet werden müssen. Sobald öffentliche Gewässer genutzt werden, gibt es Richtlinien und Gesetze, die eingehalten werden müssen, aber abhängig vom Standort sind. Als grundlegende Regelung für den Internationalen Schiffsverkehr dienen die Kollisionsverhütungsregeln.

Für das MOPS-Projekt ist langfristig ein Hochseeeinsatz geplant. Im Idealfall soll das Forschungsboot in der Nordsee bis nach Helgoland einsetzbar sein. Auf dieser Grundlage ist eine Betrachtung aller rechtlichen Rahmenbedingungen, die für dieses Vorhaben relevant sind, sinnvoll. Dies sind zum einen die international geltenden Kollisionsverhütungsregeln und zum anderen nationale Regelungen, wie die Seeschifffahrtsstraßen-Ordnung.

Kollisionsverhütungsregeln

Die Kollisionsverhütungsregeln nach dem Bundesministerium für Justiz und Verbraucherschutz [41] sind die grundlegenden Regelungen für alle Wasserfahrzeuge auf befahrbaren Gewässern. Die offizielle Bezeichnung lautet: „internationale Regeln von 1972 zur Verhütung von Zusammenstößen auf See“. Der Bezug gilt insbesondere für Schiffe auf Hoher See und den damit verbundenen Gewässern. Die KVR werden als übergeordnete Regeln angesehen, die gelten, solange keine andere Regelung (Sondervorschrift) greift oder sie verändert.

Ausweich- und Fahrregeln

Zunächst ist festgelegt, wie Wasserfahrzeuge sich im Schiffsverkehr zu verhalten haben und wie ein Ausweichmanöver durchgeführt werden muss. Grundvoraussetzung ist, dass ein Verkehrsteilnehmer über eine gute Sicht auf den Bereich um sein Schiff verfügt. Da das Boot ohne eine Besatzung und autonom fährt, muss dieser Punkt entweder ignoriert (regelwidrig) oder über eine Kamera (mit ausreichendem Zoom) realisiert werden. Zudem muss ein

Fahrzeug mit einer sicheren, den Fahrverhältnissen entsprechenden Geschwindigkeit fahren, damit es manövrierfähig ist. Des Weiteren muss eine Radaranlage vorhanden sein, um Gefahren von Zusammenstößen zu detektieren und die Fahrtrichtung anderer Fahrzeuge zu identifizieren. Diese müsste an der Plattform noch angerbracht werden.

Für ein Ausweichmanöver gibt es spezielle Verhaltensregeln: Eine Änderung des Kurses oder der Geschwindigkeit sollte nicht in einzelnen kleinen Schritten erfolgen, sondern in einem großen deutlichen Schritt, damit andere Verkehrsteilnehmer das Manöver klar erkennen und deuten kann. Wird ein Ausweichmanöver durchgeführt, ist es eventuell erforderlich die Geschwindigkeit zu senken oder gar zu stoppen. Zusätzlich muss ein ausreichender Abstand beim Passieren eingehalten werden. Diese Verhaltensweisen müssten in die System-Software implementiert werden. Alternativ könnte das Boot einfach stoppen sobald ein möglicher Zusammenstoß vorauszusehen ist und somit ein Ausweichmanöver notwendig wäre. Das Ausweichen wäre dadurch dem anderen Fahrzeug überlassen und die Plattform könnte im Anschluss mit seiner Mission fortfahren.

Weiterhin müssen spezielle Gebiete beachtet werden, in denen besondere Fahrregeln gelten. Dies sind beispielsweise enge Fahrwasser, in denen größeren Schiffen die Durchfahrt gewährt werden sollte und das Fahren möglichst weit am Rand verlangt wird. Verkehrstrennungsgebiete teilen das Gewässer mit Hilfe von Seezeichen in zwei Spuren (für jede Richtung eine) auf. Sie sind in den Seekarten abgebildet.

Wenn das autonome Wasserfahrzeug eines dieser Gewässer befahren soll, muss es erkannt und seine Regeln befolgt werden. Eventuell können Informationen aus aktuellen Seekarten einbezogen werden.

Sichtkontakt zwischen Fahrzeugen

Ist ein anderes Fahrzeug in Sicht, müssen gewisse Verhaltensregeln eingehalten werden. Ein überholendes Fahrzeug muss dem anderen Fahrzeug ausweichen. Sind die Kurse von zwei Schiffen entgegengesetzt, müssen beide in Richtung Steuerbord (nach rechts) ausweichen. Bei kreuzenden Kursen muss derjenige ausweichen, der das andere Fahrzeug an seiner Steuerbordseite hat (ähnlich wie rechts vor links im Straßenverkehr).

Schiffsausrüstung

Lichter Ein Fahrzeug im Schiffsverkehr muss über diverse Ausrüstung verfügen, um sich gegenüber anderen Verkehrsteilnehmern bemerkbar zu machen. Dies sind als Erstes verschiedene Lichter, die am Fahrzeug angebracht sein müssen. Ohne sie ist es nicht möglich bei Nacht oder verminderter Sicht ein Schiff zu erkennen. Auch am Tag sind Signalkörper notwendig. Da das Boot zurzeit noch weniger als 7 Meter lang ist,

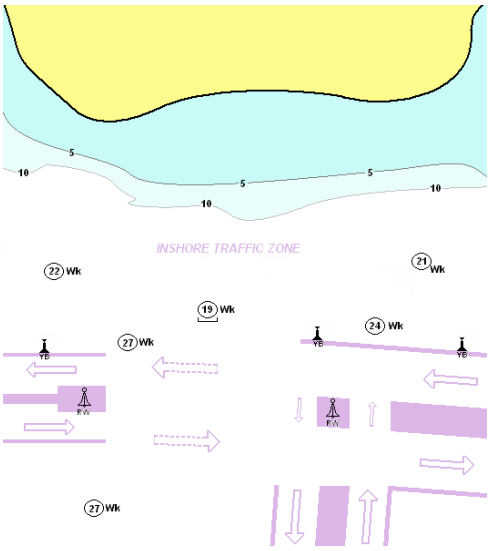


Abbildung A.14: Verkehrstrennungsgebiet [130]

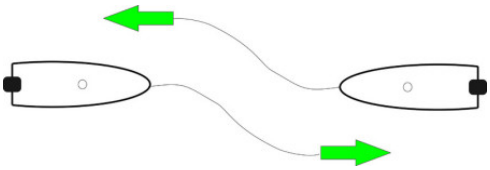


Abbildung A.15: Ausweichen bei entgegengesetztem Kurs [49]

kann stattdessen ein weißes Rundumlicht und wenn möglich zusätzliche Seitenlichter mitgeführt werden. Die Geschwindigkeit sollte jedoch nicht 7 Knoten überschreiten.

Schallsignale Als Zweites müssen Schiffe ab einer Länge von 12 Metern eine „Pfeife“ mitführen, um entsprechende Manöver- und Warnhinweise zu geben. Das Boot wäre vom Mitführen dieser zwar befreit, da es die dafür vorausgesetzte Länge nicht erreicht, jedoch müssen die Signale von anderen Schiffen von ihm erkannt werden. Dies könnte durch ein Mikrofon inklusive einer Softwareimplementierung, welche die Töne erkennt und deutet, realisiert werden. Bei Gebieten mit verminderter Sicht müssen Wasserfahrzeuge jedoch alle 2 Minuten ein kräftiges Schallsignal abgeben. Auch dies muss für das Forschungsboot ermöglicht werden.

Nationale Gewässer

In vielen Orten gelten Sondervorschriften, welche die KVR ergänzen und dessen Regelungen auf die speziellen örtlichen Verhältnisse anpasst. Diese sollten allerdings weitestgehend mit den Regeln der KVR übereinstimmen. Für unser Projekt sind vor allem die Seeschifffahrtsstraßen-Ordnung (SeeSchStrO) und die Schifffahrtsordnung Emsmündung (EmsSchO) von Bedeutung. Die Seeschifffahrtsstraßen-Ordnung nach dem dem Bundesministerium für Justiz und Verbraucherschutz [42] gilt für befahrene Gewässer innerhalb Deutschlands, mit Ausnahme der Emsmündung, für die eine weitere Sondervorschrift gilt. In ihr werden unter Anderem weitere Fahrzeugklassen definiert und festgelegt, dass in Fahrwassern fahrende Schiffe Vorfahrt haben [129].

(Hoch-)Seetauglichkeit der autonomen Plattform

An dieser Stelle soll geprüft werden, ob für den Das Boot ein Einsatz im Schiffsverkehr infrage käme und welche Voraussetzungen dafür notwendig wären.

Da die Plattform autonom fährt, gilt es als unbemanntes Schiff und somit ist es laut aktuellem Recht nicht möglich, ihn auf befahrenen Gewässern zu nutzen. Es müsste also zunächst eine Änderung der Rechtslage erfolgen. Um auf Seeschifffahrtsstraßen fahren zu dürfen benötigen Fahrzeuge eine Zulassung, welche eine ausreichende Ausrüstung und Fahrtauglichkeit bestätigt. Ebenfalls muss ein amtlich anerkanntes Kennzeichen vorhanden sein [117]. Die Ausrüstung des Bootes muss demnach erweitert werden, zum Beispiel mit einem Rundumlicht.

Um regelkonform am Schiffsverkehr teilzunehmen, müssen die Regelungen der KVR, sowie nationale Verordnungen eingehalten werden. Eine entsprechende Implementation dieser ist vermutlich zu Aufwendig. Geschieht diese allerdings nicht, könnte das Boot für Kollisionsschäden oder Schäden aufgrund von Ausweichmanövern anderer Schiffe verantwortlich sein.

Auch der Einsatz auf öffentlichen, nicht befahrenen Gewässern gestaltet sich als schwierig. Jede Region/Bundesland (oder Gemeinde) hat für seine Gewässer eigene Reglungen, die einem eigens gebauten Schiff das Befahren der Gewässer nicht ermöglichen. Ein Antrag auf Genehmigung wäre also notwendig.

Denkbar ist, die Plattform nach wie vor auf dem Tweelbäker See fahren zu lassen, wobei die Kollision mit Schwimmern und Schäden für die Umwelt des Sees unbedingt vermieden werden müssen.

A.4 Maritime Umgebung

Von Konstantin Gebel, vorgetragen am 27. Oktober 2015.

Um bei der Navigation auf See Unfälle zu vermeiden, müssen äußere Umwelteinflüsse, sowie Verkehrsregeln berücksichtigt werden. Das Wetter hat starken Einfluss auf die maritime Umgebung. Ein starker Wind sorgt beispielsweise für einen hohen Wellengang, was zum Kentern des Bootes führen könnte. Da die Plattform autonom arbeiten soll, ist es wichtig, dass diese ihre Umwelt erfassen und darauf reagieren kann. Dazu muss zunächst die Frage geklärt werden, auf welche Weisen das Boot durch die Meeresumgebung beeinflusst werden könnte. Aber nicht nur natürliche Einflüsse müssen beachtet werden. Auch andere Verkehrsteilnehmer und Seezeichen verlangen bei Konfrontation eine bestimmte Reaktion. Diese Seminararbeit geht auf die wichtigsten Aspekte der maritimen Umgebung ein und reflektiert inwiefern diese berücksichtigt werden müssen.

Umwelteinflüsse

Im Vergleich zum Verkehr auf dem Festland, herrschen auf See andere Bedingungen. Dazu gehört neben der natürlichen Umgebung auch der Einfluss, den der Mensch auf das Gewässer hat. In diesem Kapitel wird beschrieben, welche Auswirkungen Wind und Wellen auf das Boot haben. Weiterhin wird darauf eingegangen, inwiefern die Umwelt geschont werden kann.

Wind und Wellen

Die Stärke des Windes hat direkten Einfluss auf die Ausprägung der Wellen. Diese wiederum erschweren, je nach Wellengang, die Navigation auf dem Wasser. Ein erhöhter Wellengang kann ein Schiff beschädigen oder zum Kentern bringen. Die Geschwindigkeit des Windes wird meist in km/h oder in Knoten angegeben [105]. Für die Schifffahrt wird die letztgenannte Einheit häufig genutzt. Zu diesen Einheiten kann die Windstärke in Beaufort angegeben werden. Diese wird anhand einer Skala, die von null bis zwölf (Windstille bis Orkan) reicht, abgeleitet. Abbildung A.16 zeigt die Auswirkung bestimmter Windstärken auf die Höhe der Wellen. Bei einer Windstärke von drei würde das Boot noch auf dem Meer navigiern können, wobei Kehrtmanöver schon riskant wären. Ab einer Windstärke von sechs ist zu sehen, dass die Wellen brechen. Die Observationsplattform hätte hier schon starke Schwierigkeiten den Kurs zu halten und auch das Kentern ist wahrscheinlich.

Für das Boot würde dies bedeuten, dass der Wind auf irgendeine Weise erfasst werden muss. Dazu können verschiedene Ansätze verfolgt werden. Über ein Anemometer kann die Windgeschwindigkeit direkt auf der Plattform erfasst werden. Diese Sensoren messen die Windgeschwindigkeit über den Luftdruck oder mithilfe von Ultraschall. Eine weitere Methode wäre das Einbinden von Wetterdiensten. Über entsprechende Karten lässt sich die mögliche Wellenentwicklung schon weit im Voraus planen. Das Boot könnte so automatisch entscheiden, eine Mission gar nicht erst anzutreten, wenn die

Gefahr zu groß wäre. Hierbei wäre eventuell eine Anbindung an das Internet notwendig, was wiederum ausreichende Sicherheitsvorkehrungen voraussetzt.

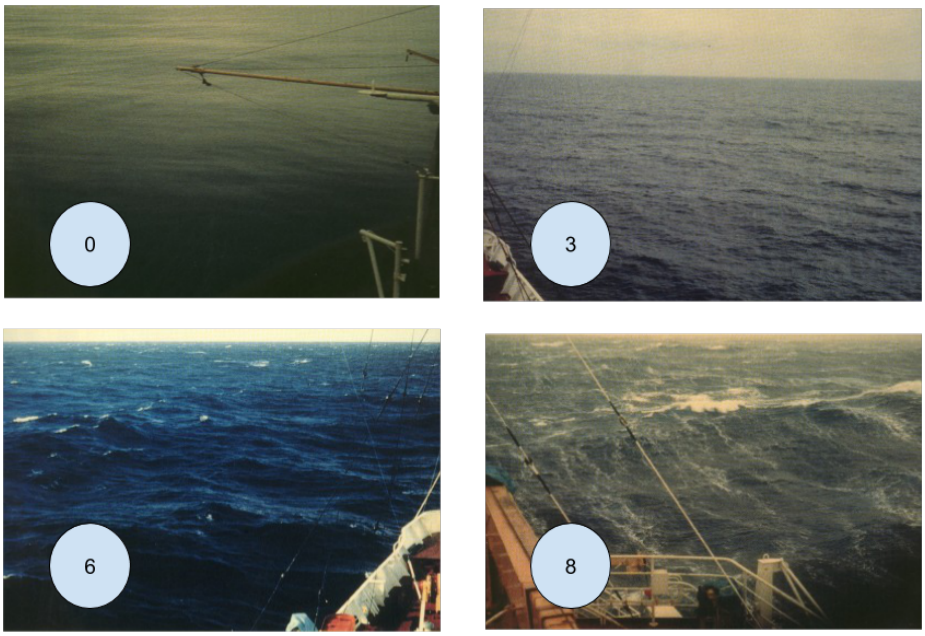


Abbildung A.16: Wellenentwicklung bei entsprechenden Windstärken

Die Windstärke ist jedoch nicht der einzige Faktor, der für die Ausprägung der Wellen verantwortlich ist. Unterirdische Plattenverschiebungen oder ein vorbeifahrendes Schiff können die Beschaffenheit der Wasseroberfläche um das Boot herum verändern. Deshalb reicht es bei Maßnahmen gegen das Kentern nicht, nur den Wind zu berücksichtigen. Ein Gyrosensor kann die Lageveränderung erfassen, aus der wiederum die Wellenstärke abgeleitet werden kann. Allerdings ist die Vorhersage der Wellenentwicklung hierbei eingeschränkt und es bleibt nicht viel Zeit für Vorbeugungsmaßnahmen.

Nun bleibt die Frage zu klären, wie verhindert werden kann, dass das Boot kentert. Zum einen können Gebiete mit hoher Wellenausprägung vermieden werden. Wenn ein Unwetter oder starker Wind bekannt ist, sollte das Boot die Mission gar nicht erst antreten oder rechtzeitig abbrechen. Bei starkem Wellengang muss das Boot harte Kursänderungen vermeiden, da er gerade in diesem Zeitpunkt besonders für seitliche Wellen anfällig ist. Es wäre auch möglich die Form des Bootes durch seitliche Flossen zu stabilisieren und den Schwerpunkt so weit wie möglich in der Mitte und tief zu halten [79]. Hier sollte man auf hohe Sendemaste und schwere Bauteile achten. Eine fest an-

gebrachte Abdeckung des Bootes ist sinnvoll, um die Bauteile vor Wind und starkem Regen zu schützen.

Strömung

Auch Strömungen können für die Steuerung der Observationsplattform problematisch sein. Hierbei werden Strömungen anhand ihrer Dauer, Ursache und Lage unterschieden. Während einige Strömungen lange an derselben Stelle aufzufinden sind, können andere kurzfristig entstehen. Strömungen können in der Tiefe, an der Meeresoberfläche und in Küstenregionen auftreten. Sie weisen meist einen Unterschied in der Wasserdichte auf, was durch eine andere Temperatur und abweichenden Salzgehalt bedingt ist. Dementsprechend kommt es zu Auswirkungen auf die Fahrtrichtung des Bootes. Seine Aufgabe ist es, entsprechend der Strömungsrichtung und Geschwindigkeit den eigenen Kurs anzupassen.

Umweltschutz

Neben umweltschützenden Maßnahmen, wie das Zurückgreifen auf alternative Energiequellen, soll in diesem Abschnitt viel mehr auf kritische Naturgebiete hingewiesen werden. Hier sind vor allem Naturschutzgebiete interessant, in denen teilweise Durchfahrtssperren herrschen. Diese muss die Observationsplattform selbstständig erkennen und umfahren können. In flachen Gewässern wachsen oft Gräser oder andere Wasserpflanzen. Diese spielen oft eine wichtige Rolle für das ökologische System und dürfen nicht zerstört werden. Auch Tiere und Menschen sind ein fester Bestandteil der Umgebung. Über entsprechende Sensoren und Bilderkennung wäre es möglich, dass ein autonom fahrendes Boot Unfällen aus dem Weg geht.

Seezeichen

Seezeichen helfen bei der Navigation durch befahrene Gewässer oder beim Anfahren einer Küstenregion. Hierbei sind vor allem Leuchttürme, Nebelhörner und Bojen bekannt. Um diese Seezeichen möglichst einheitlich zu gestalten, gibt es die International Association of Lighthouse Authorities (IALA). Diese stellt Standards für die Navigation auf Gewässern dar und trennt sich in IALA-A und IALA-B. Erstere gilt unter anderem für Europa. Dabei unterscheiden sie sich nicht allzu stark, worauf im weiteren Verlauf noch eingegangen wird. Im wesentlichen sind das Lateral- und das Kardinalsystem von Bedeutung [37].

Lateralsystem

Das Lateralsystem regelt die einzuhaltende Position eines Schiffes auf dem Fahrtwasser. Dabei werden Seezeichen eingesetzt, die drei wesentliche Merkmale aufweisen:

- Farbe: Für die Seezeichen werden die Farben Grün und Rot verwendet.
- Form: Bei der Form wird in Stumpf und Spitz unterschieden. Dabei spielt es keine Rolle, ob das Seezeichen eine Boje, oder am Festland angebracht ist. Meist werden die Formen durch einen spitzen Kegel und einen Quader realisiert.
- Nummerierung: Die Seezeichen werden fortlaufend durchnummeriert. Dabei ist besonders, dass die Seezeichen auf einer Seite jeweils die geraden Ziffern zugeordnet bekommen, während die andere Seite mit den ungeraden Ziffern gekennzeichnet ist.

Anhand des Beispiels auf Abb. A.17 lässt sich erkennen, welche Bedeutung die zuvor genannten Merkmale haben. Kommt ein Schiff von See und will ins Binnenland fahren, muss die rechte Seite den grünen und spitzen Seezeichen mit ungerader Nummerierung zugewandt sein. Beim Fahren in entgegengesetzte Richtung ist auch die Ausrichtung der Merkmale entgegengesetzt.

In Regionen mit dem IALA-B Standard besteht der einzige Unterschied darin, dass die Bedeutung der Merkmale Farbe und Nummerierung genau entgegengesetzt zum IALA-A Standard sind.

Kardinalsystem

Dieses System wird zur Kennzeichnung von Untiefen und Gefahrenstellen eingesetzt. Im Bereich rund um die Gefahrenstelle werden in allen Himmelsrichtungen Seezeichen aufgestellt, die annähernde Schiffe über das Risiko informieren. Abbildung A.18 zeigt eine Gefahrenquelle mit zugehörigen Warnzeichen. Vereinzelte Gefahrenstellen werden auch durch rote und schwarze Baken gekennzeichnet. Neuerdings werden Gefahrenstellen auch mit Radarantwortbaken ausgestattet.

Verkehrstrennungsgebiete

Ein Verkehrstrennungsgebiet (VTG) versucht den Verkehrsfluss auf dem Wasser zu ordnen. Ein VTG besitzt getrennte Fahrspuren in unterschiedliche Richtungen und liegt meistens an Engpässen oder Übergängen vom Meer ins Binnenland. Auch hier gelten die allgemeinen Verkehrsregeln und Navigationssysteme über die Seezeichen. Um den Verkehr nicht zu behindern, muss die mobile Observationsplattform in der Lage sein, auch in diesen Gebieten kollisionsfrei zu navigieren [116].

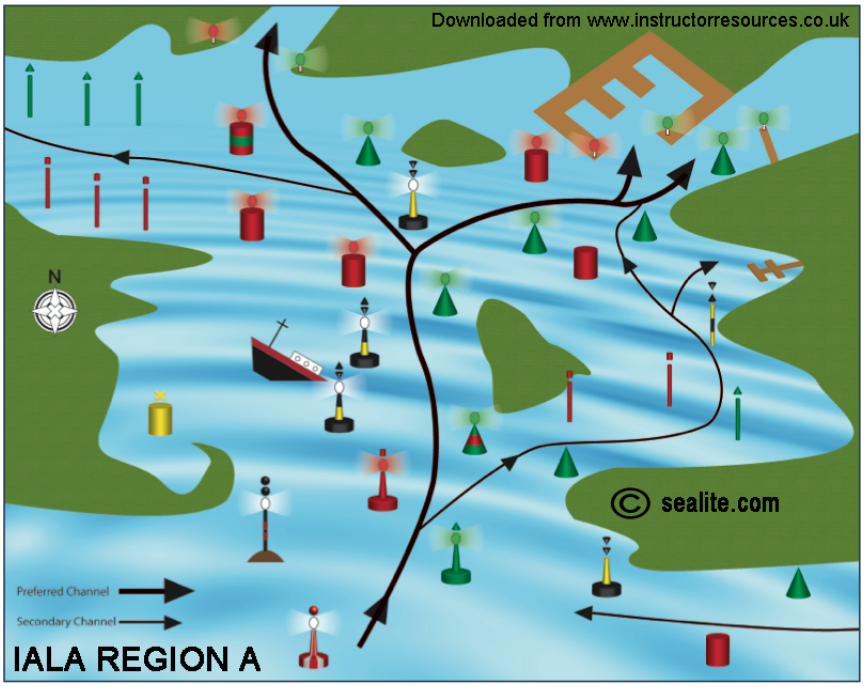


Abbildung A.17: Seezeichen zur Regelung der Fahrtseite [aus [4]]

Erkennung von Seezeichen

Auch ein autonom fahrendes Wasserfahrzeug muss sich an die Richtlinien auf See halten. Das Fehlen von Schiffsbesatzung führt dazu, dass die Erkennung von Seezeichen und Gefahren auf andere Weise realisiert werden muss. Ein Radargerät würde mitunter andere Schiffe und einige Gefahrenstellen erkennen können. Hierbei stellt die Entfernung zum anderen Objekt ein Problem dar. Auch um die Art und Bedeutung eines Seezeichens zu erkennen würde ein Radargerät nicht ausreichen.

Wie bereits erwähnt zeichnen sich Seezeichen vor allem durch ihre Form und Farbe aus. Deswegen liegt der Gedanke nahe, das Boot mit einer Kamera auszustatten und die Bilddaten auszuwerten. Da die Wahl der Farben darauf konzipiert wurde in Seegebieten gut erkannt zu werden, sollte es mit entsprechender Software kein allzu großes Problem darstellen, die Seezeichen auseinanderzuhalten. Auch Umrisse eines Objektes können über Software erkannt werden. Das Verwenden einer Kamera hat zudem den Vorteil, dass die Plattform auch auf größere Entfernung ferngesteuert werden kann. Im Falle eines Problems, kann immer noch aus der Ferne reagiert werden. Über einen Infrarotsensor ist es möglich die Entfernung zu einem Objekt abzuschätzen.

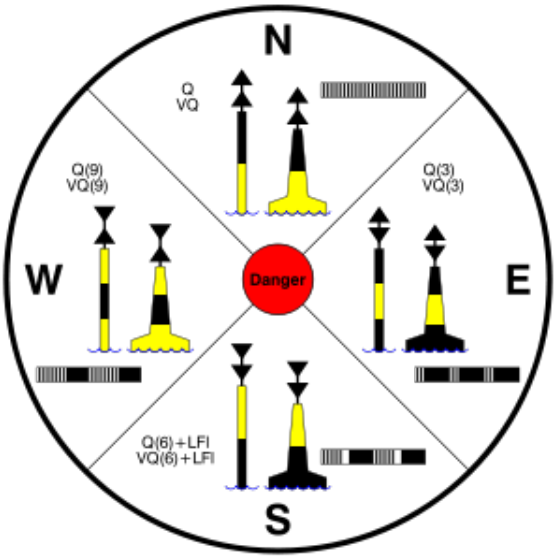


Abbildung A.18: Gefahrenquelle mit umstehenden Seezeichen (aus [131])

Seekarten

Seekarten stellen Seegebiete übersichtlich dar und besitzen unter anderem Angaben über die wichtigsten Seezeichen, Untiefen, Häfen und wichtige Schiffsrouten. Heutzutage sind vermehrt digitale Seekarten im Gebrauch. Die Vorteile sind die unproblematische Aktualisierung der Karten und die Unterstützung von Ortungsdiensten wie GPS. Für die Plattform wäre das Einbinden einer Seekarte notwendig um Routen für Missionen zu planen. Online Plattformen wie OpenSeaMap sind eine kostenlose Alternative zu teuren Seekarten. Durch Crowd Sourcing wird dieses Projekt ständig um neue Karteninformationen ergänzt, bietet aber jetzt schon eine großflächige Abdeckung der Seegebiete. Aktuell läuft eine Aktion, in der Tiefeninformationen zu Seegebieten gesammelt werden. Diese Karten lassen sich herunterladen und auch offline mit GPS-Systemen nutzen. Alternativ kann über ein Webinterface auf das Kartenmaterial zugegriffen werden. Anhand der Seekarten kann die Observationsplattform im Vorfeld bestimmte Routen verwerfen und bekannte Gefahrenstellen umgehen. Auch Naturschutzgebiete werden in den Seekarten gekennzeichnet, so kann über Kartenmaterial auf die maritime Umgebung Rücksicht genommen werden [7].

Wetterkarten

Neben Informationen zum Verkehr und Seerouten gibt es Karten, welche die aktuelle Wetterlage darstellen. Mithilfe dieser Informationen kann die Platt-

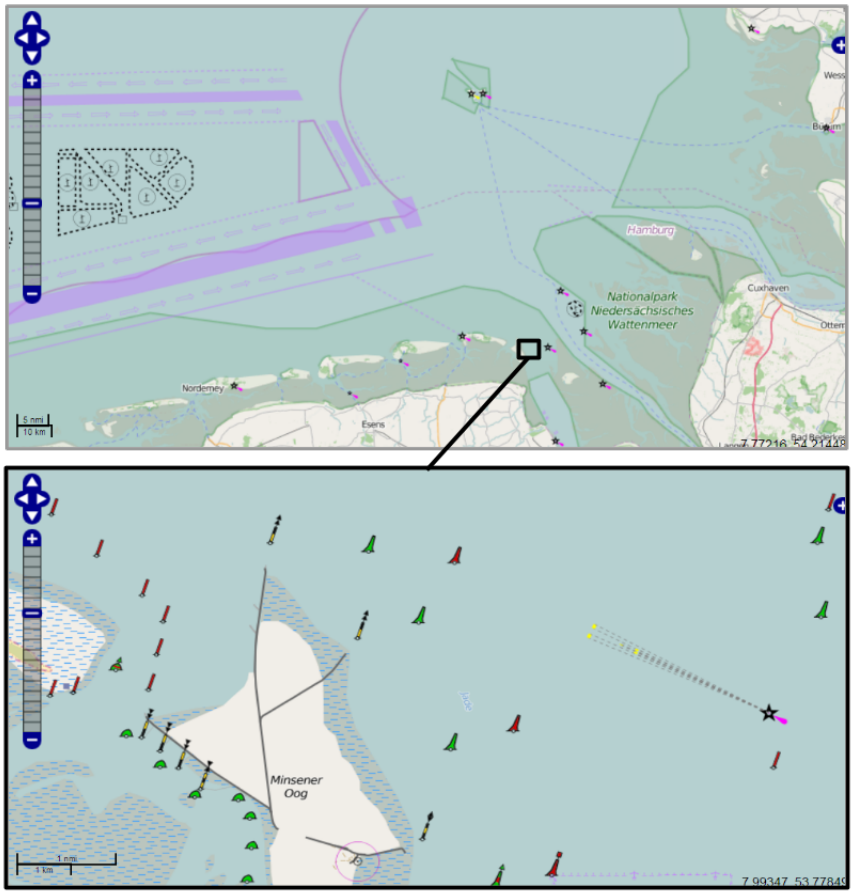


Abbildung A.19: Ausschnitt aus OpenSeaMap

form seine Missionen besser planen oder bei schlechter Witterung abbrechen. Auch lassen sich anhand der Wetterinformationen optimale Routen finden und Unwettergebiete umfahren. Somit kann ein Schaden am Schiff schon weit im Voraus verhindert werden, in dem es gar nicht erst zu einer Notsituation kommt [8].

A.5 Schiffssensorik

Von Maximilian Hipp, vorgetragen am 27. Oktober 2015.

Ein Ziel der MOPS IV-Projektgruppe wird es sein, sich mit den maritimen Sensoren zu befassen, um bekannte Modelle und Möglichkeiten auf das Projekt zu übertragen und anzuwenden. Hierbei steht die Kollisionsvermeidung im Fokus der Betrachtungen. Diese kann durch verschiedene Ansätze reali-

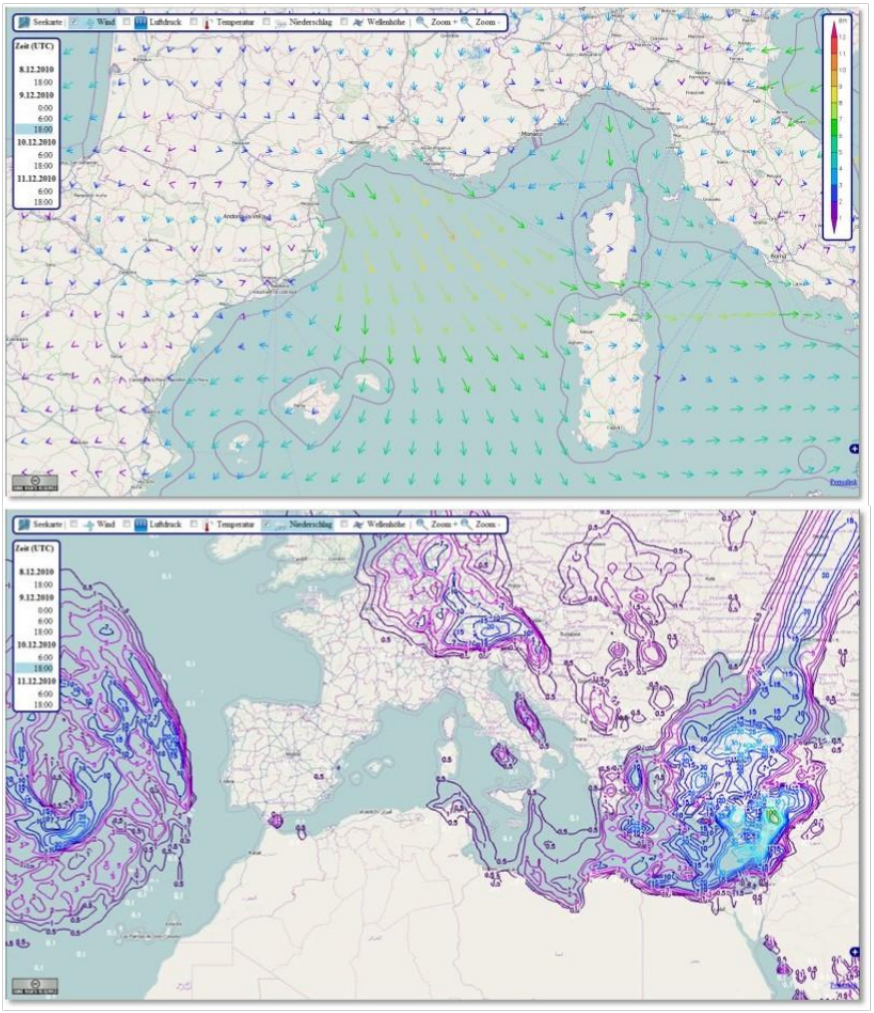


Abbildung A.20: Darstellung von Wind- und Niederschlagsinformationen

siert werden. Hierbei können Konzepte bisheriger maritimen Schiffssensorik einen möglichen Ansatz darstellen.

RADAR

Radar steht für „Radio Detection and Ranging“ und bezeichnet verschiedene Erkennungs- und Ortungsverfahren. Basis dieser Verfahren bilden elektromagnetische Funkwellen im Radiofrequenzbereich (siehe Abb. A.21).

Das Grundprinzip hinter einem Radar-Gerät ist das Aussenden von gebündelten elektromagnetischen Wellen (Primärsignal, links). Diese werden auf Objekten reflektiert und als sogenanntes „Echo“ wieder empfangen. An-

hand der Laufzeit oder Frequenzveränderung des Signals können Informationen über Entfernung, Geschwindigkeit oder Winkel des Objektes ermittelt werden. Relevante Radartypen stellen das Impuls- und das Dauerstrichradar dar.

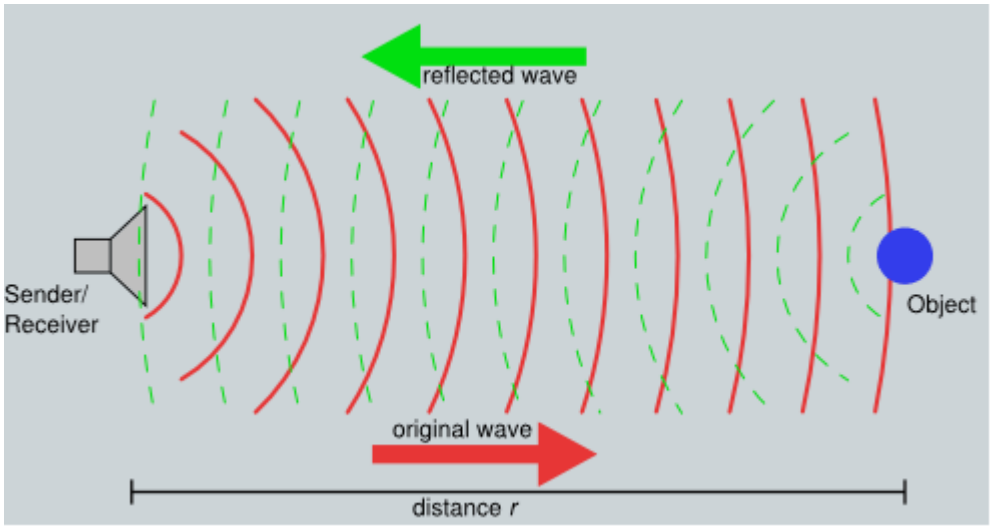


Abbildung A.21: Funktionsprinzip RADAR (aus [9])

Impulsradar

Das Impulsradar sendet kurze Impulse. Diese besitzen eine Dauer im unteren Mikrosekundenbereich. Anschließend wird auf das Echo gewartet. Aus der Dauer vom Absenden des Signals bis zum Empfang des Echos kann die Entfernung des Objektes ermittelt werden.

Die Vorteile sind die größere Mögliche Distanz, auf die mittels Radar beobachtet werden kann. Für die starken Impulse wird jedoch meist viel Energie benötigt.

Dauerstrichradar

Weiterhin gibt es das Dauerstrichradar (CW-Radar/„Continous Wave“-Radar genannt). Dieses arbeitet ununterbrochen und sendet ein dauerhaftes Signal aus. Dieses wird ebenso von Objekten reflektiert und wieder empfangen. Dadurch, dass das Signal dauerhaft ausgesandt wird, kann der reine Empfang des Echos keinen Aufschluss über die Entfernung des Objektes geben, sehr wohl jedoch über die Geschwindigkeit eines Objektes in Relation zum Sender/-Empfänger des Radar-Signals. Das Prinzip dahinter basiert auf dem Doppler-Effekt [120].

Eine Sonderform des Dauerstrichradars ist das sogenannte „Frequenzmodulierte Dauerstrichradar“ („frequency modulated continous wave radar“ = FMCW, siehe Abb. A.22). Dieses kann im Vergleich zum Dauerstrichradar zur Messung von Abstand und Höhe genutzt werden, da die gesendeten Signale eine wechselnde Frequenz besitzen. Auf Basis der Informationen von Aussendezeitpunkt und ausgesandter Frequenz zu diesem Zeitpunkt kann das empfangene Signal ausgewertet werden.

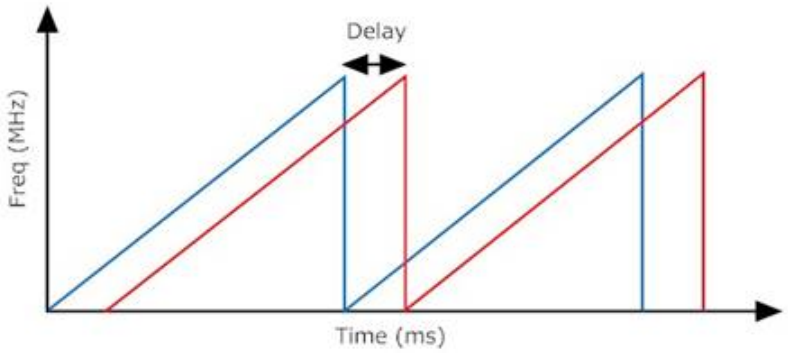


Abbildung A.22: Frequenzmoduliertes RADAR (aus [3])

Radarsensoren

Radarsensoren zählen zur externer Sensorik und wandelt die elektromagnetischen Wellen in elektrische Signale. Für Arduino gibt es bereits mehrere Ansätze für Radar-Module (siehe Abb. A.23 und A.24).

Abbildung A.23: Impulsradar (aus [46])

ARPA

ARPA („Automatic Radar Plotting Aid“) bezeichnet eine automatische Radar Plotter Einrichtung. Das System wertet Radar Informationen aus und

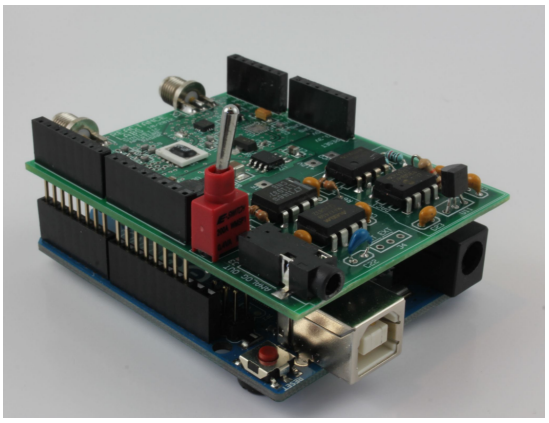


Abbildung A.24: FMCW-Radar (aus [10])

kann mittels einer grafischen Oberfläche alle Informationen übersichtlich und verständlich darstellen (siehe Abb. A.25). Als Beispiel können Schiffe angezeigt und verfolgt werden sowie deren Kurs und Geschwindigkeit abgebildet werden.

Weitere Funktionen können automatische Warnungen oder die Simulation von anstehenden Manövern sein. Auch die eigenen Kursdaten kann das Sytem plotten, sodass ein umfangreiches Hilfswerkzeug zur Navigation auf See entsteht. Zu den Informationsquellen eines ARPAS zählt das Radar, der Kompass (Meist Kreiselkompass) und die Fahrtmessanlage.

AIS

AIS (Automatic Identification System) bezeichnet ein Hilfsmittel zur Überwachung und Lenkung des maritimen Verkehrs. Das System basiert auf dem UKW Seefunkbereich und überträgt Informationen über Schiffe, deren Ladung, Anl der Passagiere, Geschwindigkeit, Schiffsname, Typ, Gefahrengüter und weitere Parameter. Ziel des Systems ist es Kollisionen zu vermeiden und die Lenkung des Schiffsverkehrs zu verbessern.Standortinformationen werden hierbei aus globalen Satelliten Systemen bereitgestellt wie z.B. GPS.

Vorteile des AIS Systems im Vergleich zu anderen Kolissionsverhütenden Systemen ist die wetter- und standortunabhängige Auswertung der Informationen. RADAR kann beispielsweise hinter Gebirge nicht empfangen werden (Radarschatten). Gleichzeitig ist ein AIS kein Ersatz für RADAR, da es viele Schiffe gibt, die kein AIS besitzen. Die Darstellung der AIS Informationen erfolgt mittels eines ARPAs oder einer elektronischen Seekarte (siehe Abb. A.26).

Seit 2004 sind alle Berufsschiffe über 300 Bruttoraumzahl in internationaler Fahrt dazu verpflichtet, ein AIS System zu betreiben. Seit 2008 auch

Abbildung A.25: Automatic Radar Plotting Aid (aus [125])

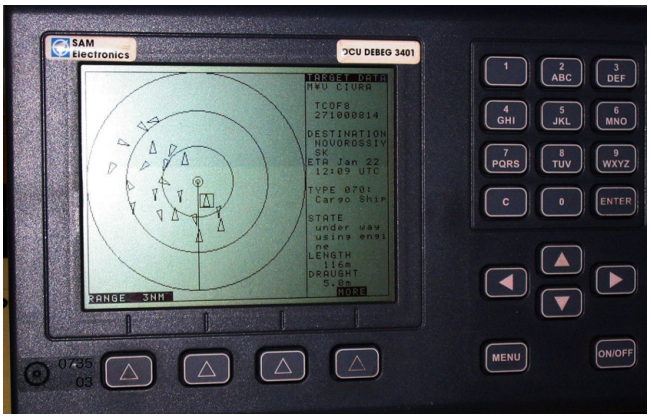


Abbildung A.26: AIS Informationen, dargestellt auf einem ARPA (aus [119])

nationale Fahrten mit über 500 Bruttoraumzahl. Auch Schiffe mit einer Länge von mehr als 20 Metern oder mehr als 50 Passagieren an Bord müssen mit einem AIS System ausgestattet sein.

AIS Systeme werden weiterhin nach verschiedenen Einsatzgebieten klassifiziert. Klasse A-Transceiver: Sind für Berufsschiffe vorgesehen und übertragen mit einer höheren Signalstärke, um von weiter entfernten Schiffen empfangen zu werden. Außerdem senden sie häufiger Signale aus.

Klasse B-Transceiver: Bezeichnet quasi abgespeckte Varianten von Klasse A-Transceivern. Sie senden mit weniger Signalstärke und sind somit kostengünstiger. Sie können für nicht ausrüstungspflichtige Schiffe genutzt werden.

ATON-Transceiver: Diese Transceiver können auf Tonnen und anderen Schifffahrtszeichen installiert werden und Informationen über Art und Position des Schifffahrtszeichens aussenden.

Kompass

Ein Kompass (siehe Abb. A.27) wird eingesetzt um mittels Erdmagnetfeld eine Richtung oder einen Kurs, eine Himmelsrichtung oder eine Peilrichtung zu bestimmen. Das Erdmagnetfeld wird hierbei zur Bestimmung der Nordrichtung eingesetzt. Trotz GPS und weiteren Hilfssystemen findet der Kompass weiterhin große Verwendung, da dieser auch unabhängig von Technik funktioniert und eine Kurswinkelbestimmung mittels eines Kompasses weiterhin genauer und schneller durchführbar ist. Auch unter Wasser stellt der Kompass eine gute Möglichkeit der Richtungsbestimmung dar, da GPS-Signale nur schwer unter die Wasseroberfläche gelangen. Heutzutage werden alternativ zum herkömmlichen Kompass digitale Magnetometer (siehe Abb. A.28) eingesetzt, die Signale des Erdmagnetfelds auswerten.

Abbildung A.27: Traditioneller Kompass (aus [122])

Ohne das Erdmagnetfeld kommen Kreiselkompasse und Sonnenkompasse aus. Der Kreiselkompass orientiert sich parallel zur Rotationsachse der Erde. Dadurch kann an ihm die Nord- und Südrichtung abgelesen werden. Der Nachteil eines Kreiselkompasses ist, dass er zwei bis vier Stunden für das Einschwingen benötigt. Kreiselkompasse werden primär auf langsam fahrenden Schiffen eingesetzt, da ein kurs- und geschwindigkeitsabhängiger Fahrfehler

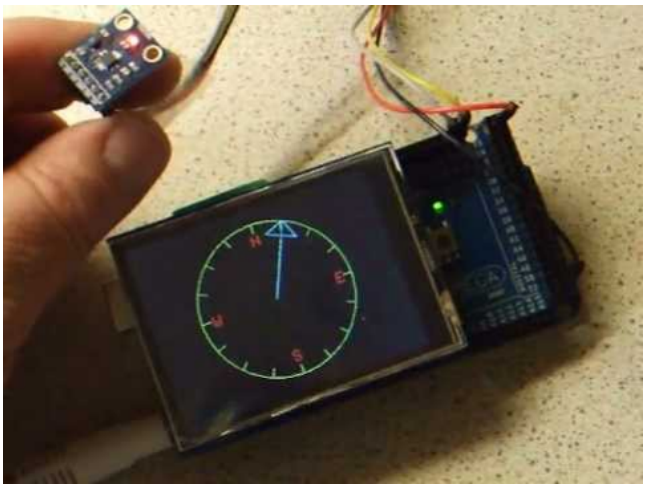


Abbildung A.28: Magnetometer (aus [1])

zu berücksichtigen ist und sie die astronomische Nordrichtung anzeigen können. Auf einen rotierenden Kreisel wirkt ein Drehmoment ein, solange der Kreisel nicht in Richtung Norden zeigt. Erst, wenn die Drehachse in Rotationsrichtung der Erde zeigt, wird das Drehmoment null.

GPS

GPS („Global Positioning System“) heißt eigentlich NAVSTAR GPS, dient der Positionsbestimmung mittels Satellitennavigation und wurde von den Amerikanern entwickelt. Weitere Satellitensysteme zur Positionsbestimmung heißen GLONASS (Russland), BEIDOU (China) und GALILEO (EU). Diese Systeme sind bisher jedoch nur eingeschränkt oder gar nicht nutzbar.

GPS-Endgeräte empfangen lediglich Informationen. Das Aussenden des eigenen Standortes ist nicht möglich. Für die Positionsbestimmung werden die Informationen von vier verschiedenen Satelliten benötigt. Aus den Signalen von drei Satelliten kann mittels Triangulation die Position in der Ebene berechnet werden, der vierte Satellit ergänzt die Höheninformation des zu bestimmenden Objektes (siehe Abb. A.29).

Bei dem Erstempfang von Satellitensignalen wird aus den ausgesandten Ephemeriden (Bahndaten) der zukünftige Satellitenstandort zu jedem Zeitpunkt berechnet. Beim Eintreffen des Signals kann dadurch ermittelt werden, wie lange das Signal zum Empfänger gebraucht hat und somit Rückschlüsse auf die Wegstrecke gezogen werden.

Oftmals funktioniert die GPS-Positionsbestimmung bis auf wenige Meter genau. Bei einer Flughöhe von 25000 km werden zumeist 24 bis 30 Satelliten

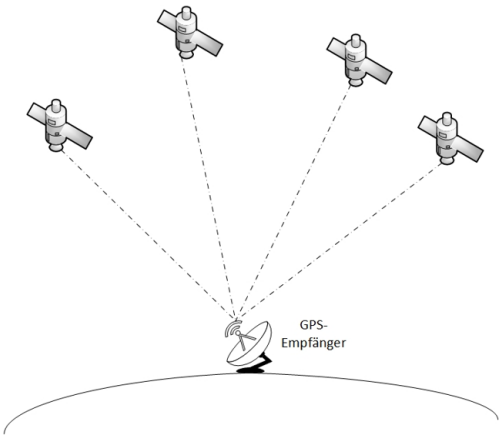


Abbildung A.29: Positionsbestimmung mittels vier Satelliten (aus [54])

zur Positionsbestimmung verwendet. Die Redundanz der Satelliten stellt auch bei bedeckten Wetterverhältnissen die optimale Empfangsqualität sicher.

Zur Kommunikation zwischen der genutzten Empfangshardware kann das NMEA 0183 Protokoll genutzt werden. Dieses wurde auch von MOPS III genutzt. Hinter diesem entwickelten Standard verbirgt sich eine Vereinigung von Herstellern, Vertreibern und Forschungseinrichtungen, welche sich das Ziel gesetzt haben, GPS-Daten mittels eines beliebigen Navigationsprogrammes auswerten zu können. Der Standard basiert auf einer RS422-Schnittstelle und einer Definition von Datensätzen [6]. An dem MOPS befindet sich ein GM65 Empfänger am Bug.

Funkortung

Mittels Funkortung kann z.B. ein Schiff seine aktuelle Lage in Relation zu Ortungspunkten ermitteln. Hierbei unterscheidet man zwischen Entfernungsverfahren und Peilverfahren. Richtsendeverfahren

Eine oder mehrere ortsfeste Funkstellen senden modulierte Wellen aus. Diese Modulationen sind als Richtungsinformationen interpretierbar und können von dem Schiff akustisch oder optisch dargestellt werden.

Entfernungsverfahren

Bei den Entfernungsverfahren wird die Entfernung zu einer Sendestation ermittelt, indem ein elektrischer Impuls zu einem aktiven Rückstrahler ausgestrahlt wird. Der Empfänger des Rücksendesignals kann aus der Signallaufzeit seine eigene Position relativ zur Rücksendestation ermitteln. Da die Standorte der Sendestationen bekannt sind, lässt sich die eigene Position ableiten. Für die genaue Positionsbestimmung sind wie bei GPS der Empfang der Signale

aus drei Quellen erforderlich. In der Seefahrt kann außerdem das Hyperbelverfahren genutzt werden, wenn die Anzahl der zur Verfügung stehenden Signale lediglich zwei beträgt.

Richtempfangsverfahren

Bei Richtempfangsverfahren (Peilverfahren) wird die Herkunftsrichtung von Signalen bestimmt. Als Beispiel lässt sich der Radiokompass anführen. Dieser peilt den Sender an, dessen Frequenz eingestellt ist. Eine weitere Anzeige im Schiff zeigt im nächsten Schritt den Winkel der eigenen Position (Längsachse) in Relation zur Sendestation an. Mittels Kreuzpeilung kann somit aus zwei angepeilten Stationen die eigene Position hergeleitet werden.

Differenzentfernungsmessverfahren

Bei diesen Verfahren wird die Laufzeitdifferenz von Impulsen gemessen. Zu den Verfahren gehört zum Beispiel das LORAN Verfahren. Das Verfahren basiert auf zu Ketten gruppierten Sendestationen, die aus einem Hauptsender und zwei oder mehr Nebensendern besteht, die im Abstand von einigen hundert Kilometern platziert wurden. Diese Ketten senden ein vordefiniertes Schema an Impulsen, die von einem Schiff empfangen werden können. Aus der zeitlichen Differenz, mit der die Signale bei dem Schiff eintreffen kann das Schiff seine Position berechnen.

Fazit

Dank der Informationen konnte ein Einblick in die technischen Bereiche maritimer Sensorik zur Navigation erlangt werden. Da eine Anforderung an den MOPS ist, Kollisionen zu verhüten müssen sich zwangsläufig Gedanken über die Art und Weise der Erkennung von Objekten und der Auswertung der Daten gemacht werden. An dem OFFIS Oldenburg stehen Radar-Messgeräte zur Verfügung, sodass dieses System durchaus in Betracht käme, zumindest genauer betrachtet zu werden für den Einsatz am MOPS. Da die Anschaffung von geeigneter Hardware ansonsten sehr teuer wäre, würde sich diese Methode gut eignen.

Die Erkenntnisse über AIS und ARPA kann man sich ebenfalls zu nutze machen. Basierend auf den Einsatzgebieten des MOPS, könnten stationäre Hindernisse bereits mit AIS Sendern ausgestattet worden sein, wodurch eine Erkennung dieser hinfällig wäre.

A.6 Projektmanagement

Von Zahra Paya, vorgetragen am 27. Oktober 2015.

Projektmanagement besteht aus zwei Teilen: Projekt und Management.

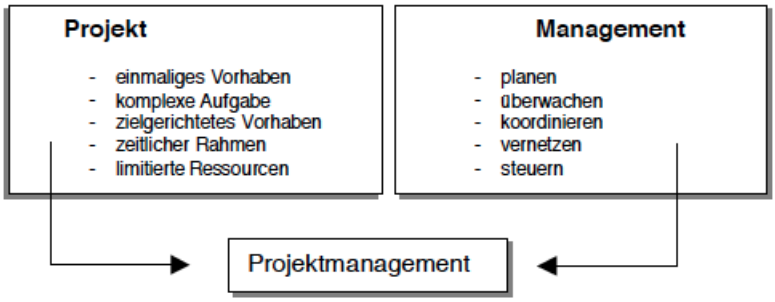


Abbildung A.30: Projektmanagment (aus [17])

Laut Duden bedeutet Projektmanagement: „Gesamtheit der Planungs-, Leitungs- und Kontrollaktivitäten, die bei Projekten anfallen“ [14].

Es gibt oft in den Projekten verschiedene Experten aus mehreren Abteilungen, die in einem Team gemeinsam arbeiten sollen, um komplexe und neuartige Aufgaben lösen zu können. Projektmanagement ermöglicht höhere Flexibilität und schnellere Entscheidungswege und zeigt einen erhöhten Planungsaufwand zu Projektbeginn. Es muss in vielfacher Hinsicht vom Projektstart bis zum Projektende planend und steuernd einwirken, um ein Projekt zum erfolgreichen Abschluss zu bringen [18].

Projektmanagement-Methoden

Der Begriff „Methode“ soll nicht mit dem Begriff „Werkzeug“ verwechselt werden. Die Methode beschreibt vielmehr die Art und Weise, in welcher spezifischen Werkzeuge eingesetzt werden, um ein Ziel zu erreichen. Die Methodik entscheidet darüber, welche Werkzeuge in welcher Reihenfolge angewandt werden. Im Projektmanagement ist die Festlegung einer bestimmten Methode der Grundstein für die Planung eines Projekts [19].

In dieser Arbeit werden die Methoden wie Wasserfallmodell, V-Modell, Spiralmodell, Scrum und Kanban definiert.

Wasserfallmodell

Wasserfallmodell ist ein lineares Vorgehensmodell, in dem jede Phase vollständig abgeschlossen sein muss, bevor die nächste Phase beginnen kann. Am Ende jeder Phase erfolgt eine Überprüfung, um festzustellen, ob das Projekt auf dem richtigen Weg ist, sonst darf es nicht fortgesetzt werden. In diesem Modell beginnt der Test erst, nachdem die Entwicklung abgeschlossen ist [29].

Die sequenziellen Phasen im Wasserfallmodell:[25]

Analyse-Phase Alle Anforderungen des Systems, das entwickelt werden soll, werden in dieser Phase erfasst und dokumentiert.

Die Anforderungsspezifikationen aus der ersten Phase werden in dieser Phase untersucht und der Systementwurf wird vorbereitet. Der Systementwurf hilft sowohl bei der Angabe von Hardware- und Systemanforderungen, als auch bei der Definition der Gesamtsystemarchitektur.

Systementwurf-Phase Die Anforderungsspezifikationen aus der ersten Phase werden in dieser Phase untersucht und der Systementwurf wird vorbereitet. Der Systementwurf hilft sowohl bei der Angabe von Hardware- und Systemanforderungen, als auch bei der Definition der Gesamtsystemarchitektur.

Implementierung-Phase Mit Eingängen aus dem Systementwurf wird das System zunächst in kleine Programme (Einheiten), die in der nächsten Stufe integriert werden entwickelt. Jede Einheit wird entwickelt und getestet auf ihre Funktionalität, die als „Unit Testing“ bezeichnet wird.

Integrations- und Test-Phase Alle in der Implementierungsphase entwickelten Einheiten werden nach der Prüfung jeder Einheit in eine Anlage integriert.

Einsatz-Phase Sobald das funktionale und nicht funktionale Testen erfolgt ist, wird das Produkt in der Kundenumgebung bereitgestellt oder in den Markt entlassen.

Die Abb. A.31 zeigt das Wasserfall-Diagramm.

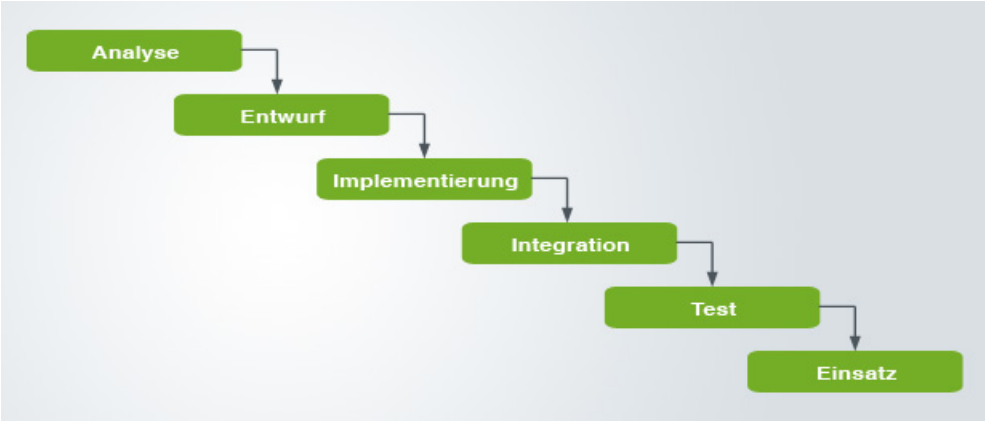


Abbildung A.31: Wasserfall-Diagramm (aus [26])

Es gibt einfache Möglichkeiten der Planung und Kontrolle, da jede Phase spezifische Leistungen und eine Überprüfung hat und Phasen werden eins

nach dem anderen verarbeitet und abgeschlossen werden. Aber Fehler werden unter Umständen spät erkannt und müssen mit erheblichem Aufwand entfernt werden. Es gibt auch Schwierigkeiten durch Rückschritte und Änderungen.

V-Modell

V-Modell bedeutet Verifizierung und Validierung. Das V-förmiges Phasenmodell ist eine sequentielle Ausführungspfad von Prozessen. Jede Phase muss abgeschlossen sein, bevor die nächste Phase beginnt. Das Testen des Produkts ist parallel mit der entsprechenden Phase der Entwicklung geplant [28].

Aktivitäten, die auf der Entwurf Seite anfallen, sind die folgenden: [27]

Systemanforderungsanalyse Hier werden Informationen gesammelt, die über das vorgeschlagene System und der Endbenutzer, mit denen Software-Anforderungsspezifikation Dokument geschrieben wird, das die Grundlage des Projekts ist.

System-Architektur Hier definiert man die Schnittstellen, Datenbanktabellen und deren Abhängigkeiten.

System-Entwurf Das Ziel ist, die Design Funktionalitäten der Software vorzubereiten.

Software-Architektur Hier wird der Systemtest definiert, die Anforderungsdefinition liefert den Abnahmetest und sichert damit die Korrektheit des fertigen Produktes.

Software-Entwurf Das gesamte System gliedert sich in kleine Module, die dann wiederum in das Gesamtsystem integriert sind.

Aktivitäten, die auf der Validierungsseite anfallen, sind die folgenden:

Unit-Test Dies ist die erste Phase der Validierung, in der kleine Module entwickelt werden, um festzustellen, ob sie für einen bestimmten Zweck passen.

Integrations-Test Sobald die Module bereit sind, werden sie integriert. Dieser Schritt hilft, um Defekte in der Oberfläche und die Wechselwirkung zwischen zwei verschiedenen Modulen zu bestimmen.

System-Integration Sie überprüft das Artefakt der Phase System-Entwurf.

Abnahme und Nutzung Hier wird überprüft, ob die Anforderungen erfüllt wurden und die Nutzer ihre Ziele erreichen können.

Folgende Abb.A.32 zeigt das V-Modell-Diagramm.

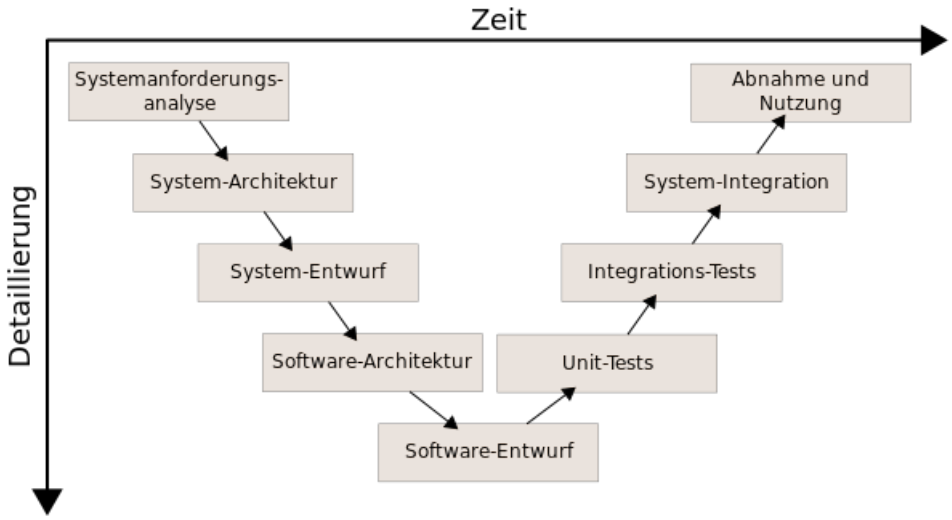


Abbildung A.32: V-Modell (aus [132])

Es gibt zeitlich nachfolgende Testphasen, die in der Realisierungsphase dargestellt werden. Eine Gegenüberstellung führt zu möglichst hoher Testabdeckung, da die Spezifikation der jeweiligen Entwicklungsstufen die Grundlage für die Tests in den entsprechenden Teststufen sind.

Jedoch ist das V-Modell zu allgemein, zu generisch und mit viel Bürokratie verbunden, insbesondere werden die hohen Anforderungen an die Dokumentation während der Entwicklung in kleinen Projekten als belastend und für kleine Projekte eher ungeeignet empfunden.

Spiralmodell

Der gesamte Entwicklungsprozess wird beim Spiralmodell auf einer aus vier Quadranten bestehenden Spirale abgebildet. Wegen einer Risikoabschätzung wird der weitere Entwicklungsverlauf in jedem Zyklus abgesichert. Die Entwicklungsergebnisse werden von den betroffenen Benutzern am Ende jedes Zyklus überprüft. Erst nach Abnahme der Ergebnisse wird mit dem nächsten Zyklus begonnen [72].

Die Phasen im Spiralmodell:

Analyse Hier werden die Ziele, Alternative und Rahmenbedingungen festgelegt.

Entwurf Hier werden Alternativen bewertet, Risiken werden identifiziert und aufgelöst.

Realisierung Das Produkt der nächsten Phase wird entwickelt und verifiziert.

Einsatz In dieser Phase werden die Schritte 1-3 überprüft und die Projektfortsetzung geplant.

Die Abb.A.33 zeigt das Spiralmodell-Diagramm.

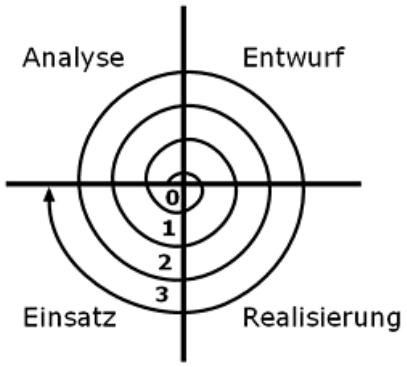


Abbildung A.33: Spiralmodell-Diagramm (aus [24])

Beim Spiralmodell können vorhandene Modelle implementiert werden. Durch das ständige Simulieren und Analysieren werden vorhandene Risiken erkannt, abgeschätzt und minimiert. Dieses Modell benötigt wegen oft auftretender Neuentscheidungen sehr hohen Managementaufwand. Es funktioniert nicht gut für kleine Projekte.

Scrum

Scrum wird als empirische, inkrementelle und iterative Methode im Projektmanagement beschrieben. Es erlaubt die schnelle Abstimmung im Team und ermöglicht adaptives Planen durch die Fokussierung auf wenige zentrale Regeln und Selbstorganisation [13].

Die Rollen im Scrum sind folgende: [23]

Scrum Master Er ist verantwortlich dafür, dass ein Scrum-Projekt zum Erfolg wird und führt die entsprechenden Scrum-Regeln ein. Er moderiert die anfallenden Meetings und kümmert sich um die Beseitigung etwaiger Hindernisse im Scrum-Prozess. Er übernimmt ausschließlich administrative Aufgaben, ohne dabei selbst zu entwickeln und ohne dabei konkrete Arbeitsanweisungen geben zu dürfen.

Product Owner Er bestimmt die Anforderungen und die strategischen Richtungen wie Priorisierung von Anforderungen, die in Abstimmung mit

dem Entwicklungsteam im sog. Product Backlog als sog. User Stories erfasst werden. Damit werden die benötigten Funktionalitäten aus der Sicht des Benutzers beschrieben. Er entscheidet über Features, Kosten und Timings. Entscheidungen des Product Owners sind dabei verbindlich, denn er ist verantwortlich für die korrekte Erstellung eines Produktes.

Product Backlog Dabei gibt es eine priorisierte und rein nutzerorientierte Liste mit Anforderungen, die das zu entwickelnde Produkt berücksichtigen muss. Es enthält User Stories, die in einem Sprint implementiert werden sollen.

User Story User Stories formulieren die Product Backlog-Einträge und beschreiben, welche Produkteigenschaft der Benutzer will.

Team Scrum-Teams sind selbstorganisierend und interdisziplinär und entscheiden selbst, wie sie ihre Arbeit am besten gestalten, statt dieses durch andere Personen außerhalb des Teams vorgegeben zu bekommen.

Sprint Backlog Sprint Backlog ist eine Liste der Aufgaben, die von dem Scrum-Team identifiziert werden, um während des Sprints abgeschlossen zu werden.

Sprint Review Sie steht am Ende eines Sprints und das Team stellt dem Product Owner vor, was es im Sprint geschafft hat. Der Product Owner entscheidet dann mithilfe von definierten Kriterien, ob das Ergebnis abgenommen werden kann oder nicht.

Abbildung A.34 zeigt das Grundprinzip des Scrum-Modell.

Bei Scrum gibt es auf der einen Seite hohe Effektivität durch Selbstorganisation und hohe Transparenz durch regelmäßige Meetings und Backlogs. Hohe Flexibilität wird durch adaptives Planen erreicht. Auf der anderen Seite wird die Koordination mehrerer Entwicklungsteams bei Großprojekten erschwert. Es gibt nur wenige konkrete Handlungsempfehlungen.

Kanban

Kanban ist ein relativ neues Vorgehensmodell, damit man Entwicklungsprozesse einfacher verwalten kann. Im Vergleich zu Scrum werden weniger strukturierende Regelwerke definiert. Der bestehende Prozess wird beim Kanban in kleinen Schritten verbessert. Der Entwicklungsprozess wird auf einem Kanban-Board in Spalten unterteilt.

Die einzelnen Anforderungen werden auf Karteikarten geschrieben, die zur Veranschaulichung des Entwicklungsprozesses durch die einzelnen Stationen fortgeschrieben werden. Es darf immer nur eine festgelegte Anzahl von Tickets für jede Spalte bearbeitet werden. Man kann sofort sehen, wie viele Aufgaben

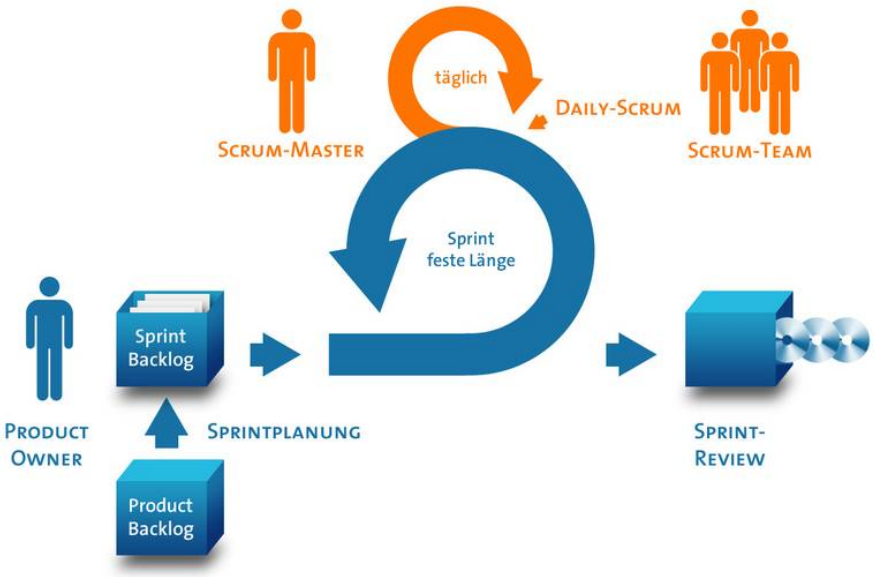


Abbildung A.34: Grundprinzip des Scrum-Modells (aus [22])

in „To do“, in „Bearbeitung“ oder in „Abnahme“ stehen und wie viele schon erledigt sind [38].

Abbildung A.35 zeigt das Kanban-Bord.

Wegen direkter Visualisierung des Projektablaufs kann man Probleme anhand von Ticket-Häufungen schneller bemerken. Es ist ein umfassendes System, das im Prinzip auf jeden bestehenden Entwicklungsprozess aufgesetzt werden kann.

Da es vor jeder Stufe die Puffer gibt, geht Kanban häufig mit langen Durchlaufzeiten einher. Kanban ist für große Projekte ungeeignet und je größer das Team ist, desto unübersichtlicher wird das Kanban-Projektmanagement.

Projektmanagement-Tools

Um die Projektmanagementsmethode verwenden zu können, stehen verschiedene Projektmanagment-Tools zu Verfügung, die gleichzeitig die Aufgaben des Managements und die operative Projektarbeit unterstützen. Nachstehend werden einige diesen Tools vorgestellt.

Jira

Jira ist ein webbasiertes Tool für Projektmanagement. Es dient zur Organisation des Projektprozesses. Dadurch kann man sowohl Scrum als auch Kanban

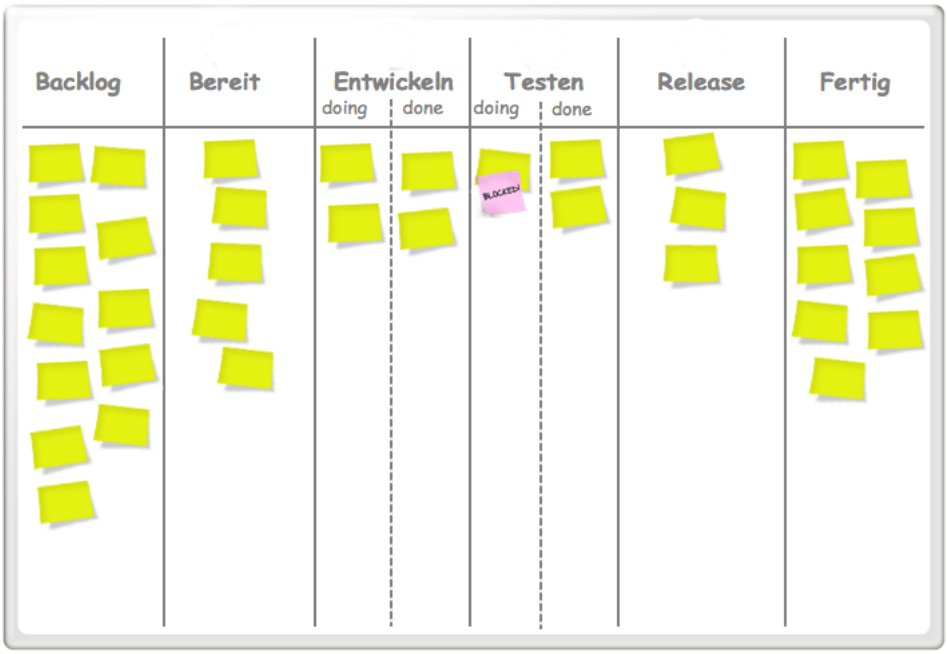


Abbildung A.35: Kanban-Board (aus [12])

verwenden. Jira besetzt Kanban-Bord und dabei werden die Aufgaben als Ticket erstellt. Jedes Ticket zeigt die Aufgaben und die Zuweisung der Personen. Die Tickets können einfach nur editiert werden und jede Änderung daran wird aufgezeichnet. Aufgaben können priorisiert, verfolgt, geprüft und verwaltet werden. Die Scrum-Rollen werden auch im Jira gezeigt. Der Product Owner verwaltet mit dem Aufnehmen der User Stories seinen Backlog in Jira [15].

Die Abb. A.36 zeigt Jira-Tool.

CodeBeamer

Ein anderes alternatives Tool, das beide Methoden, Scrum und Kanban, benutzt, heißt CodeBeamer. Die Verwendung des CodeBeamers erlaubt es verteilten Teammitgliedern, effektiver zusammenarbeiten zu können. Es hat Echtzeit-Transparenz und Echtzeit-Kontrolle über die vielfältigen Aktivitäten im Projekt und ermöglicht Traceability und Impact-Analyse über den gesamten Projektlebenszyklus. Es gibt Progress- und Trend-Reports für Projektfortschritt und Projektqualität, die anhand historischer Daten des Projektes erzeugt werden. Wie beim Jira kann man hier auch Produkt Backlog,

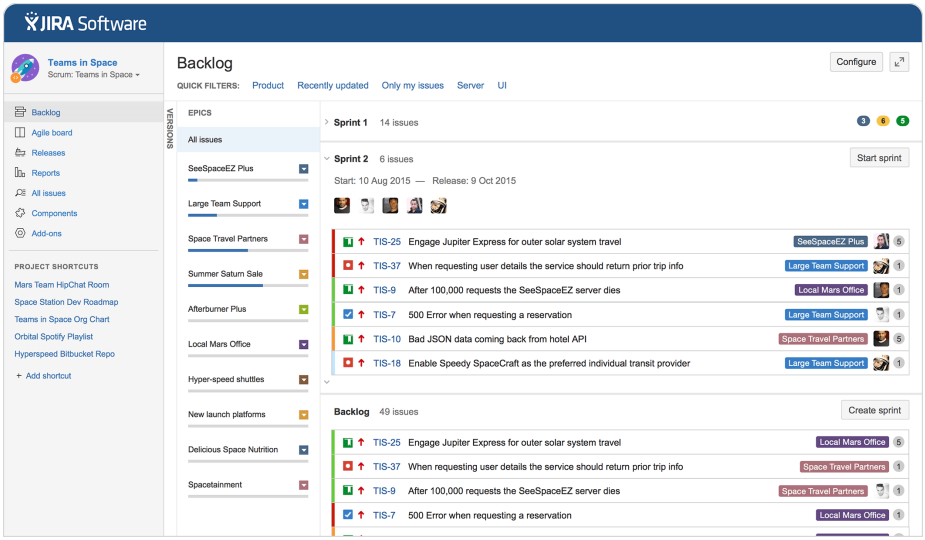


Abbildung A.36: Jira-Tool (aus [16])

User Stories usw. definieren. In Kanban-Bord beschreibt man die Tasks und kann damit den Stand einzelner Stories mit Drag und Drop ändern oder diese Mitgliedern des Projekt-Teams zuordnen [67].

Die Abb.A.37 zeigt das CodeBeamer–Tool.

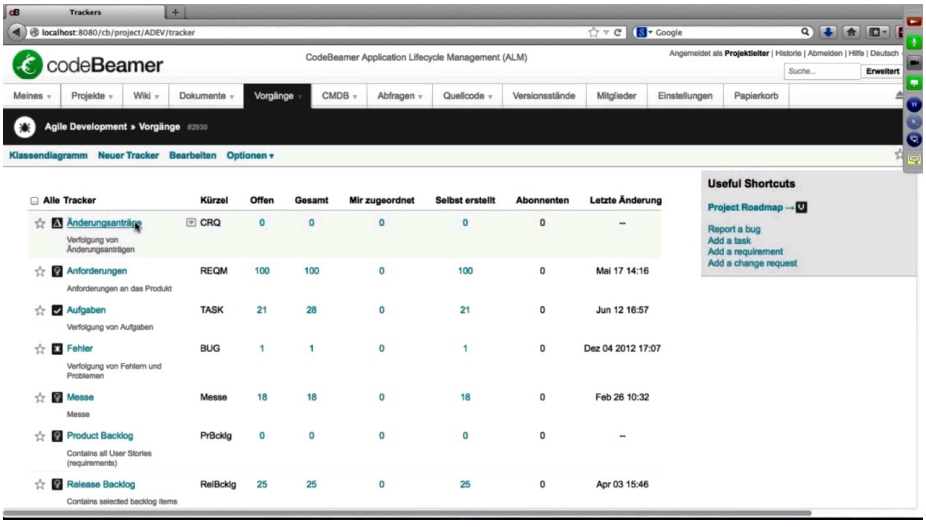


Abbildung A.37: CodeBeamer-Tool (aus [68])

Redmine

Redmine ist eine freie, webbasierte Projektmanagementsoftware. Sie wird für Benutzerverwaltung, Projektverwaltung und Ticketverwaltung benutzt. Bei Redmine kann man unterschiedliche Projekte anlegen oder große Projekte in den einzelnen Systembausteinen trennen und verschiedene Rolle definieren. Tasks von Redmine zeigen die Aufgabenverteilung und jeder Task enthält die Priorität, die aufgewendete Zeit, die Kategorie usw. Gantt-Diagramme in Redmine geben eine Übersicht über abgeschlossene und noch zu erledigende Arbeiten, dabei erfolgt die Darstellung grafisch in Form von Balken [21].

Die Abb.A.38 zeigt das Redmine-Tool.

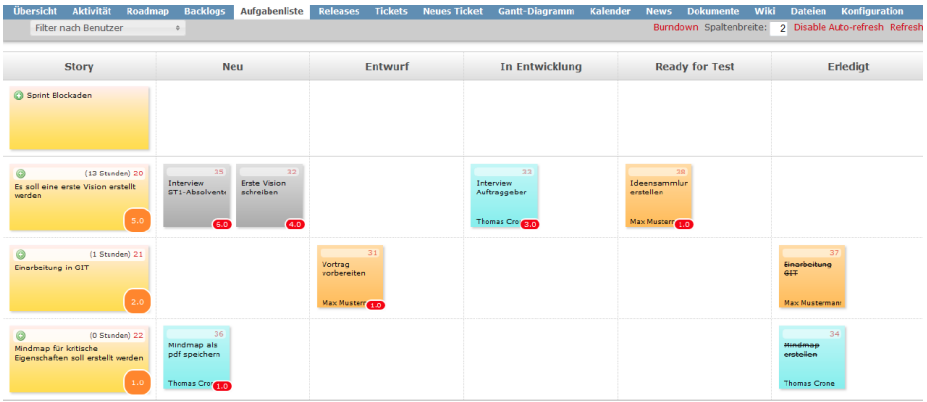


Abbildung A.38: Redmine-Tool (aus [20])

Schlussfolgerung

Hinsichtlich der Methoden, die erläutert werden, sind Scrum und Kanban geeignete Projektmanagement-Methoden, die für MOPS IV verwendet werden können, da es mit Wasserfallmodelle schwierig ist, Rückschritte zu haben. Sowohl V-Modell als auch Spiralmodell sind angemessen für große Projekte. Mit Scrum und Kanban erhält ein Team ein besseres Verständnis vom Ziel des Projektes, es gibt mehr Sichtbarkeit über den Fortschritt und klar definierte Aufgaben und Ergebnisse.

Als Projektmanagement-Tools sind CodeBeamer und Jira geeignet, die beide Scrum- und Kanban-Methoden verwenden, aber es sind keine Open-Source Tools. Die Universität Oldenburg besitzt bereits die Lizenz für Jira und MOPS4 kann es als Projektmanagement-Tool verwenden.

A.7 Verwandte Projekte

Von Eike Hagena, vorgetragen am 27. Oktober 2015.

Das Projekt MOPS3 (Marine Observation Platform for Surfaces) befasst sich mit der Entwicklung eines autonomes Bootes, welches an der Carl von Ossietzky Universität Oldenburg von Studenten entwickelt wurde. Neben der MOPS3-Plattform gibt es weitere autonome Boote, die von verschiedenen Institutionen, Unternehmen oder anderen Hochschulen entwickelt werden. In diesem Abschnitt sollen diese Projekte etwas näher erläutert werden. Dabei gehe ich zuerst auf schon entwickelte autonome Unter- und Überwasserfahrzeuge ein, danach stelle ich noch zwei in der Nordsee befindliche Messstationen vor.

Autonome Wasserfahrzeuge

Als autonomes Wasserfahrzeug bezeichnet man Boote und Schiffe, die selbstständig, also ohne Crew, in der Lage sind Aufträge durchzuführen (vgl.[100]). Als erstes sollen zwei autonome Unterwasserfahrzeuge vorgestellt werden, danach noch zwei autonome Überwasserfahrzeuge.

HUGIN 1000

Der Hugin 1000, von der Firma Kongsberg Maritime GmbH, ist ein autonomes Unterwasserfahrzeug, was vor allem für Vermessungen und die Datenerfassung unter Wasser eingesetzt wird. Es kann beispielsweise den Meeresboden durch verschiedene Sensoren erfassen und so eine Hochauflösende Darstellung erstellen. Dabei kann es, wie zum Beispiel das MOPS3-Boot auch, auf zwei unterschiedlichen Arten gesteuert werden. Das U-Boot wird entweder durch einen Nutzer gesteuert oder es fährt voll autonom. Der Hugin 1000 kann bis zu einer Tiefe von 3000 Metern abtauchen und besitzt einen Wendekreis von 15 Metern. Die Batterie ermöglicht eine Laufzeit von 24 Stunden bei einer Geschwindigkeit von vier Knoten (vgl.[70]).

Abbildung A.39: Hugin 1000 (aus [70])

REMUS 100

Der REMUS 100 ist ein weiteres autonomes Unterwasserfahrzeug der Forma Kongsberg Maritime GmbH. Allerdings ist diese Version kompakter und Leichter als der HUGIN 1000. Im Gegensatz zum HUGIN 1000 kann der REMUS 100 nicht 1000 Meter tief tauchen, sondern nur bis zu 100 Meter tief. Durch die kompakte Bauweise ist dieses Unterwasserfahrzeug aber besonders für den Betrieb in Küstengebieten geeignet. Da der Remus 100 mit nur 37 kg zudem sehr leicht ist, kann das Unterwasserfahrzeug sehr leicht transportiert werden. Das System eignet sich dann beispielsweise für die Meeresforschung, Suche, Bergung und als Unterstützung bei der Fischerei (vgl. [71]). Die Marine nutzt das System beispielsweise für die Suche nach Seeminen sowie Unterwasserhindernissen. Das hat den Vorteil, dass die Marine viel größerer Gebiete absuchen kann und gleichzeitig das Risiko für die Minentaucher verringert (vgl. [106]).

Abbildung A.40: Remus 100 (aus [71])

ASV Roboat

Das autonome Segelboot ASV (Autonomous Surface Vehicle) Roboat wird seit 2007 von der INNOC (Österreichische Gesellschaft für innovative Computerwissenschaften) entwickelt. Es ist 3.75 m lang und wiegt inklusiver des entwickelten Steuer- und Energieversorgungssystems 300 kg. Das Boot ist in der Lage, die Ruderanlage und das Segel vollautonom zu kontrollieren. Dafür kommunizieren verschiedene, am Boot angebrachte, Sensoren. Die Sensoren kommunizieren mit einem am Bord befindlichen Linux-PC. Neben der Windenergie, wird das Boot zusätzlich durch Solarpaneele betrieben, welche eine Leistung von 285 Watt liefern. Dies kann auf der Abb. A.41 betrachtet werden. Außerdem liefert eine Direktmethanol-Brennstoffzelle Energie als Reserve. Als Kommunikationssystem bietet das Boot WLAN, UMTS/GPRS und Satellitenkommunikation. So wird sichergestellt, dass eine ständige Kommunikation zwischen dem Boot und der Entwickler stattfindet. Das Kommunikationssystem hilft außerdem, dass die Route des Bootes verfolgt werden kann, eine neue Route übertragen wird oder um Messdaten zu übermitteln [65]. Als Einsatzzweck ergeben sich verschiedene Möglichkeiten:

- Da das Boot energieautark ist, kann das Boot auf den Weltmeeren beispielsweise für die Vermessung oder die Ermittlung von Fischbeständen eingesetzt werden.
- Eine kostengünstige Frachtenbeförderung ist denkbar, da weder Treibstoff benötigt wird, noch ein Personaleinsatz erforderlich ist.
- Überwachung von entlegenen und gefährlichen Regionen oder auch die Versorgung von abgelegenen Inseln.
- Durch die Aufzeichnung der vielen Messwerte, kann das System als Segeltrainer fungieren [66].

Abbildung A.41: ASV Roboat (aus [65])

SSA Avalon

Die SSA (Students Sail Autonomously) Avalon wird von der ETH Zürich entwickelt und ist ein ähnliches Segelboot wie das ASV Roboat. Das Projekt, der Bau eines autonomen Segelbootes, wurde neben den Vorlesungen des Bachelor-Studiums durchgeführt. Das Boot besitzt ähnliche Spezifikationen wie die des ASV Roboats. Es besitzt zum Beispiel auch zusätzliche Solarzellen zum Antrieb, die Avalon ist mit 4 Metern aber etwas länger. Das Entwickelte Modul, welches sich auf dem Boot befindet, ist so konstruiert, dass es auf jedem Segelboot eingesetzt werden kann. Dort kann es beispielsweise den Segler bei bestimmten Manövern unterstützten oder auch selbstständig eine Rettungsmission einleiten, sollte ein Segler über Bord gegangen sein (vgl. [107]).

Stationäre Messplattformen

In diesem Abschnitt möchte ich zwei stationäre Messplattformen vorstellen. Beide Messplattformen befinden sich in der Nordsee und werden zum Beispiel für den Schutz des Wattenmeeres eingesetzt.

ICBM

Die Messstation „Spiekeroog“ wurde 2002 vom Institut der Chemie und Biologie des Meeres der Carl von Ossietzky Universität Oldenburg im Wattenmeer der Nordsee aufgestellt. Das Wattenmeer bildet als Biotop den Lebensraum vieler Vögel und Fische und wurde von der UNSCO zum Weltnaturerbe ernannt. Um in Zukunft dieses Gebiet besser zu schützen und zum Beispiel auf Extremereignisse, wie Sturmfluten oder dem Anstieg des Meeresspiegels, besser zu reagieren, wurde die Messstation errichtet. Durch die gemessenen Daten der Station ist es zum Beispiel möglich das Ökosystem besser zu verstehen (vgl. [63]). Die Station selbst ist 35 Meter lang und 10 Meter im Meeresgrund verankert. Die Energie für die Datentechnik bezieht die Station aus einer Windkraftanlage, sowie an dem Container angebrachte Solarzellen. Gemessen wird neben meteorologischen Werten, wie Luftdruck, Temperatur, Wind und Luftfeuchtigkeit, auch hydrographische Werte (vgl. [95]).

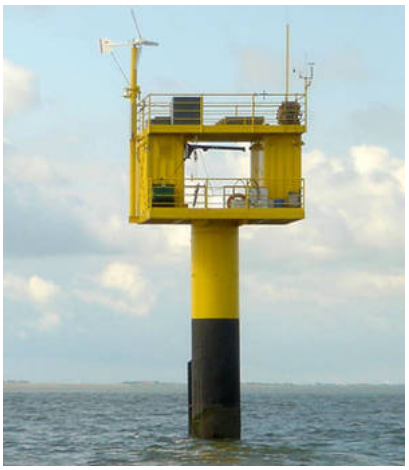


Abbildung A.42: Messstation „Spiekeroog“ (aus [63])

FINO1

Die Forschungsplattform FINO1 wurde 2003 in der Nordsee errichtet, um die Auswirkungen von Offshore-Anlagen auf die Umwelt zu untersuchen. Darüber

hinaus wird sie genutzt, um physikalische, chemische, biologische und hydrologischen Forschungsprojekte umzusetzen (vgl. [56]). Da in Zukunft immer mehr Offshore-Windkraftanlagen errichtet werden, wurde von der Bundesregierung beschlossen, Forschungsplattformen zu errichten, die sich in unmittelbarer Nähe befinden. Auf der Plattform wird unter anderem die Windstärke, Windrichtung und die Stärke der Meeresströmungen gemessen. Mit den ermittelten Ergebnissen möchte man den Bau von Windkraftanlagen optimieren und herausfinden, ob und wie sich umliegende Biotope durch den Bau von Windparks verändern. Deswegen findet auf der Plattform eine ökologische Begleitforschung statt, die beispielsweise das Vorkommen der Schweinswale untersucht (vgl. [55]).

A.8 Alternative Energiequellen

Von Eike Hagena, vorgetragen am 27. Oktober 2015.

Hier möchte ich einen kleinen Überblick darüber geben, welche alternativen Energiequellen es in der Schifffahrt gibt. Dabei konzentriere ich mich vor allem auf regenerative Energien wie Windkraft, Wasserkraft und Sonnenkraft

Sonnenpaneele

Solarmodule können, je nachdem wie viele Module angeschlossen werden und wie die Größenordnung des Energiebedarfs ist, als alleinige oder zusätzliche Energiequelle genutzt werden. Dabei unterscheidet man grundsätzlich zwischen drei Modularten, den Semi-flexiblen- Hochflexiblen- und Hochleistungsmodulen. Semi-flexible Solarmodule bieten sich besonders für seegehende Boote an. Diese Module sind bis zu einem bestimmten Grad biegbar und durch die verwendeten Materialien (Edelstahl und spezielle Kunststoffe) seewasserfest. Hochflexible Solarmodule eignen sich vor allem dann, wenn das Schiff im Hafen oder vor Anker liegt. Voraussetzung für ein solches Modul, ist eine Glatte Oberfläche, auf der das Modul aufgestellt wird. Hochflexible Module sind trittfest und können ganz einfach nach dem Ablegen wieder verstaut werden. Die Hochleistungsmodule werden fest an das Schiff angebracht. Sie sind nicht biegbar, bieten aber das beste Preis-/Leistungsverhältnis. Das liegt daran, dass diese Module eigentlich für die Versorgung von Häuser genutzt werden. Daher sind die Module, bedingt durch ihre Abmessungen, nur für Yachten mit einer entsprechenden Größe gedacht. Solarmodule haben den Vorteil, dass sie einfach und kostengünstig durch Selbstmontage anzubringen sind und nach der Installation keine weitere kosten entstehen. Außerdem besitzen Solarmodule eine lange Lebensdauer, ca. 20 Jahre, und sind Modular erweiterbar (vgl. [101]).

Windgeneratoren

Mittlerweile gibt es viele Windgeneratoren die für den Einsatz auf hoher See gedacht sind. Das liegt überwiegend daran, dass der Energiehunger immer mehr auf Schiffen steigt. Um diesen Energiehunger, beispielsweise auch nachts stillen zu können, machen kleine Windgeneratoren Sinn. Die großen Vorteile von Windgeneratoren sind, dass nach der Anschaffung jahrelang keine weiteren Kosten auf einem zukommen und gegenüber Solarpaneele weniger Platz beansprucht wird. Außerdem kann ein Windgenerator auch bei Stillstand des Bootes, zum Beispiel wenn es vor Anker liegt, die Batterie des Elektromotors aufladen. Je nachdem wie groß der Energieverbrauch am Bord eines Bootes ist, die Stärke des Windes und der Größe der Rotorfläche, kann ein kleiner Windgenerator bis zu 100 Ah pro Tag erzeugen (vgl. [102]).

Flettner-Rotor

Der Flettner-Rotor wurde bereits vor ca. 90 Jahren von Anton Flettner entwickelt. Der Rotor funktioniert im Prinzip wie eine Art Windsegel. Dafür muss ein auf dem Schiff befindlicher Zylinder elektrisch in Rotation versetzt werden. Bläst nun Wind gegen den Zylinder, wird der Wind auf der Seite beschleunigt, wo der Drehsinn des Zylinders mit der Windrichtung übereinstimmt. Auf der anderen Seite des Zylinders, wird der Wind dagegen abgebremst und strömt langsamer. Dadurch entsteht ein Druckunterschied. Die schnelle Seite erzeugt einen Unterdruck, die langsame Seite erzeugt einen Überdruck. Durch diesen Druckunterschied entsteht eine Kraft, die sich quer zur Windströmung verhält. Hierfür kann die Abb. A.43 betrachtet werden.

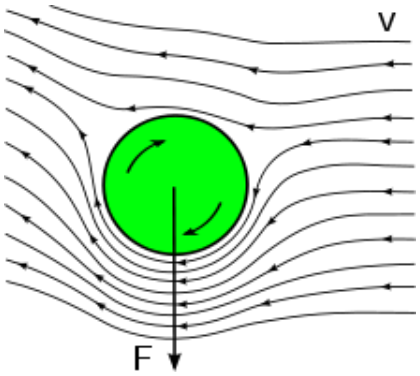


Abbildung A.43: Funktionsweise des Flettner-Rotors (aus [73])

In den letzten Jahren wurde dieses Prinzip von der Firma Enercon mit dem „E-Ship 1“ wieder aufgegriffen. Das Schiff ist 130 Meter lang und wird von der Firma als Frachtschiff für Windenergieanlagen-Komponenten einge-

setzt. Laut Unternehmen spart das E-Ship rund 15 % Treibstoff durch die Unterstützung des Flettner-Rotors [73].

Abbildung A.44: E-Ship der Firma Enercon (aus [73])

Wellenkraft

Der Antrieb eines Schiffes durch die Kraft der Wellen, ist ein Prinzip, welches aufgrund der Motorisierung in Vergessenheit geraten ist. Schon 1850 wurden Boote entwickelt, die nur durch die Kraft der Wellen angetrieben wurden. Besonders in Japan arbeiten Forscher nun wieder daran, die Wellenströmung für Schiffe auszunutzen. Das Prinzip ist sehr simpel. Eine Konstruktion am Bug mit zwei Flügelblättern wandelt die Kraft, die durch die auf und ab bewegenden Wellen entsteht, in Antriebswellen um. Um sich die Konstruktion besser vorstellen zu können, kann die Abb. A.45 betrachtet werden.

Eine solche Konstruktion wurde 2008 erfolgreich durch den Japaner Kenichi Horie mit der *Suntory Mermaid II* umgesetzt. Das Schiff, das allein durch die Wellenkraft angetrieben wurde, legte eine ca. 7000 Kilometer lange Strecke in drei Monaten zurück. Der Wellenantrieb ist deswegen ähnlich wie der Flettner-Rotor eher als Ergänzung gedacht. Mit einer entsprechenden Konstruktion lässt sich er Kraftstoffverbrauch um bis zu 15 % verringern (vgl. [61]).

Brennstoffzelle

Bei der Brennstoffzelle wird der Motor durch Wasserstoff betrieben. Besonders U-Boote werden durch diese Technologie angetrieben, da die Technologie unabhängig von der Außenluft funktioniert und zudem besonders leise ist. Die Brennstoffzelle selbst wandelt nach der Reaktion von Wasserstoff mit

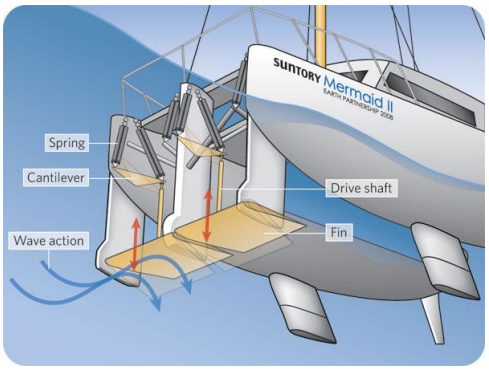


Abbildung A.45: Wellenkraft-Antrieb der *Suntory Mermaid II* (aus [103])

Sauerstoff, die freigewordene Energie in elektrische Energie um. Dadurch wird wiederum ein Elektromotor betrieben. Am Ende bleibt reiner Wasserdampf übrig. Nachteilig ist aber, dass die Herstellungskosten für Wasserstoffgas sehr hoch sind und das System einen hohen Energieverlust mit sich bringt (vgl. [60]).

Fazit und Ausblick

Neben den von mir vorgestellten alternativen Antrieben gibt es durchaus noch weitere. Das zeigt, wie viele Möglichkeiten wir theoretisch besitzen, um unser Boot anzutreiben. Meiner Meinung nach sind für uns besonders die Sonnenpaneele und der Windgenerator denkbar. Das liegt daran, dass sie einfach umzusetzen sind, sie kombinierbar sind und im weiteren Verlauf keine zusätzlichen Kosten auf uns zukommen. Auch könnte durch die größere Energieentstehung eventuell ein stärkerer Motor umgesetzt werden. Besonders spannend fand ich auch die Entwicklung des ASV Roboats, da dieses Projekt schon sehr weit fortgeschritten ist und sehr gut umgesetzt wurde. Die Österreichische Gesellschaft für innovative Computerwissenschaften konnte zum Beispiel mit dem Segelboot schon mehrere Preise gewinnen.

A.9 Kollisionserkennung und -verhütung

Von Björn Koopmann, vorgetragen am 3. November 2015.

Bei dem Befahren eines zunächst unbekannten Gewässers existieren zahlreiche Objekte, die aufgrund ihrer Größe, Beschaffenheit oder Eigenbewegung mögliche Kollisionsrisiken darstellen. Die Sicherheit unseres autonomen Wasserfahrzeugs vernachlässigend, stellen gleichzeitig auch wir eine potentielle

Gefahr für die Gesundheit und das Leben unserer Mitmenschen sowie das umgebende Ökosystem dar.

Um die zahlreichen Probleme bei der automatischen PfadRegelung zu beheben, wurde durch die vorherige Projektgruppe bereits ein einfaches System zur Verhütung von Kollisionen mit statischen Hindernissen implementiert. Die durchgeführten Berechnungen basieren dabei auf Daten, die im Rahmen der Missionsplanung über die Onshore-Software erhoben werden. Während die einzelnen Elemente der tatsächlichen Systemumgebung ein dynamisches Eigenverhalten aufweisen, erfasst das System seine Umgebung einmalig in einer vergleichsweise abstrakten Sichtweise und ist daher für die sichere Navigation in einem realen Einsatzszenario kaum geeignet.

Die Realisierung eines echtzeitfähigen Steuergeräts zur autonomen Pfadplanung fokussierend, ergeben sich in diesem Anwendungsfeld auch für die neue Projektgruppe zahlreiche Herausforderungen, für dessen Lösung eine schwerpunktbezogene Einarbeitung in die zu Grunde liegende Thematik unerlässlich ist. Neben den aktuell handhabbaren Hindernissen sollen hierzu auch dynamische Hindernisse – also Objekte mit einer eigenen Fahrtrichtung und -geschwindigkeit – in die Betrachtungen einbezogen werden.

Abbildung A.46: Statische und dynamische Hindernisse

Das Ziel dieser Ausarbeitung ist es, die wesentlichen Verfahren und Methoden zur Erkennung, Identifikation und Verfolgung von Objekten, der Prädiktion und Verhütung von Kollisionen sowie möglichen Handlungsalternativen zur Modifikation und Optimierung des bisherigen Ansatzes vorzustellen. Aufgrund des begrenzten Umfangs dieser Arbeit und der hohen Komplexität des betrachteten Anwendungsgebiets werden einige Ausführungen auf die für unsere Projektarbeit relevanten Aspekte beschränkt.

Aktuell eingesetztes System

Wie bereits in der Ausarbeitung „*Software des bestehenden Systems*“ in Kapitel A.1 beschrieben, gliedert sich die Software in zwei wesentliche Bestandteile. Neben einer Onboard-Software, die für die Steuerung des Bootes und die Regelung der Motoren zuständig ist, wird zur Konfiguration und Überwachung sowie zur Planung der auszuführenden Missionen ein Onshore-Tool mit einer graphischen Benutzeroberfläche eingesetzt.

Vor dem Beginn jeder Mission erhält der Benutzer die Möglichkeit, einzelne Wegpunkte in einem der tatsächlichen Umgebung entsprechenden Kartenausschnitt zu markieren. Um eine Beschränkung des Expeditionsraumes vorzunehmen, können zudem Gefahrenbereiche oder auszulassende Gebiete in Form von Polygonen eingezeichnet werden. Die über *OpenSeaMap* und eine *JXMapViewer2*-Schnittstelle bereitgestellten Daten ermöglichen somit die manuelle Berücksichtigung statischer Hindernisse.

Nach der Übertragung der entgegengenommenen Missionsdaten und dem Starten des Systems kann die Bahnplanung erfolgen. Im Verlauf der Ausführung wird ein kollisionsfreier Pfad von der Start- zur Zielposition des aktuellen Missionsabschnitts durch die *PathPlanner*-Komponente bestimmt. Vorbild für diese Berechnungen ist der gewichtete *A*-Algorithmus ARA** [76], der ein effizientes Verfahren zur Suche des kürzesten Weges in einem ungerichteten, kantengewichteten Graphen darstellt. Zur Beschreibung der Funktionalität wird das in Abb. A.47 dargestellte Hindernisfeld verwendet.

Abbildung A.47: Beispielhaftes Hindernisfeld (aus [32, S. 19])

Vor der Ausführung der Pfadsuche wird ein als Adjazenzliste implementierter Graph erzeugt, der alle möglichen Pfade repräsentiert. Mithilfe spezieller Methoden kann anschließend der Kern des Graphens extrahiert und in dem vorliegenden Kollisionsgraphen gekennzeichnet werden. Der Informationsgehalt wird dabei auf die relevanten Positionsinformationen der vorliegenden Hindernisse beschränkt.

In der darauffolgenden Kompositionsphase können nun zwischen allen paarweise verschiedenen Knoten neue Kanten erzeugt werden, solange sie keines der Hindernisse schneiden. Ein Entlangfahren des Bootes an dem Rand eines Polygons ermöglichend, wird das Berühren eines Pfades dabei nicht als Schneiden des Hindernisses gewertet. Um die spätere Auswertung der hinterlegten Informationen zu vereinfachen, erfolgt die Gewichtung der Kanten mit der Länge des Weges, den sie repräsentieren (vgl. [31, S. 72]).

Nach dem Hinzufügen der Start- und Zielpositionen des aktuellen Missionsabschnitts lässt sich der gesuchte Pfad anhand der Gewichtung der Kan-

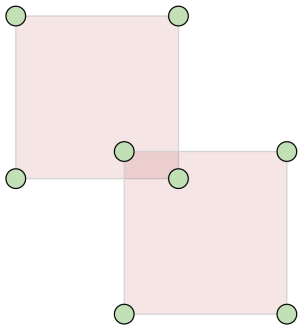


Abbildung A.48: Kennzeichnung der Polygonpunkte (aus [32, S. 19])

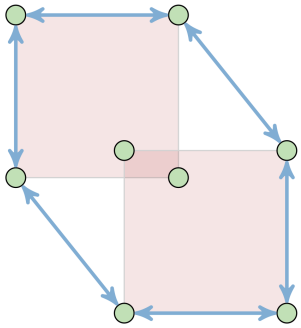


Abbildung A.49: Darstellung als ungerichteter Graph (aus [32, S. 19])

ten ablesen. Da durch dieses Verfahren stets der kürzeste Pfad ausgegeben wird, ergibt sich der Kandidat über die minimale Summe an einbezogenen Gewichten. Das Ergebnis des Algorithmus für das in Abb. A.47 beschriebene Hindernisfeld ist in Abb. A.50 dargestellt.

Die vorliegende Methode löst das Pfadproblem über eine Reduktion der Komplexität auf nur wenige miteinander verbundene Kurven im Raum und stellt damit ein klassisches *Wegkartenverfahren* dar. Um die Vergleichbarkeit mit den in Abschnitt A.9 vorgestellten alternativen Ansätzen zu erhöhen, ist in Abb. A.51 ein weiterer *Sichbarkeitsgraph* dargestellt, der dem Ergebnis einer Berechnung mithilfe des zuvor beschriebenen Verfahrens entspricht.

Während die drei grau markierten Polygone zuvor erkannte Hindernisse visualisieren, entspricht die rot gestrichelte Linie dem gesuchten kollisionsfreien Pfad. Auch in dieser Abbildung werden die Nachteile des bisher eingesetzten Verfahrens deutlich: Während ausschließlich statische Hindernisse berücksichtigt werden, ist ein Umfahren der detektierten Hindernisse entlang des berechneten Pfades nur ohne einen in der Realität unverzichtbaren Sicherheitsabstand möglich.

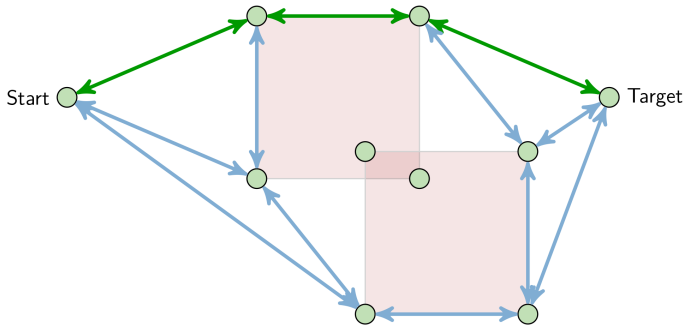


Abbildung A.50: Hinzufügen der Start- und Zielpositionen (aus [32, S. 19])

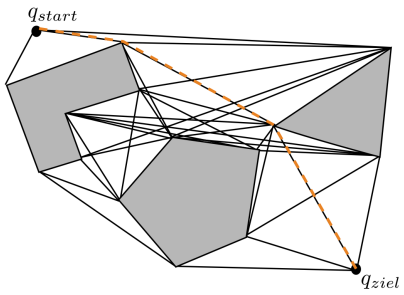


Abbildung A.51: Sichtbarkeitsgraph (aus [47, S. 14])

Klassifikation von Hindernissen

Um die nachfolgenden Ausführungen zur Definition eines Prozesses zur Kollisionsverhütung zu vereinfachen, ist es zunächst sinnvoll, die zu erkennenden Hindernisse anhand geeigneter Bewertungskriterien in Gruppen einzuteilen. Die Entfernung eines Hindernisses vom Bug unseres Boots bezeichnen wir hierzu im Folgenden mit der Variable d – den technologiebedingten Mindestabstand zwischen einem möglicherweise eingesetzten Radargerät sowie einem gerade noch detektierbaren Objekt mit der Konstante R.

Die erste Gruppe bilden Hindernisse, die – unter Verwendung einer geeigneten Sensorsuite – im sogenannten *Fernbereich* ($d \geq R$) zu verorten sind. In diese Klasse fallen vor allem großen Segel- und Motorboote, die bereits lange im Voraus erkannt werden können. Sie stellen das geringste Kollisionsrisiko dar und lassen sich aufgrund ihrer bekannten Verhaltensparameter (Geschwindigkeit, Kurs) mit einem großen Sicherheitsabstand umfahren. Mithilfe eines geeigneten Radar- oder AIS-Systems können die relevanten Sensordaten in digitaler Form empfangen und vergleichsweise einfach weiterverarbeitet werden.

Abbildung A.52: Klassifikation von Hindernissen

Die zweite Klasse umfasst Schwimmer, Tiere, Treibgut und Seezeichen – also jene Objekte, die aufgrund ihrer geringen Größe und der damit verkleinerten Signatur nicht über ein herkömmliches Radarsystem erfasst werden können. Sie treten vorrangig im *Nahbereich* des Bootes ($d < R$) auf und stellen somit ein kritisches Kollisionsrisiko dar, zu dessen Vermeidung eine schnelle Sensorauswertung und kurze Antwortzeiten erforderlich sind.

In der dritten und letzten Äquivalenzklasse sind Hindernisse *unter der Wasseroberfläche* zusammengefasst. Sie stellen aufgrund des verringerten Tiefgangs eine erhöhte Gefahr für die Antriebseinheit unseres Wasserfahrzeugs dar und werden mithilfe der bisher bekannten Sensoren in der Regel nicht hinreichend genau erkannt. Für die ausreichende Abdeckung dieses Anwendungsfalls wäre der ergänzende Einsatz eines Sonargeräts denkbar.

Verfahren zur Kollisionserkennung

In dem von uns betrachteten Anwendungsfeld besteht die besondere Herausforderung bei der Umsetzung einer effizienten und zuverlässigen Kollisionserkennung darin, aus dem Wissen über die aktuelle Position, der Fahrtrichtung und der Position sowie der Bewegungsrichtung und der Geschwindigkeit möglicher Hindernisse eine Abschätzung der Kollisionsrisiken durchzuführen und geeignete Gegenreaktionen abzuleiten. Mithilfe des folgenden konzeptionellen (Online-)Workflows lässt sich dieses Verhalten unter Einbezug der benötigten Sensordaten vergleichsweise einfach realisieren:

1. Erkennen von Kollisionsrisiken
2. Verfolgen detektierter Hindernisse über die Zeit
3. Abschätzen der Positionen für zukünftige Zeitpunkte

4. Einleiten von Gegenmaßnahmen
   (a) Verringern der Geschwindigkeit
   (b) Einleiten von Ausweichmanövern
   (c) Einsatz von Signal- und Warnkörpern
5. Iterative Neubewertung der Risikosituation

Unter Annahme einer unbegrenzt hohen Sensorauflösung sowie der Verfügbarkeit unbeschränkter Rechenleistung ergeben die zuvor beschriebenen Arbeitsschritte eine vierdimensionale Karte (x, y, z, Zeit t), die für jeden Punkt des relevanten Konfigurationsraums eine exakte Aussage über die bestehenden Kollisionsrisiken zulässt. Die weniger optimalen Voraussetzungen einer realen Anwendung vernachlässigend, lassen sich auch hierbei die Kollisionsrisiken mit einer gewissen Wahrscheinlichkeit abschätzen.

Da sich unser autonomes Wasserfahrzeug in der Regel über der Wasseroberfläche befindet und die absolute Höhe nur geringen Schwankungen unterliegt, ist die Verwendung von 2D-Ortsinformationen ausreichend. In Bezug auf die zeitliche Modellierung genügt es weiterhin, nur auf eine beschränkte Folge von diskreten Zeitpunkten zurückzugreifen. Die Umsetzung des zuvor beschriebenen Workflows forcierend, ist die Realisierung der drei nachfolgenden Schritte erforderlich.

Computer Vision

Der Begriff der *Computer Vision* bezeichnet die Fähigkeit des „maschinellen Sehens“ und fasst Verfahren und Methoden zusammen, die eine rechnerbasierte Abbildung des menschlichen visuellen Systems anstreben. Neben dem Erkennen von Objekten in verschiedenartigen Sensordaten sowie dem Vermessen und Klassifizieren detektierter Objekte, beinhaltet sie auch das Treffen von Entscheidungen auf Grundlage der errechneten Informationen.

Bezogen auf unser autonomes Wasserfahrzeug beinhaltet diese Aktivität das Erkennen von Hindernissen sowie das Bestimmen ausgewählter objektspezifischer Parameter, wie beispielsweise seiner Größe oder der Distanz zum Aufnahmepunkt. Bei der Umsetzung einer effektiven Kollisionserkennung können folgende *Computer Vision*-Algorithmen hilfreich sein:

- Farbklassifikatoren
- Hough-Transformation, Kontrastanalyse
- Sobel-Operator, Wavelets, Gauß-Laplace-Pyramide
- Modell-und Mustererkennung

Target Tracking

Mithilfe des sogenannten *Target Trackings* lässt sich das automatische Verfolgen von zuvor erfassten Objekten in Sequenzen von Sensordaten realisieren. Durch die iterative Wiedererkennung von Mustern und Modellen in den Eingangsdaten ergibt sich eine Reihe gedachter Trajektorien, die die einzelnen Hindernispositionen im Raum über der Zeit abbilden.

Das gewünschte Verhalten realisierend, werden die verschiedenen Algorithmen je nach Vorgehensweise in „Tracking-by-Detection"- und „Detection-by-Tracking"-Verfahren klassifiziert. Während die Ansätze der ersten Gruppe vor allem auf dem eingangs beschriebenen Ablauf beruhen, kann die Effizienz der Verfahren aus der zweiten Klasse durch die Extraktion heuristischer Bewegungsprofile und der anschließenden Detektion in den so verkleinerten Bildbereichen weiter gesteigert werden.

In Hinblick auf den konkreten Anwendungsfall ist das Verfolgen von Objekten ein sinnvolles Hilfsmittel, um weitere objektspezifische Parameter, wie beispielsweise die Ausrichtung, den Kurs oder die Geschwindigkeit eines sich bewegenden Hindernisses zu bestimmen. Als mögliche Prototypen für die Implementierung eines effizienten *Target Tracking*-Algorithmus bieten sich hierfür insbesondere die folgenden Verfahren an:

- $\alpha/\beta/\gamma$-Filter
- Kalman-Filter
- Sequentielle Monte-Carlo-Methode

Interpolation des zukünftigen Verhaltens

Auf Grundlage der zuvor gesammelten Informationen können im Anschluss durch die *Interpolation des zukünftigen Verhaltens* wertvolle Informationen über die zu erwarteten Positionen der erkannten Hindernissen und den damit verbundenen Kollisionsrisiken gewonnen werden.

Die Positionen der detektierten Hindernisse werden dazu mit dem Voranschreiten der globalen Zeit zunächst als Punkte in einer zweidimensionalen Karte notiert. Unter Anwendung der nachfolgend vorgestellten Ausgleichsrechnungen ergeben sich daraus – wenn auch nur probabilistisch – mögliche zukünftige Positionen der erkannten Hindernisse.

Regression Unter einer *Regression* wird eine Untersuchung des Zusammenhangs zwischen mehreren Datenpunkten verstanden. Die resultierende Funktion enthält dabei keine Unsicherheiten oder Messfehler – sie werden als konstant angenommen und aus den Ergebnissen herausgerechnet.

Wie auch bei allen anderen Verfahren der Ausgleichsrechnung unterliegen die Koeffizienten der Stützstellen einem stochastischem Modell, das die Eigenschaften der realen Umgebung abstrahiert.

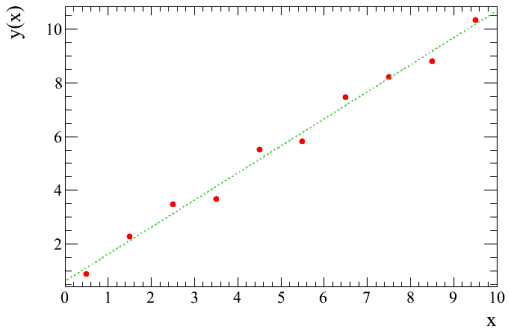


Abbildung A.53: Beispielhaftes Ergebnis einer Regression (aus [118])

Fitting Ein *Fitting* ist eine Funktionsanpassung, die die Koeffizienten eines Polynoms festen Grades bestimmt. Das Ergebnis ist im Gegensatz zu dem einer Regression mit einer Unsicherheit versehen, sodass das Resultat einer Kurvenschar entspricht, in der der tatsächliche funktionale Zusammenhang mit einer gewissen Wahrscheinlichkeit enthalten ist.

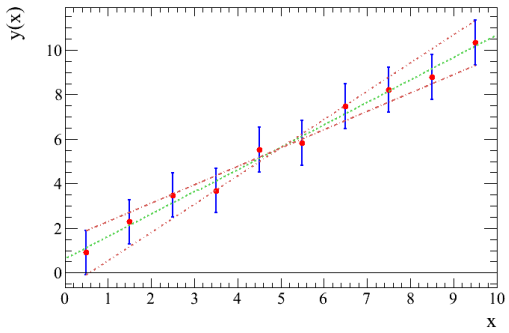


Abbildung A.54: Beispielhaftes Ergebnis eines Fittings (aus [118])

Alternative Ansätze

Neben dem aktuell realisierten Verfahren zur Bestimmung eines kollisionsfreien Pfades durch ein Hindernisfeld existieren weitere Ansätze, die sich direkt aus Modifikationen der bekannten Vorgehensweise oder indirekt aus den charakteristischen Eigenschaften der Systemumgebung ableiten lassen.

Die Gliederung des nachfolgenden Abschnitts orientiert sich im Wesentlichen an der Bachelorarbeit *„Pfadplanung mit harmonischen Potentialfeldern“* von Johannes Diemke [47]. Anhand des beispielhaften Hindernisfelds, das

bereits in Kapitel A.9 vorgestellt wurde, können auf den folgenden Seiten die möglichen Vorgehensweisen betrachtet und ihre Vor- und Nachteile herausgearbeitet werden.

Wegkartenverfahren

Die erste Klasse der im Rahmen dieser Ausarbeitung behandelten Ansätze bilden die sogenannten *Wegkartenverfahren*. Sie stellen die Verbindungen zwischen den Wegpunkten im Konfigurationsraum durch ein Netz von Kurven dar. Durch die Reduktion der Komplexität lässt sich die Suche nach einem geeigneten Pfad als Graphensuche implementieren und mithilfe bekannter Verfahren, wie beispielsweise dem A*-Algorithmus, lösen (vgl. [47, S. 13]).

Voronoi-Diagramme Ein *Voronoi-Diagramm* (auch: Dirichlet-Zerlegung) ist die Zerlegung eines gegebenen Raumes in Regionen. Jede enthaltene Region wird dabei durch genau ein Zentrum charakterisiert und umfasst alle Punkte, die in Bezug zur euklidischen Metrik

$$d(x, y) = \|x - y\|_2 = \sqrt{(x_1 - y_1)^2 + ... + (x_n - y_n)^2}$$

näher an dem Zentrum der Region liegen als an jedem anderen Zentrum. Die Grenzen zwischen den Voronoi-Regionen bilden die Menge der möglichen Pfade durch das Hindernisfeld. Da der euklidische Abstand im zweidimensionalen Raum über den Satz des Pythagoras bestimmt werden kann und der Pseudocode des Verfahrens frei verfügbar ist, lässt sich dieses Verfahren besonders einfach implementieren.

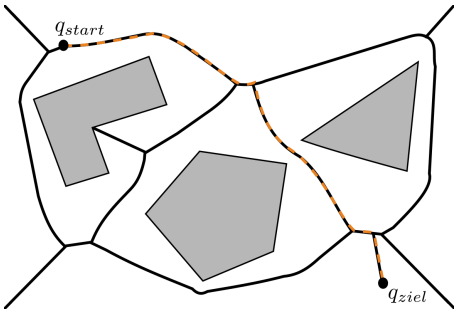


Abbildung A.55: Voronoi-Diagramm (aus [47, S. 14])

Probabilistische Wegkarten Die Erzeugung einer *probabilistischen Wegkarte* ist ein heuristisches Verfahren zur Suche eines freien Pfades, das sich vor allem für hochdimensionale Konfigurationsräume eignet. Durch das zufällige

Hinzufügen von Konfigurationen und dem anschließenden Versuch, diese miteinander zu verbinden, wird schrittweise ein kollisionsfreier Pfad durch das Hindernisfeld generiert (vgl. [47, S. 13]).

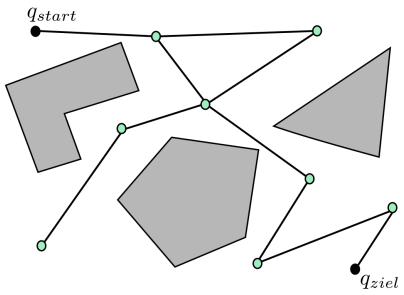


Abbildung A.56: Probabilistische Wegkarte (aus [47, S. 14])

Zellaufteilungsverfahren

Eine weitere Klasse von Algorithmen stellen die sogenannten *Zellaufteilungsverfahren* dar. Sie bedienen sich klassischer Devide-and-Conquer-Prinzipien und zerlegen den Konfigurationsraum anhand ausgewählter Merkmale in unterschiedlich große Abschnitte, die über einen Graphen miteinander verbunden werden. Die Suche nach einem geeigneten Pfad reduziert sich somit auf die Suche nach einer Sequenz von freien miteinander verbundenen Zellen.

Trapezförmige Zellaufteilung Die *trapezförmige Zellaufteilung* ist ein Zellaufteilungsverfahren, dass den Konfigurationsraum durch das vertikale Verbinden der Eckpunkte sämtlicher Hinderniszonen mit den Grenzen des Sichtfeldes segmentiert. Die im vorherigen Abschnitt beschriebene Vorgehensweise anwendend, kann anschließend über die Suche einer geeigneten Sequenz von Zellen der gewünschte Pfad extrahiert werden.

Nachbarschaftsrelationen Eine Erweiterung des soeben vorgestellten Verfahrens lässt sich direkt aus einer genaueren Betrachtung der *Nachbarschaftsrelation der trapezförmigen Zellaufteilung* herleiten. Durch eine Verbindung der Zentren benachbarter Zellen ergibt sich ein Graph, der alle möglichen Pfade über dem gegebenen Konfigurationsraum abbildet. Im Anschluss an die Erzeugung des Graphens kann ein geeigneter Pfad ausgewählt oder die Ergebnisse für eine wiederholte Verwendung abgespeichert werden.

Approximierende Zellaufteilung Das Verfahren der *approximierenden Zellaufteilung* ist eine der naheliegendsten Methoden zur Suche eines freien

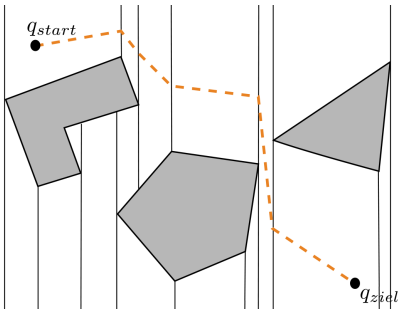


Abbildung A.57: Trapezförmige Zellaufteilung (aus [47, S. 14])

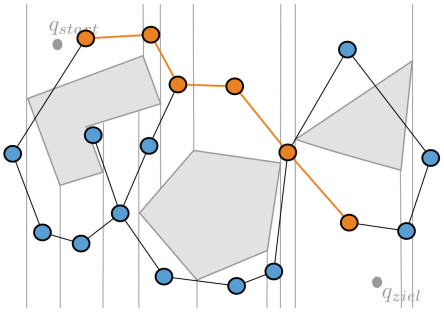


Abbildung A.58: Nachbarschaftsrelation der trapezförmigen Zellaufteilung (aus [47, S. 14])

Pfades über einem gegebenen Hindernisfeld. Der freie Konfigurationsraum wird hierbei zunächst mittels eines gleichmäßigen Gitters in eine endliche Anzahl gleich großer Zellen eingeteilt. Mithilfe bekannter Heuristiken und Backtracking-Verfahren lässt sich anschließend ein Pfad extrahieren, der den spezifizierten Suchkriterien am ehesten entspricht.

Potentialfeldmethoden

Die letzte Klasse der in dieser Ausarbeitung betrachteten Ansätze bilden die sogenannten *Potentialfeldmethoden*. Sie definieren ein differenzierbares Potentialfeld über dem freien Konfigurationsraum, in das die Eigenschaften der Hindernisse sowie der Start- und Zielpunkte abgebildet werden.

Während die Ausgangsposition q_{start} dem globalen Maximum – also einem Potential mit einer stark abstoßenden Kraftwirkung – und die Zielposition q_{ziel} einem globalen Minimum entsprechen, erfolgt die Bewegung des zu steuernden Körpers stets entlang der Potentiale. Aufgrund der ebenfalls abstoßenden Wirkung der Hindernisregionen gleitet der Agent sanft durch

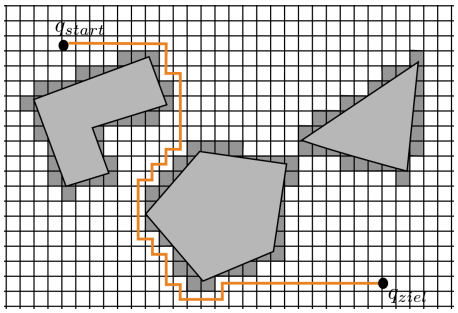


Abbildung A.59: Approximierende Zellaufteilung (aus [47, S. 14])

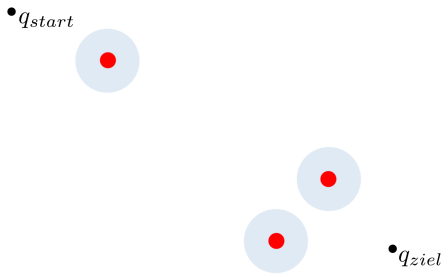


Abbildung A.60: Gegebenes Hindernisfeld (aus [47, S. 15])

das eingangs definierte Potentialfeld. Im Gegensatz zu den vorherigen Methoden wird der freie Pfad bei diesem Verfahren also dynamisch, d.h. in einer Simulationsumgebung oder zur Laufzeit des realen Systems, generiert.

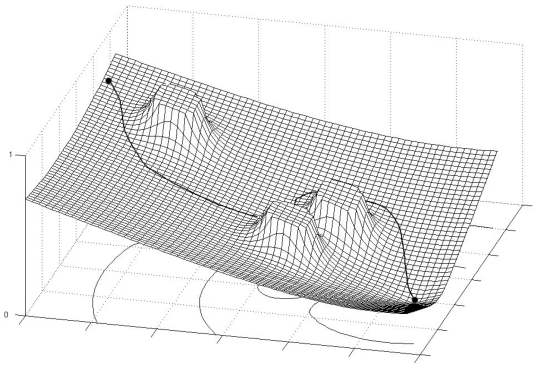


Abbildung A.61: Resultierendes Potentialfeld (aus [47, S. 15])

Hilfsmittel und Tools

Unabhängig von dem gewählten Verfahren zur Bestimmung eines kollisionsfreien Pfades bedarf es geeigneter Algorithmen und Methoden, die eine schnelle und robuste Auswertung der empfangenen Sensordaten ermöglichen. Die folgenden Hilfsmittel können die Implementierung der gesuchten Signalverarbeitungskette vereinfachen und das Finden schneller Problemlösungen ermöglichen.

OpenCV-Bibliothek

Die OpenCV-Bibliothek ist eine plattformunabhängige Sammlung von Algorithmen für das „maschinelle Sehen“ – also der Implementierungen der in Abschnitt A.9 beschriebenen *Computer Vision* sowie des *Target Trackings* aus Kapitel A.9. Neben der vollautomatischen Klassifikation von zwei- und dreidimensionalen Objekten, der Eigenbewegungsschätzung aus einer Reihe von Sensordaten und verschiedenen Realisierungen des optischen Trackings sowie des Kalman-Filters sind auch Methoden zur Modell- und Mustererkennung enthalten. Die Bibliothek ist gut dokumentiert und wird durch zahlreiche Beispiele vervollständigt.

Universitäre Einrichtungen

Als Ergänzung zu den in dieser Ausarbeitung genannten Quellen kann es sich im Verlauf der Bearbeitungsphase als sinnvoll erweisen, bereits bestehendes Know-how anderer Abteilungen in die laufenden Entwicklungen einzubeziehen. Die Mitarbeiter der *Abteilung für Mikrorobotik und Regelungstechnik* haben langjährige Erfahrung mit der FPGA-basierten Bilderkennung und könnten als mögliche Ansprechpartner wertvolle Hinweise für die Umsetzung der automatischen Kollisionserkennung liefern. Auch die *Abteilung für Automatisierung und Messtechnik* unter Leitung von Prof. Dr. Andreas Hein befasst sich mit der rechnergestützten Auswertung von (medizinischen) Bild- und Sensordaten. In der Lehrveranstaltung *Signal- und Bildverarbeitung* (inf210) werden die Grundlagen und einige weiterführende Konzepte dieses Anwendungsbereichs vermittelt.

Fazit

Im Rahmen dieser Ausarbeitung konnten verschiedene Ansätze und Methoden zur Erkennung und Verhütung von Kollisionen erarbeitet und vorgestellt werden. Für die Implementierung einer effizienten Kollisionserkennung ist die Realisierung einer Computer Vision, die Umsetzung eines Target Trackings sowie die Bereitstellung von Methoden für die Interpolation des zukünftigen Verhaltens möglicher Hindernisse von zentraler Bedeutung. Das in Abschnitt

A.9 vorgestellte Verfahren zur Kollisionsverhütung gibt konkrete Handlungsanweisungen und könnte als Grundlage für die Konzeption einer entsprechenden Systemkomponente verwendet werden.

Bezogen auf unsere Projektgruppe würde sich der Einsatz einer Kombination von RADAR und AIS im Fernbereich, einem Kamerasystem zur Detektion von Objekten im Nahbereich sowie einem SONAR-System zur Erfassung von Unterwasser-Hindernissen besonders gut eignen. Aufgrund der hohen Komplexität dieser Systeme und der Vielfältigkeit des zu Grunde liegenden Entwurfsraums sind eine erweiterte Einarbeitung sowie die gruppeninterne Diskussion möglicher Lösungsansätze ausdrücklich erwünscht.

A.10 Umweltsensorik und Modelling and Control

Von Peter Tank, vorgetragen am 3. November 2015.

Diese Seminararbeit entstand im Rahmen der Projektgruppe Marine Observation Platform for Surfaces IV an der Universität Oldenburg im Masterstudiengang Eingebettete Systeme und Mikrorobotik.

Motivation

Um eine gute Einschätzung der biologischen und chemischen Vorgänge in Gewässern zu bekommen ist es notwendig an verschiedenen Stellen des zu überwachenden Gewässers Messungen vorzunehmen. Dieses bindet eine erhebliche Menge an Arbeitskraft seitens des wissenschaftlichen Personals. Daher gibt es Überlegungen diese Messungen automatisiert und autonom durchzuführen. Das Projekt MOPS hat deswegen das Ziel eine autonome Plattform zu entwickeln, mit der automatisiert Messungen in Gewässern durchgeführt werden können.

Ziel der Arbeit

Diese Seminararbeit soll für die Projektdurchführung wichtige Teilthemen beleuchten. Konkret geht es um die Betrachtung der Umweltsensorik und der Modellierung und Regelung. Es wird ein Einblick in die technischen Möglichkeiten gegeben und deren Funktionsweise diskutiert.

Aufbau der Arbeit

Nach der Einleitung wird zuerst die als Nutzlast bezeichnete Umweltsensorik und deren Hilfsmittel aufgezeigt. Anschließend wird die Modellierung und Regelung diskutiert. Abschließend wird ein Fazit gezogen und mögliche Arbeiten für das weitere Projektvorgehen aufgezeigt.

Nutzlast

Die Grundaufgabe des MOPS soll das Anfahren bestimmter Wegpunkte sein, an denen die sogenannte Nutzlast zum Einsatz kommt. Zur Nutzlast gehören die Umweltsensoren und zugehörige Seilwinde. In den vorherigen Projektgruppen MOPS1 und MOPS2 gab es eine Auswahl bezüglich der Sensorik. Dieses Kapitel befasst sich mit diesen.

Trübheitssensor

Der Trübheitssensor misst die Trübung durch Detektierung von Schwebeteichen in Flüssigkeiten. Von vorherigen Projektgruppen wurde der Sensor Nephelometric Turbidity der Firma Ponsel ausgewählt [31].

Abbildung A.62: Trübheitssensor (aus [2])

Technische Daten Folgend werden die für die Nutzung wichtigsten technischen Daten aufgezeigt.

Bussysteme	RS-485-Modbus und SDI-12
Versorgungsspg.	12 V
Datenspannung	5 V
Schutzklasse	IP68
Accuracy	5 %
Auflösung	0.01 bis 1 NTU

Tabelle A.1: Technische Daten NTU (aus [2])

Wie in Tab. A.1 gezeigt, ist der Betrieb über zwei Bussysteme möglich. Bisher ist der SDI-12-Bus in Betrieb. Der Grund hierfür ist die schnellere Implementierung. Die Nutzung der RS-485-Modbus-Kommunikation wäre störunempfindlicher, da die Kommunikation mittels differentieller Signale erfolgt.

Funktionsweise Das Grundprinzip der Messung ist ein optisches Verfahren. Es wird seitens einer Lichtquelle ein Lichtstrahl, meist im Infraroten Bereich, ausgesendet und an einer Lichtsenke wird die Intensität gemessen. Es ist davon auszugehen, dass durch Schwebeteilchen der Lichtstrahl abgeschwächt wird und daher ein Rückschluss auf die Trübung der Flüssigkeit geschlossen werden kann. Wird der Strahl von der Lichtquelle gemessen, handelt es sich um eine direkte Messung. Diese bietet sich an, wenn mit starken Trübungen zu rechnen ist. Sollte dieses Messverfahren bei schwachen Trübungen eingesetzt werden, ist damit zu rechnen dass das empfangene Signal sehr intensiv ausfallen kann. Ein Empfänger kann somit schnell übersteuert werden und das Messergebnis wäre unbrauchbar. Um dieses zu vermeiden, können zwei Maßnahmen ergriffen werden. Zuerst kann ein Empfänger mit einem höheren Eingangsbereich ausgewählt werden. Dieses zieht aber meist entweder höhere Kosten oder eine schlechtere Genauigkeit nach sich. Deswegen bietet sich die indirekte Messung an. Das Prinzip wird in Abb. A.63 gezeigt.

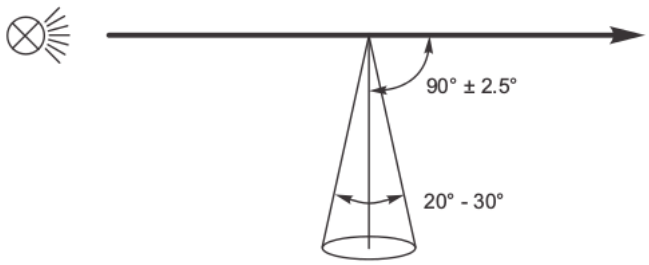


Abbildung A.63: Indirektes Messverfahren (aus [62])

Bei dieser Messmethode wird davon ausgegangen, dass die mit IR-Licht angestrahlten Teilchen nur einen kleinen Teil des Lichts absorbieren und den großen Teil reflektieren. Das reflektierte Licht wird dann in einem Rechten Winkel zur Hauptstrahlrichtung gemessen. Da es sich bei dieser Messmethode um eine Messung für wenige Schwebeteilchen handelt, ist das reflektierte Licht auch wesentlich geringer als die Lichtmenge bei der direkten Messung. Dadurch kann hierbei ein wesentlich effizienterer Empfänger verwendet werden.

Salzgehaltssensor

Der Salzgehaltssensor bestimmt die Leitfähigkeit des Wassers und schließt aus dieser Information auf den Salzgehalt.

Abbildung A.64: Salzgehaltsensor (aus [93])

Technische Daten Folgend werden die, für die Nutzung wichtigsten technischen Daten aufgezeigt.

Bussysteme	RS-485-Modbus und SDI-12
Versorgungsspg.	12 V
Datenspannung	5 V
Schutzklasse	IP68
Accuracy	1 %
Auflösung	0.01 bis 1 passend zum Messbereich

Tabelle A.2: Technische Daten C4E (aus [93])

Tabelle A.2 zeigt dass es hier zwei Bussysteme zur Auswahl gibt. Auch hier wurde wie beim Trübheitssensor der SDI-12 Bus implementiert. Aufgrund der höheren Störrobustheit wäre es zu empfehlen, auf die RS485-Modbus Kommunikation zurückzugreifen.

Funktionsweise Der Salzgehalt des Wassers wird über die Bestimmung des Leitwerts durchgeführt. Hierzu befinden sich Elektroden im zu vermessenden Medium. Zwei Elektroden aus Graphit bestimmen den Leitwert und zwei Elektroden aus Platin sind zur Referenzrechnung verbaut um u.a. Verschmutzungen auszugleichen [93].

Low-Power-Sleep
1200 Baud
12 V Devices
5 V Data
Half-Duplex
7 Databits and even Parity
up to 10 Slaves

Tabelle A.3: Technische Daten SDI-12

Interfaces

Die Interfaces dienen zur Kommunikation der Komponenten. Hier sind die Interfaces SDI-12 und RS-485-Modbus zu nennen.

SDI-12 Die SDI-12 Schnittstelle wurde entwickelt um bis zu zehn Slaves an einen Master anschließen zu können. Tabelle A.3 zeigt weitere Features.

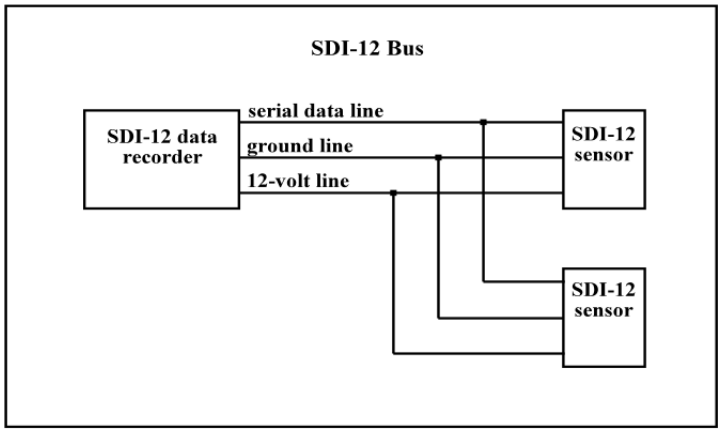


Abbildung A.65: SDI-12 Busprinzip (aus [58])

Abbildung A.65 zeigt den möglichen Netzaufbau des Busses. Über die serial data line wird ein Kommando vom Master (hier data recorder) an einen Slave (hier sensor) geschickt. Dieser erwacht aus dem Low Power Sleep und antwortet. Danach wird der Slave in den Low Power Sleep fallen. Über die 12 V Leitung werden alle Devices versorgt. Auf der serial data line werden die Daten mit einem 5 V Pegel übertragen.

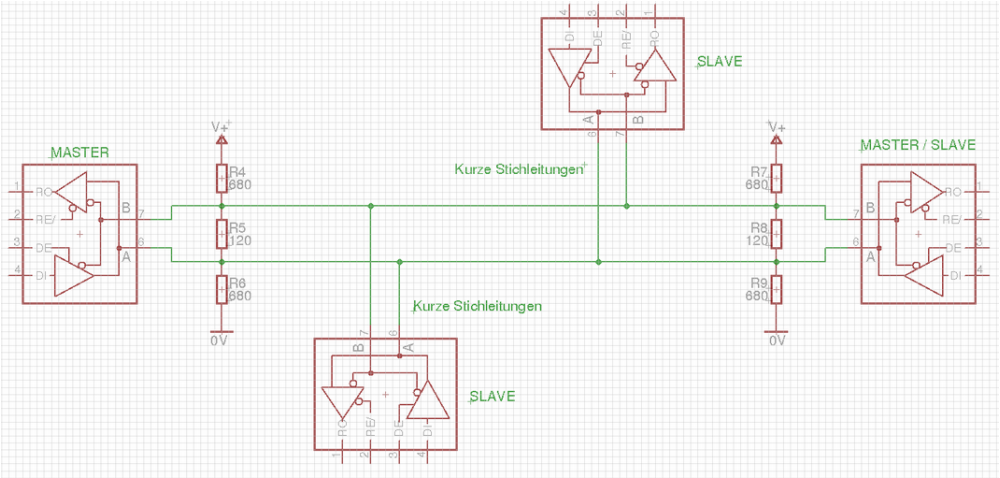


Abbildung A.66: Modbus Busprinzip (aus [57])

RS-485-Modbus Im Folgenden wird der RS-485 Standard mit aufgesetztem Modbus beschrieben.

RS-485 RS-485 ist ein Schnittstellenstandard bei dem Daten auf bis zu 1200 m differentiell mit bis zu 32 Slaves an einen Master angeschlossen werden können.

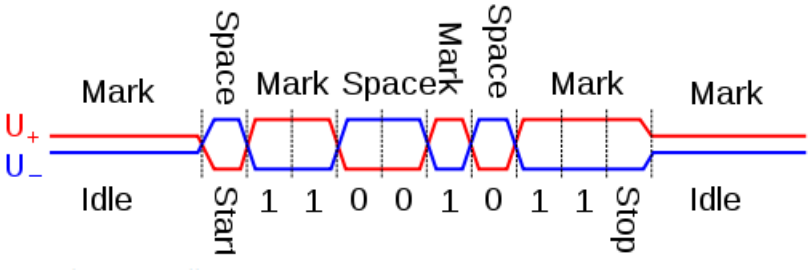


Abbildung A.67: Signale RS-485 (aus [64])

In Abb. A.66 sind an jeder Seite drei Widerstände eingesetzt. Der 120 Ohm Widerstand agiert als Terminator um Reflektionen auf der Leitung zu vermeiden. Die anderen beiden sind als Pull-Up bzw. Pull-Down gesetzt um ein sauberes Potenzial bieten zu können. Die beiden Leitungen halten jeweils einen invertierten Pegel um die differentielle Funktion darzustellen. Hierdurch wird eine höhere Störsicherheit gewährleistet. Abbildung A.67 zeigt den prinzipiellen Signalverlauf bei differentieller Übertragung. Sollte ein Signalweg ge-

stört werden, kann aus dem anderen Signalweg die Information entnommen werden. Sollten beide Signalwege gestört werden ist ein energetisch stärkerer Störimpuls nötig um den Signalhub beider Signale zu stören.

Modbus Protokoll Das Modbus ist ein Protokoll welches entweder auf der RS-485 Schnittstelle oder auf Ethernet aufgesetzt wird. Hierzu sind drei Modi einzustellen.

Modi	*Schnittstelle*
ASCII	RS-485
Remote Terminal Unit (RTU)	RS-485
TCP	Ethernet

Tabelle A.4: Modi Modbus (aus [52])

In Tab. A.4 wird gezeigt das die auf RS-485 basierenden Modi ASCII und RTU sind. Der ASCII-Modus überträgt ASCII-Zeichen die direkt auf dem Bus mitgelesen werden können. Dieses kann u.a. beim Debuggen sehr von nutzen sein. Nachteilig ist eine geringe Datendurchsatzrate, im Vergleich zu den anderen Modi, zu erwähnen. Der RTU-Modus überträgt Binärcode und hat eine höhere Datendurchsatzrate als der ASCII-Mode. Das Modbus Protokoll basiert auf Master-Anfragen und Slave-Antworten. Hierzu wird ein 16 Bit Register genutzt indem die Slaveadresse, ein Funktionscode und die Daten mit Errorcheck gesetzt werden [52, S. 8]. Wobei im ASCII-Mode die Message mit „:“ beginnt und endet mit einem Carrage Return. Während beim ASCII-Mode zwei Zeichen für eine Message gesendet werden, wird bei RTU ein Stream geschickt.

Anschlüsse

Um die Nutzlast mit der Auswertelogik zu verbinden wird das Standardkabel aus der maritimen Technik verwendet, dass M12 Kabel des NMEA2000 Standards.

Tabelle A.5 zeigt die Pinbelegung der NMEA-Verbindungen. Die NMEA-Verbindungen werden von der Sensorik zum Arduinosensorcape gelegt.

Nutzlast

Die Nutzlast ist das Werkzeug um an den Missionspunkten Messungen durchführen zu können.

Pin	*Ader*	*Funktion*
1	Shield	-
2	Rot	Pluspol
3	Schwarz	GND
4	Weiß	RS485
5	Blau	SDI-12/RS485

Tabelle A.5: Steckverbindung NMEA (aus [31, S. 95])

Sensorkäfig Der Sensorkäfig wird in [59] beschrieben. Die primären Aufgaben des Käfigs sind es die Sensorik aufzunehmen und diese vor mechanischer Belastung zu schützen. Der Käfig soll auch ein Instrument zur Tiefenbestimmung tragen. Ausgewählt wurde hierfür der PSA-916. Es wird durch eine Elektronikbox die informations- und energietechnische Anbindung sichergestellt. Durch eine Reihe von Feldtests müsste die korrekte und sichere Funktionsweise sichergestellt werden.

Seilwinde Die Seilwinde wird benötigt um den Sensorkäfig zu Wasser zu lassen, an Bord zu holen und auf die richtige Einsatztiefe zu bringen. Die Seilwinde ist über einen Auslegearm mit dem Sensorkäfig verbunden. Des Weiteren ist die Winde mit einem Kameramodul gekoppelt um deren Funktion zu überwachen. Laut [59, S. 26] sollte die Winde im Bugbereich des Bootes untergebracht werden. Es sollte zusätzlich eine Transportsicherung für den Sensorkäfig im Bugbereich des MOPS angebracht werden. Hierdurch kann sichergestellt werden dass auf der Fahrt zu einem Missionspunkt keine Schäden an Boot und Käfig auftreten.

Modeling and Control

Bei der Modellierung eines Prozesses oder eines Objektes geht es darum eine möglichst zweckmäßige mathematische Beschreibung zu erstellen. Durch diese können Vorgänge und Zusammenhänge berechnet und simuliert werden. Die Regelung setzt auf der mathematischen Beschreibung auf und versucht das Objekt oder den Prozess zu einem bestimmten Verhalten anzuregen und ihn dort stabil zu halten.

Rigid Body Equation of Motion
Hydrodynamic Forces and Moments

Tabelle A.6: Methoden der kinetischen Modellierung (aus [80])

Modellierung

Bei der Modellierung eines angetriebenen Objekts gibt es zwei Modellierungsarten. Die Kinematische und die Kinetische. Die Arten unterscheiden sich nach Komplexität und gewünschter Genauigkeit des Modells.

Kinematisches Modell Wird ein Messverfahren angewandt bei dem die Erfassung unmittelbar zum gesuchten Messergebnis führt, wird dieses als kinematisch bezeichnet. Modelliert wird nur die Bewegung als zeitliche Änderung des Zustands, ohne Bezug zu den verursachenden Kräften [45]. Das zu modellierende Objekt wird in einer vereinfachten Form dargestellt um den Fokus der Modellierung auf die Bewegung zu legen. Als Massepunkt dargestellt wird das Objekt mit Bewegungsgleichungen belegt. Dadurch entsteht eine Bahn aus Positionen die durch inertiale Sensoren und GPS kontrolliert werden können. Hierdurch lässt sich dass Fahrverhalten des MOPS auf sechs Freiheitsgrade nachvollziehen.

Kinetisches Modell Im Gegensatz zum kinematischen Modell wird mit der echten Objektoberfläche modelliert. Hierdurch sollen die äußeren Einflüsse auf das Objekt möglichst real dargestellt werden können. Als äußere Einflüsse werden Kräfte und Momente am Objekt verstanden. Kinetische Modelle behandeln die aus den äußeren Kräften entstehenden Beschleunigungen [80]. Im Beispiel des MOPS sind äußere Einflüsse u.a. die Strömung, Wind und Wellenbewegungen. Zur Vorgehensweise dieser Modellierung gibt es mehrere Methoden, von denen einige hier kurz erwähnt werden. Tabelle A.6 zeigt Beispielhaft zwei Methoden [80].

Bei der Rigid Body Equation of Motion Methode werden eine Massenmatrize (M_{RB}) und eine Matrize mit den Koriolis- und Zentripetal-Kräften (C_{RB}) in Verbindung mit Geschwindigkeit und Beschleunigung gebracht, um mit den äußeren Kräften und Momenten gleichgesetzt zu werden (siehe Formel A.1). Hiermit kann die Bewegung eines Objekts durch Flüssigkeiten ohne Berücksichtigung des kinetischen Einflusses gezeigt werden. Daher muss das Modell z.B. durch die eine weitere Betrachtung erweitert werden. Hydrodynamic Forces and Moments bildet den Zusammenhang mit den hydrodynamischen Dämpfungen und die daraus entstehenden Kräften [80].

$$\tau_{RB} = M_{RB}\ \dot{v} + C_{RB}(v)v \tag{A.1}$$

Regelung

Die bisher wurde im MOPS eine PositionsRegelung implementiert, welche in verschiedenen Varianten nutzbar ist. Die Grundlage für diese Regelungsvarianten stellt ein PID-Regler dar. Die PID-Regelung existiert jeweils für die Regelung des Kurses und für die Regelung der Geschwindigkeit. Hierbei ist die Betrachtung der Zeitkonstanten der Regelstrecken wichtig. Der Regler mit den kleineren Zeitkonstanten muss von dem Regler mit den längeren Zeitkonstanten überlagert werden [113, S. 288]. Bezeichnet wird dieses Reglerkonstrukt als KaskadenRegelung und der unterlagerte Regler wird als Hilfsregler oder als innerer Regelkreis bezeichnet [35, S. 157].

Parameter-Identifizierung Die Parameter-Identifizierung ist das Herzstück der Regelungstechnik. Hiermit wird u.a. über die Systemstabilität oder die Reaktionsgeschwindigkeit des Systems entschieden.

Verfahren zur Parameter-Identifikation Es gibt eine Vielzahl an Verfahren zur Identifikation der Systemparameter. Herunter gebrochen auf das wesentliche Vorgehen, kristallisieren sich zwei Basistechniken heraus. Das Empirische Einstellen und das Auswerten der Sprungantwort.

Empirisches Einstellen Beim Empirischen Einstellen, oder Einstellen durch Probieren werden zuerst alle Parameter auf Null gesetzt. Dann wird der P-Anteil der Regelung schrittweise erhöht, bis das System anfängt zu schwingen. Danach wird der I-Anteil erhöht bis die Schwingungen aufhören. Der D-Anteil dient dazu das Systemverhalten zu optimieren, bzw. kann die Systemstabilität erhöhen.

Um diese Einstellmethode auf das MOPS IV-Projekt zu projizieren wird eine Sollposition dem Regler vorgegeben. Der P-Anteil wird soweit erhöht bis das Boot um die um die Zielposition schlingert. Danach wird der I-Anteil erhöht bis die Position wirklich getroffen wird. Da für jede Parameteränderung eine neue Testfahrt nötig ist, ist diese Methode aufgrund des hohen Zeitbedarfs nicht unbedingt geeignet.

Auswerten der Sprungantwort Bei dieser Vorgehensweise wird eine Sprungantwort auf das System gegeben und die Systemantwort aufgezeichnet. Aus der hieraus entstandenen Messkurve können Schlüsse auf die Systemparameter gezogen werden.

Abbildung A.68 zeigt die Zeiten und Werten aus denen, nach Verwendung der Einstellregeln, die Systemparameter generiert werden. Es gibt verschiedene Einstellregeln bei denen die auswählt wird, bei der das System am stabilsten reagiert. Ein Beispiel für Einstellregeln kommt von Ziegler und Nichols (siehe Formel A.2 und Tab. A.7). Andere Einstellregeln kommen z.B. von

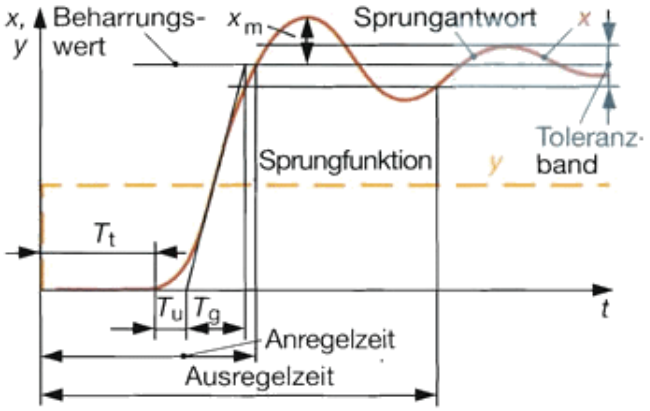


Abbildung A.68: Auswerten der Sprungantwort (aus [111])

Oppelt oder von Rosenberg [35, S. 143 ff.]. Formel A.2 zeigt die Grundberechnung für die Einstellung der Parameter. K_s stellt hierbei den Faktor aus Behaarungswert und Höhe der Eingangsgröße dar.

$$\lambda = \frac{T_g}{T_u \cdot K_s} \tag{A.2}$$

Regler	K_R	T_N	T_V
P	λ	-	-
PI	$0,9 \cdot \lambda$	$3,33 \cdot T_U$	-
PID	$1,2 \cdot \lambda$	$2 \cdot T_U$	$0,5 \cdot T_U$

Tabelle A.7: Einstellregeln nach Ziegler und Nichols (aus [35, S. 145])

Um die Einstellregeln anwenden zu können, empfiehlt es sich das Boot nach Kurs und Geschwindigkeit zu überwachen. Zuerst wird aus dem Stillstand heraus mit maximaler Beschleunigung auf einem Geradeauskurs gefahren bis die maximale Geschwindigkeit erreicht wird. Hieraus können die Parameter für den Geschwindigkeitsregler ermittelt werden. Darauf folgend wird eine rechtwinklige Kurve angestrebt. Hier wird die Fahrt z.B. via GPS überwacht. Die Kurvenfahrt sollte bei mehreren verschiedenen Geschwindigkeiten durchgeführt werden. Hierdurch soll gezeigt werden wie die Parameter evtl. abweichen. Sollte mit verschiedene Störgrößen gerechnet werden, kann evtl. aus den hier ermittelten Parametersätzen (der verschiedenen Kurvenfahrten) eine Anpassung vorgenommen werden.

Regelungsvarianten Um eine designierte Position anfahren zu können, ist es essentiell die Position zu regeln. Hierzu wurde bisher verschiedene Verfahren, und deren Kombination, implementiert [31, S. 60 ff.].

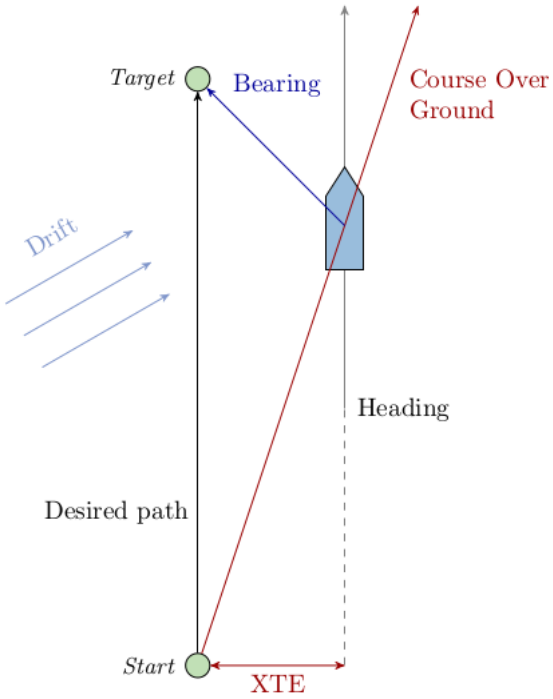


Abbildung A.69: Parameter der Regelungsverfahren (aus [31])

Abbildung A.69 zeigt eine prinzipielle Fahrt des Bootes zu einem Zielpunkt. Durch den Einfluss von Strömungen (als Drift bezeichnet), Wind und Trägheit des Bootes sind Abweichungen vom vorhergesehenen Kurs möglich. Verschiedene Regelungsvarianten versuchen den Einfluss dieser Störgrößen zu minimieren. Tabelle A.8 zeigt die implementierten Regelungsvarianten.

Direkte Regelung des Heading
Direkte Regelung des COG
Indirekte Regelung des COG
Regelung des XTE

Tabelle A.8: Regelungsvarianten (aus [31])

Laut [31, S. 65] ist es noch nicht möglich gewesen, einen Regler zu implementieren der bei schwachen und bei starken Auftreten von Störgrößen zufriedenstellende Ergebnisse erzielt. Aufgrund dessen wäre es denkbar in der aktuellen Projektphase eine erneute Parameter Identifikation, evtl. auch jeweils für jede Regelungsvariante, durchzuführen und den Einsatz der Regelungsvarianten erneut zu überdenken.

Seakeeping

Mit Seakeeping ist die Fähigkeit eines Boots oder Schiffs gemeint gegen raue Seeverhältnisse zu widerstehen. Rauhe Seeverhältnisse beschreiben z.B. hohe Wellen, starke Windverhältnisse und Strömungen die die Regelung des Boots, bzw. die Schwimmfähigkeit gefährden könnten. Tabelle A.9 zeigt die identifizierten Gefährdungen und die dazu vorgeschlagenen Maßnahmen.

Gefährdung	*Maßnahme*
Überspülen	Abdecken des Rumpfes
Wellen oder starker Seitenwind	Zentrierung der Massen
Starke Strömungen	Motorleistung ausreichend dimensionieren

Tabelle A.9: Maßnahmen zur Erhaltung der Seefähigkeit

Bevor im Folgenden die Gefährdungen und die zugehörigen Maßnahmen diskutiert werden sei Erwähnt, dass es sinnvoll ist, die in Tab. A.9 erwähnten Gefährdungen im Vorfeld zu vermeiden. Dies könnte durch ein Notfallprogramm, welches den MOPS zurück zur Heimatposition holt, geschehen. Das heißt der MOPS müsste z.B. durch eine eigene Wetterstation herannahende Schlechtwetterfronten detektieren können und entscheiden ob ein Missionsabbruch anstehen soll oder ob die Mission weitergeführt werden kann.

Überspülen Ein Überspülen und ein damit verbundener Wassereinbruch gefährdet die Schwimmfähigkeit des Boots. Des Weiteren beeinträchtigt die zusätzliche Masse des Wasser die Regelung des Bootes. Dieses könnte durch eine Abdeckung des Rumpfes verhindert werden.

Wellen oder starker Seitenwind Höhere Wellen oder starker Seitenwind könnten das Boot umwerfen. Hierzu ist es Sinnvoll den Schwerpunkt auf einen möglichst niedrigen und mittigen Punkt im Boot zu platzieren. Evtl. ist es

sogar Sinnvoll einen Ballasttank unter dem Boot anzubringen. Laut [51] wird die Schiffsstabilität nach Formel A.3 berechnet.

$$h_M = \frac{I_y}{V_W} - s \tag{A.3}$$

h_M	Stabilitätskriterium
I_y	Flächenträgheitsmoment der Schwimmfläche
V_W	verdrängtes Wasservolumen
s	Abstand Schwerpunkt S_G zum Auftriebsschwerpunkt S_A

Tabelle A.10: Bezeichner der Schiffstabilitäts-Formel

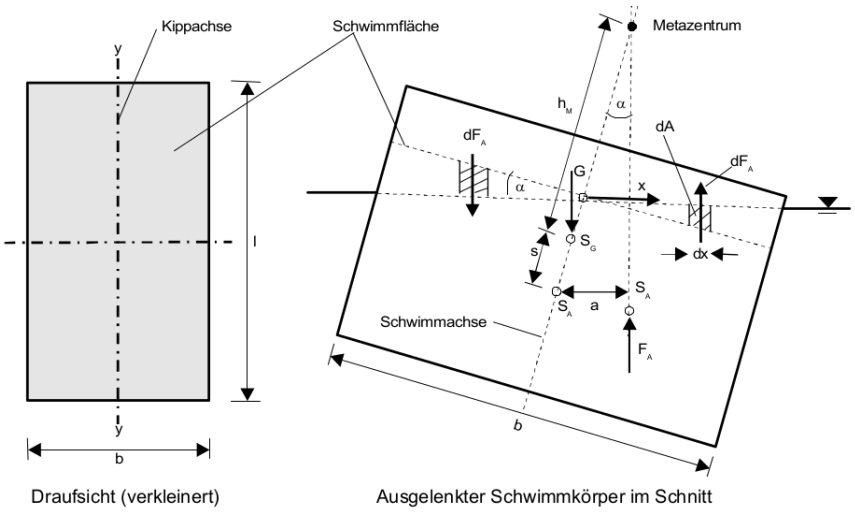


Abbildung A.70: Stabilitätskriterien (aus [51])

Stabilität nach Formel A.3 und Abb. A.70 entsteht wenn $h_M > 0$ ist. Dieses kann durch zwei Maßnahmen sichergestellt werden. Ersten durch ein sehr großes Verhältnis von Flächenträgheitsmoment der Schwimmfläche zu verdrängtes Wasservolumen. Sprich ein großes Flachenverhältnis gegen ein kleines verdrängtes Wasservolumen. Oder durch ein kleinen Wert von s. Dieses wird durch einen kleinen Abstand Schwerpunkt S_G zum Auftriebsschwerpunkt S_A erreicht.

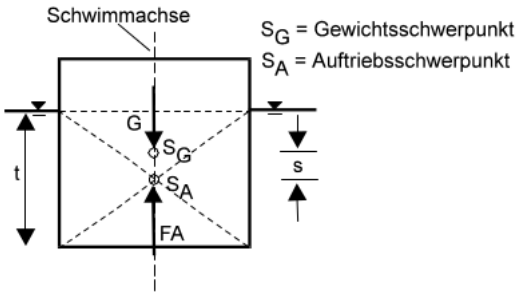


Abbildung A.71: Schwimmschwerpunkt (aus [51])

Laut Abb. A.71 ist es sinnvoll den Gewichtsschwerpunkt an oder unter den Auftriebsschwerpunkt zu bringen. Dieses könnte z.B. durch einen Ballasttank erreicht werden.

Starke Strömungen

Beim Wechsel der Gezeiten treten starke Strömungen auf. Unter normalen Wetterbedingungen treten Strömungsgeschwindigkeiten von $1,4\,m/s$ auf.[39] Unter rauen Wetterbedingungen ist davon auszugehen das sich diese Geschwindigkeit stark erhöht. Auf den Seekarten des BSH [11] wird dieses belegt.

$$1.4\frac{\mathrm{m}}{\mathrm{s}} \cdot 3.6 = 5.04\frac{\mathrm{km}}{\mathrm{h}} \tag{A.4}$$

$$\frac{5.04\frac{\mathrm{km}}{\mathrm{h}}}{1,852} = 2.72\ \mathrm{kn} \tag{A.5}$$

Formel A.5 zeigt dass die Strömungen unter Normalbedingungen schon 2.72 kn betragen kann. Laut [31, S. 52] beträgt die bisher gemessene Höchstgeschwindigkeit des Wasserfahrzeugs $8\frac{\mathrm{km}}{\mathrm{h}}$.

$$\frac{8\frac{\mathrm{km}}{\mathrm{h}}}{1.852} = 4.32\ \mathrm{kn} \tag{A.6}$$

$$4.32\ \mathrm{kn} - 2.72\ \mathrm{kn} = 1.6\ \mathrm{kn} \tag{A.7}$$

Durch Formel A.7 wurde belegt, dass unter normalen Wetterbedingungen bei der Fahrt gegen die Strömung nur eine maximale Geschwindigkeit von 1.6 kn einstellt. Unter rauen Wetterbedingungen mit stärkeren Strömungen reduziert sich die maximale Geschwindigkeit. Dieses kann dazu führen dass das Boot evtl. nicht mehr auf Kurs zu halten ist. Abhilfe kann durch die Verwendung von Motoren mit größerer Leistung geschaffen werden.

Fazit und kritische Bewertung

Im bisherigen Projektverlauf ist seitens der Sensorik und der Regelung schon eine solide Grundarbeit geschaffen worden. Jetzt gilt es die Feinarbeit anzugehen. Seitens der Nutzlast gibt es schon Sensoren in einem Käfig bei denen die Busanbindung auf ein robusteres Protokoll umgestellt werden könnte. Da seitens anderer Projektziele ein leistungsstärkerer Antrieb und evtl. ein Radargerät implementiert werden sollen, kann mit einer höheren EMV-Belastung gerechnet werden. Dieses kann den Protokollwechsel notwendig machen. Des Weiteren sollte die Ausbringposition nach den weiteren Einbauten noch einmal überprüft werden. Hiernach gilt es noch zu klären ob eine Transportsicherung für den Sensorkäfig während der Fahrt notwendig ist. Um die Regelung letztendlich zu optimieren kann die Parameteridentifikation bzw. die Regelungsvarianten noch einmal überarbeitet werden.

Anhang B

Protokolle

B.1 Sitzungprotokolle

B.1.1 Kick-off-Meeting am 13. Oktober 2015

Moderator: *Sascha Hornauer*
Protokollant: *Björn Koopmann*
Abwesende Gruppenmitglieder: *Maximilian Hipp (bis 13:00 Uhr)*

TOP 1 – Begrüßung Sascha begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Vorstellung der Projektgruppe Im Anschluss an die Begrüßung und eine kurze Vorstellung der Betreuer werden die Ergebnisse der Vorgängergruppen sowie die Aufgabenstellung der aktuellen Projektgruppe präsentiert. Die Bearbeitungszeit beträgt zwei Semester. Neben einem Abschlussbericht zum Ende der Projektgruppe muss im März 2016 ein Zwischenbericht abgegeben werden. Weitere Informationen sind dem entsprechenden Foliensatz zu entnehmen, der im Stud.IP zu finden ist.

TOP 3 – Terminfindung Als Termin für die regelmäßigen Gruppensitzungen einigen sich die Gruppenmitglieder auf den Zeitslot am Dienstag von 10:00 bis 12:00 Uhr. Die erste Sitzung wird voraussichtlich am 20. Oktober stattfinden. Für diesen und alle weiteren Termine steht der Projektgruppen-Raum U105 im OFFIS ab sofort zur Verfügung.

TOP 4 – Themen für die Seminarvorträge In den ersten Wochen der Bearbeitungszeit sollen durch die Erstellung von Seminarvorträgen erste Inhalte erarbeitet und die Eigenschaften des bestehenden Systems erfasst werden. Im Anschluss an eine kurze Vorstellung der verfügbaren Themen wird die durch Sascha moderierte Einteilung vollzogen.

B.1.2 Gruppensitzung am 20. Oktober 2015

Moderator: *Dennis Pilny*
Protokollant: *Björn Koopmann*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Dennis begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Vorstellung der Seminarvorträge

- Michael hält seinen Vortrag zur *Hardware des bestehenden Systems* (A.2).
- Dennis hält seinen Vortrag zur *Software des bestehenden Systems* (A.1).
- Henning hält seinen Vortrag zu den *Rechtlichen Rahmenbedingungen* (A.3).

TOP 3 – Moderation

- In jeder Woche soll die Einladung zur nächsten Gruppensitzung bis spätestens Montagmittag um 12:00 Uhr per E-Mail an alle Gruppenmitglieder verschickt werden.
- Vorschläge für mögliche Tagesordnungspunkte werden durch den Moderator über Slack oder das Etherpad *Vorschläge* entgegengenommen.
- Michael schickt eine E-Mail mit dem nächsten Moderator/Protokollanten herum. Die Liste wird auch im DokuWiki veröffentlicht.

TOP 4 – Kommunikation

- Der E-Mail Verteiler `pg-mops-4@informatik.uni-oldenburg.de` steht den Mitgliedern der Projektgruppe ab sofort zur Verfügung.
- Ein Slack-Chat wird eingerichtet. Dieser ist unter der Adresse `https://mops4.slack.com` oder über die bereitgestellten Apps verfügbar.

TOP 5 – Rollenverteilung

- Konstantin übernimmt die Aufgabe des Webauftritt-Beauftragten.
- Peter übernimmt die Aufgabe des ICBM-Beauftragten.
- Zahra übernimmt die Aufgabe der Wiki-Beauftragten.

B.1.3 Gruppensitzung am 27. Oktober 2015

Moderator: *Michael Beering*
Protokollant: *Konstantin Gebel*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Michael begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Vorstellung der Seminarvorträge

- Konstantin hält seinen Vortrag zur *Maritimen Umgebung* (A.4).
- Maximilian hält seinen Vortrag zur *Schiffssensorik* (A.5). Es wird angemerkt, dass wir uns die Frage stellen sollten, welche Sensoren für unser Projekt geeignet sind. Insbesondere bei dem Radar sollten wir uns erkundigen, ob eine Anbindung an die bestehende Hardware möglich ist.
- Zhara hält ihren Vortrag zum *Projektmanagement* (A.6). Die Mehrheit möchte Jira als Tool zur Projektverwaltung nutzen.
- Eike hält seinen Vortrag zu *verwandten Projekten* (A.7) und *alternativen Energiequellen* (A.8). Es wird angemerkt, dass der Bootsaufbau der Plattform je nach Projektziel überarbeitet werden kann.
- Es wird angefragt, ob wir einen eigenen Internetzugang am OFFIS bekommen können. Sascha wird sich um die Beantwortung der Frage bemühen.
- Um einen Schlüssel für das OFFIS zu bekommen, sollen wir uns bei Sascha melden. Das Pfand für einen Schlüssel beträgt 40,00 Euro.

TOP 3 – Vorstellung von Wiki, Etherpad und Slack

- Dennis stellt das von ihm vorbereitete DokuWiki vor. Er geht auf die Struktur der Seiten ein und wie diese bearbeitet werden. Er weist auf die spezielle Syntax des Wikis hin.
- Die Frage nach einem zweiten Termin wird gestellt. Es wird festgehalten, dass wir uns nach der Seminarphase darum kümmern.
- Die angefertigten Protokolle sollen spätestens zwei Tage nach der Gruppensitzung im Wiki hochgeladen werden.

B.1.4 Gruppensitzung am 3. November 2015

Moderator: *Konstantin Gebel*
Protokollant: *Eike Hagena*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Konstantin begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Vorstellung der Seminarvorträge

- Björn hält seinen Vortrag zur *Kollisionserkennung und -verhütung*. Sascha merkt an, dass ein Radargerät der Abteilung eventuell für unsere Zwecke genutzt werden könnte.
- Peter hält seine Vorträge zu *Umweltsensoren* und *Control and Modeling*.

TOP 3 – Vorstellung des Git

- Max stellt das Git kurz vor und zeigt im Zuge dessen, wie der Pfad, der Benutzername und das Passwort konfiguriert werden müssen.
- Alle Gruppenmitglieder werden dazu aufgefordert, ihre mit PuTTYgen erzeugten Public-SSH-Keys an Max zu senden.

TOP 4 – Projektmanagement

- Jira ist im OFFIS eingerichtet, wir besitzen im Normalfall alle Rechte, ansonsten sollen wir uns bei Sascha melden. Es gibt eventuell Probleme, wenn wir vom OFFIS aus darauf zugreifen wollen.
- Wenn es neue Beschlüsse gibt, sollen diese auf einer eigens für diesen Zweck angelegten Wikiseite festgehalten werden. Die neuen Beschlüsse werden vom jeweiligen Protokollanten erweitert. Wichtig dabei ist, dass das Datum des Beschlusses mit dabei steht.
- Die Foliensätze der einzelnen Seminarvorträge sollen als PDF ins Wiki geladen und auf der entsprechenden Seiten eingebunden werden.
- Der Termin für das zweite Treffen wird über eine Doodle-Umfragen bestimmt. Eventuell müssen wir einen anderen Raum für das zweite Treffen finden.
- Für die Ausarbeitungen der Seminarthemen gibt es keine Vorgaben hinsichtlich der Länge.
- Alle Gruppenmitglieder sollen sich bis zur nächsten Woche Gedanken über die Formulierung des übergeordneten Projektziels machen.

B.1.5 Gruppensitzung am 10. November 2015

Moderator: *Eike Hagena*
Protokollant: *Maximilian Hipp*
Abwesende Gruppenmitglieder: *Michael Beering (Urlaub)*

TOP 1 – Begrüßung Eike begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Projektziele Die Gruppe schlägt folgende Ideen für die Projektziele vor:

- Zukünftige Nutzung regenerativer Energie zur Optimierung des Antriebs. Dafür sollen Solar-Module genutzt werden.
- Einsatz stärkerer Motoren für die Schiffssteuerung.
- Einsatz von Radargeräten zur Erkennung von Objekten auf dem See.
- Konstruktion eines Korb- und Seilwindensystems, um die Sensoren zur Durchführung von Messungen in der Tiefe herablassen zu können.
- Ermöglichen eines Missionsabbruch in Gefahrensituationen.
- Absicherung gegen eindringendes Wasser.

Die genaue Ausformulierung der Projektziele sowie der Vision soll zu einem späteren Zeitpunkt in mehreren Teams erfolgen.

TOP 3 – Urlaubsplanung Alle Gruppenmitglieder erhalten während der Weihnachtsferien, also im Zeitraum vom 22. Dezember 2015 bis zum 3. Januar 2016 Urlaub. Die ersten Gruppensitzung im neuen Jahr findet damit am 5. Januar statt. Die Beantragung von Urlaubstagen oder -wochen sollte in Zukunft nach dem folgenden Muster ablaufen:

1. Der Termin wird dem Projektleiter (Dennis) frühzeitig mitgeteilt.
2. Der Projektleiter bestätigt Termin oder lehnt die Anfrage ab.
3. Der Beantragende trägt den Urlaub in den Google-Kalender ein.

TOP 4 – Termine Sascha wird damit beauftragt, den Projektgruppen-Raum U105 an den folgenden Terminen für weitere Gruppensitzungen zu reservieren:

- Montag, 14:00 bis 18:00 Uhr
- Dienstag, 8:00 bis 10:00 Uhr
- Mittwoch, 16:00 bis 20:00 Uhr
- Donnerstag, 16:00 bis 20:00 Uhr

Als weiterer Termin für eine zweite wöchentliche Gruppensitzung wird Donnerstag von 18:00 bis 20:00 Uhr festgelegt.

TOP 5 – Verschiedenes

- Konstantin Gebel übernimmt die Aufgabe des Jira-Beauftragten.
- Björn Koopmann ist Doku-Beauftragter.
- Dennis Pilny wird als Projektleiter ernannt.
- Max Hipp richtet einen FTP-Server ein.
- Der Einsatz von Scrum als Vorgehensmodell wird beschlossen.
- Dennis und Konstantin bereiten das Backlog vor.

TOP 6 – Vorläufige Planung für die Zukunft

- Zwei bis drei Wochen für die Erarbeitung der Projektvision
  - Björn arbeitet eine LaTeX-Vorlage aus, in der die zuvor eingeteilten Teams ihre Ideen ausformulieren können.
- Erarbeitung von Systemumgebung und Abgrenzung
  - Was kann die aktuelle Lösung? Was soll die neue können?
  - Gespräche, Interviews, Zusammenfassungen
- Weitere zwei bis drei Monate für eine umfassende Anforderungsanalyse

B.1.6 Gruppensitzung am 17. November 2015

Moderator: *Maximilian Hipp*
Protokollant: *Björn Koopmann*
Abwesende Gruppenmitglieder: *Peter Tank (Urlaub)*

TOP 1 – Begrüßung Max begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Vorstellung des ersten Sprints Max präsentiert die von Zahra, Dennis und Konstantin angelegten Tasks für den aktuellen Sprint. Die Dauer des ersten Sprints beträgt drei Wochen. Er endet am 7. Dezember 2015.

Dennis stellt die einzelnen Aufgabenpakete vor und legt den Umfang den anfallenden Arbeiten fest. Dabei sind insbesondere die folgenden Punkte relevant:

- Der Zeitansatz für die Erstellung der Projektvisionen beträgt zwei Wochen. Es sollen drei unabhängige Versionen in Gruppen aus jeweils drei Personen erstellt werden, die am 1. Dezember 2015 zu einer gemeinsamen Version zusammengefügt werden.

- An dem Interview mit Prof. Dr. Oliver Zielinski sollen ein Interviewer (Peter), ein Protokollant (Michael) und ein Beobachter (Björn) teilnehmen. Zur Vorbereitung des Gesprächs wird ein Interviewleitfaden erstellt, dessen Inhalt mit den anderen Gruppenmitgliedern abgestimmt wird.

- Das Treffen mit Prof. Dr.-Ing. Axel Hahn wird voraussichtlich in der ersten Dezemberwoche stattfinden. Sascha weist darauf hin, dass für diesen Termin eine gute Vorbereitung aller Gruppenmitglieder sinnvoll ist. Die Inhalte der Projektvision sollen nach Möglichkeit bereits in das Gespräch einbezogen werden. Dennis koordiniert die Vorbereitungen.

Konstantin und Max erläutern das Prinzip der Story Points. Für den ersten Sprint wird keine Aufwandsschätzung durchgeführt. Alle Gruppenmitglieder werden darum gebeten, ein Profilbild in ihren Jira-Account einzubinden. Für die Verwendung des Scrum Boards sind die folgenden Spalten vorgesehen:

1. *ToDo* – Die Aufgabe ist geplant, wurde aber noch keinem Bearbeiter zugewiesen oder konnte bisher nicht abgearbeitet werden.

2. *In Progress* – Die Aufgabe befindet sich in Arbeit.

3. *In Review* – Die Lösung der Aufgabe wird durch ein zweites Gruppenmitglied auf Vollständigkeit, Korrektheit und Konsistenz überprüft.

4. *Cancelled* – Der Bearbeiter konnte keine hinreichend gute Lösung für die gestellte Aufgabe finden.

5. *Done* – Die Aufgabe wurde vollständig bearbeitet und durch den Reviewer überprüft. Es sind keine weiteren Schritte zur Fertigstellung erforderlich.

TOP 3 – Einteilung der Gruppen für die Projektvision Für die Erstellung der vorläufigen Projektvisionen werden drei Gruppen mit jeweils drei Personen gebildet. Die Aufstellung der Gruppen ergibt sich wie folgt:

Gruppe A	*Gruppe B*	*Gruppe C*
Michael	Eike	Konstantin
Björn	Henning	Maximilian
Peter	Zahra	Dennis

TOP 4 – Verschiedenes

- Björn präsentiert die LaTeX-Vorlagen für die Projektvision und die Sitzungsprotokolle. Die fertigen Protokolle sollen ins Git-Repository hochgeladen und per E-Mail an den Verteiler gesendet werden.

- Die zusätzliche Gruppensitzung am 19. November 2015 wird aufgrund fehlender Themen entfallen. Die nächste (zusätzliche) Sitzung findet damit am 26. November 2015 statt. Der freigewordene Zeitslot kann durch die zuvor bestimmten Kleingruppen für die Erarbeitung der Projektvision genutzt werden.
- Zahra, Konstantin und Dennis erklären sich bereit, auch den nächsten Sprint zu planen. Ihre Ergebnisse werden am 8. Dezember 2015 vorgestellt.

B.1.7 Gruppensitzung am 24. November 2015

Moderator: *Björn Koopmann*
Protokollant: *Henning Lawatsch*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Björn begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Vorstellung der LaTeX-Vorlage Björn stellt sein Dokument für den Zwischenbericht vor. Ein vorläufiges Inhaltsverzeichnis ist bereits vorhanden. Chapter sind einzeln zu bearbeiten, allerdings nur eine Person zur selben Zeit. Bilder gehören in den Ordner `images`, Sourcecode in den Ordner `listings`. Für das Compilen sind die Dateien `compile.bat` (unter Windows) sowie `compile.sh` (unter Linux) vorgesehen. Es wird darum gebeten, die Hinweise zur Verwendung zu beachten.

TOP 3 – Brainstorming: Relevante Stakeholder Um die verschiedenen Interessen an Mops iv herauszufiltern, wurde in einem gemeinsamen Brainstorming eine Mindmap erstellt, die in Abb. B.1 enthalten ist.

- **Mops iv:** In erster Linie steht unsere Gruppe
- **Mops i-iii:** Möglicherweise besteht auch Interesse von den Vorgängergruppen, insbesondere Mops iii.
- **Universität:** Das Projekt wird von der Universität Oldenburg geleitet
- **Axel Hahn:** Axel Hahn ist der zuständige Professor
- **Betreuer:** Unterstützend wirken Sascha und Mohammad beim Gruppengeschehen mit
- **OFFIS:** Die Gruppensitzungen werden im OFFIS abgehandelt, welches um die entsprechende Ausrüstung verfügt.
- **ICBM:** Das ICBM ist der Auftraggeber, Anforderungssteller und eventuell auch Nutzer

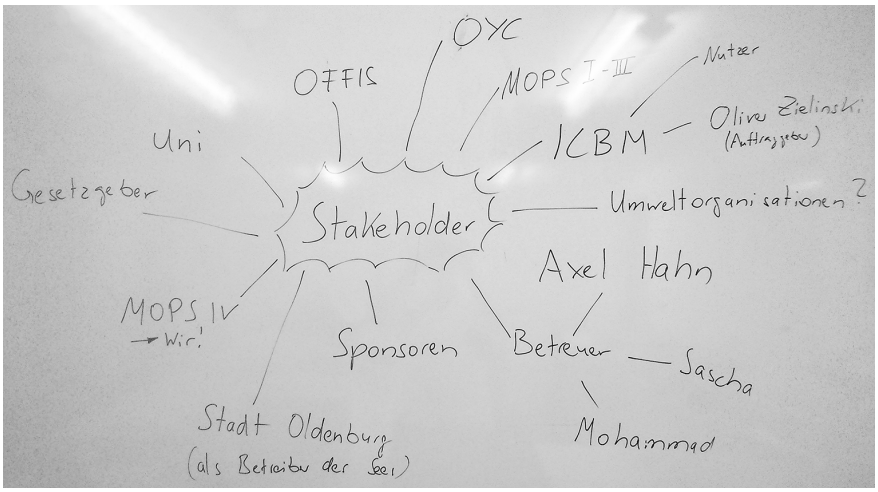


Abbildung B.1: Erfasste Stakeholder

- **Oldenburger Yacht-Club:** Stellt uns Ausrüstung am Tweelbäker See zur Verfügung.
- **Gesetzgeber:** Ein autonomes, selbstgebautes Boot benötigt die rechtliche Zustimmung Seitens der Zuständigen Behörden.
- **Stadt Oldenburg:** Der Tweelbäker See gehört zu Oldenburg und ist damit unter der Verantwortung der Stadtverwaltung.
- **Sponsoren:** Möglicherweise gewinnen wir im Laufe des Projekts Sponsoren, die das Projekt unterstützen.

TOP 4 – Inhalte der geplanten Interviews Relevante Frage sind:

- Was haben wir bereits gemacht? Zeigen unserer Projektvision.
- Wichtige Fragen an Zielinski: Wie soll die Plattform zum Einsatz kommen? Wo soll er eingesetzt werden? Welche Sensoren werden benötigt? Soll das Wasserfahrzeug nur für gutes Wetter geeignet sein? Welche Messungen sind relevant?
- Wichtige Fragen an Hahn: Welche Sicherheitsanforderungen gibt es? Welche Seezeichen müsse beachtet werden? Sind Positionslichter oder Signalkörper erforderlich? An welchem Hafen können wir wann eine Testfahrt durchführen? Welches Budget steht uns zur Verfügung?
- Wie sieht sein Zeitplan für uns aus?

TOP 5 – Weitere Vorgehensweise Teamintern soll bereits überlegt werden, welches die nächsten relevanten Schritte sind. Nach der Fertigstellung der Projektvision und Durchführung der Interviews werden Kleingruppen gebildet, die sich mit speziellen Rahmenbedingungen befassen (Spezialgebiete). Darauf entwickeln wir eine Anforderungsdefinition.

TOP 6 – Verschiedenes

- Am Donnerstag, den 24. November 2015, fällt das zusätzliche Gruppentreffen aus. Jeder Gruppe steht es frei sich an diesem Termin zu Treffen und ihre Aufgaben zu erledigen.
- Weitere Ergänzungen zu den Interview-Fragen können an Peter gerichtet werden.
- Michael übernimmt die Aufgabe des Raum-Beauftragen und übernimmt die Kommunikation mit der Projektgruppe SESAdata.
- Eike und Henning sind ab jetzt für die Finanzen zuständig, d.h. für die Verwaltung von Rechnungen und Budgetplanung.
- Der Zwischenbericht soll Anfang bis Mitte März abgegeben werden.

B.1.8 Gruppensitzung am 1. Dezember 2015

Moderator: *Henning Lawatsch*
Protokollant: *Zahra Paya*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Henning begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Vorstellung der Projektvisionen Im Anschluss an eine ausführliche Präsentation werden die Inhalte der drei Teams in der Gesamtgruppe diskutiert.

TOP 3 – Zusammenführung der Visionsdokumente Zur Erstellung der Endvision werden die einzelnen Kapitel aller Visionen schrittweise betrachtet, besprochen und zusammengefügt.

Sascha schlägt vor, die spezifizierten Eigenschaften in eine Menge von funktionalen und nicht-funktionalen Anforderungen zu sortieren. Er findet auch die Kollisionsvermeidung einen guten Punkt.

Eike und Henning erklären sich bereit, die Endvision zu überarbeiten.

TOP 4 – Vorbesprechung für die Interviews Peter stellt seine Ideen für das Treffen mit Herrn Prof. Dr. Zielinski vor. Es soll kurz um die Projektvision und seine Vorstellungen gehen. Die Priorität der Punkte soll am Donnerstag mit Herr Prof. Dr. Hahn erfragt werden. Eine Liste mit Fragen wird in einem Etherpad gesammelt und kann ergänzt werden.

Das geplante Interview wird am 8. Dezember 2015 um 9:00 Uhr in Raum W15 1-148 stattfinden. Weitere Informationen werden von Peter als Termineinladung gesendet.

TOP 5 – Verschiedenes

- Sacha bittet Dennis und Max darum, einen weiteren Account mit Leserechten für das Mops iii-Wiki einzurichten.
- Es wird beschlossen, dass die Seminarausarbeitungen im nächsten Sprint in den Zwischenbericht übertragen werden sollen.
- Björn merkt an, dass noch nicht alle Gruppenmitglieder ein Profilbild für ihre Jira- bzw. Slack-Accounts hochgeladen haben. Die entsprechenden Einstellungen sollen im Verlauf der nächsten Woche vorgenommen werden.

B.1.9 Gruppensitzung am 3. Dezember 2015

Moderator: *Dennis Pilny*
Protokollant: *Björn Koopmann*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Dennis begrüßt die anwesenden Gruppenmitglieder.

Auch Prof. Dr. Hahn begrüßt die anwesenden Mitglieder der Projektgruppe. Nach einer kurzer Vorstellungsrunde lobt er die bisherige Arbeit und verweist auf die positiven Rückmeldungen von Sascha und Mohammad.

TOP 2 - Vorstellung der Projektvision Dennis hält einen Kurzvortrag über den aktuellen Stand der Projektgruppe. Im Anschluss an die Vorstellung der Projektvision gibt er einen kurzen Überblick über das geplante Vorgehen und beschreibt die für die nächsten Wochen vorgesehenen Arbeitspakete.

TOP 3 - Gewichtung der Projektziele Die Überarbeitung der Projektvision intensivierend, werden die zuvor erarbeiteten Projektziele schrittweise auf ihre Relevanz und Konsistenz mit den jeweils anderen Zielvorstellungen überprüft. Eine Mitschrift der im weiteren Verlauf mit Prof. Dr. Hahn geführten Gespräche ist dem Interviewprotokoll aus Anhang B.2.1 zu entnehmen.

TOP 4 – Vorbesprechung des Interviews Peter stellt seine Ideen für den Interview-Leitfaden des Gesprächs mit Prof. Dr. Zielinski vor und wird durch die übrigen Gruppenmitglieder ergänzt. Auch die diesbezüglichen Anmerkungen von Prof. Dr. Hahn sind im Abschnitt B.2.1 zu finden.

B.1.10 Gruppensitzung am 8. Dezember 2015

Moderator: *Zahra Paya*
Protokollant: *Dennis Pilny*
Abwesende Gruppenmitglieder: *Konstantin Gebel, Henning Lawatsch, Eike Hagena (ab 12:00 Uhr)*

TOP 1 – Begrüßung Zahra begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Sprint Review Das Erstellen des Visionsdokuments wurde abgeschlossen. Inhaltliche Änderungen die auf dem Interview mit Prof. Hahn und Prof. Zielinski aufbauen werden im kommenden Sprint hinzugefügt.

Eine Möglichkeit die Dokumentation kontinuierlich fortzuführen, wäre es zugewiesene Tasks zwingend an die Dokumentation dieser zu knüpfen. So dass ein Task erst dann als erfüllt gilt, wenn auch dieser auch ausreichend dokumentiert wurde.

TOP 3 – Sprint Retrospektive

Positiv	*Negativ*
Die Zusammenarbeit (auch in den Kleingruppen) hat gut geklappt	Die Zusammenführung des Visionsdokumentes in der Sitzung war sehr langwierig
Jira zur Projektorganisation ist gut	Das Tracking der Stunden im Jira hat nicht gut funktioniert
Die Dokumentation funktioniert gut	Es wurde in Anbetracht der KP bisher zu wenig Zeit investiert
	Es hat keine Aufwandschätzung der Tasks stattgefunden

TOP 4 – Vorstellung des Dokuments für den Zwischenbericht Björn stellt die überarbeitete Version des Zwischenberichts vor. Für die Arbeit am Zwischenbericht sind die *Hinweise zur Verwendung* unbedingt zu beachten. Dieser enthält alle definierten Konventionen und Hilfsmittel zum Arbeiten am LaTeX-Dokument. Bei Änderungswünschen soll sich an Björn gewandt werden.

Zusätzlich wurde noch einmal angemerkt, dass LaTeX-Code, der sich nicht kompilieren lässt nicht ins Git gepusht werden darf, d.h. vor jedem Push muss überprüft werden, ob der Code sich kompilieren lässt.

TOP 5 – Besprechung des Interviews mit Prof. Zielinski Peter stellt die Ergebnisse des Interviews mit Prof. Zielinski vor. Die detaillierten Inhalte finden sich in dem Interviewprotokoll von Michael.

TOP 6 – Überarbeitung der Projektvision Für die Überarbeitung der Projektvision wurden die bereits erarbeiteten Lösungsideen betrachtet, mit Anmerkungen versehen und mit Noten (in Schulnoten Skala (1-6)) bewertet. Grundlage für die Bewertung ist dabei die Priorisierung, die aus den Interviews mit Prof. Hahn und Prof. Zielinski herausgekommen ist. Im Folgenden sind die

Antrieb Note 5, da die Verbesserung des Antriebs relativ einfach durch den Kauf stärkerer Motoren gelöst werden kann. Eine Verbesserung des Antriebs würde im aktuellen Zustand keinen großen Mehrwert für den Projekterfolg bringen.

Ausfallsicherheit (Sicherheit) Note 2+, da der Verlust des Bootes während einer Mission nicht ...

Benutzerfreundlichkeit Note 2, da die Verbesserung der Benutzerfreundlichkeit ein wesentlicher Bestandteil zur Inbetriebnahme Dritter nötig ist. Geplant ist, dass das Boot unbeaufsichtigt bis zu 12 Stunden Messungen durchnimmt.

Kollisionserkennung Note 1, da die Erkennung von dynamischen Hindernissen noch nicht implementiert ist.

Kollisionsvermeidung Note 1, analog zur Kollisionserkennung.

Energieversorgung Note 5, da analog zum Antrieb, die Versorgungslage leicht durch die Erhöhung der Batteriekapazität erreicht werden kann. Dies stellt die Projektgruppe nicht vor einer großen Herausforderung.

Rechtlicher Rahmen Note 6, da der Rechtliche Rahmen weiterhin eine Grauzone ist, die bereits von vorherigen Projektgruppen angegangen wurde, ohne ein konkretes Ergebnis zu liefern.

Regelung Note 1, da die Verbesserung der Bahnplannung unter Einbeziehung der Witterungsbedingungen ein wichtiger Bestandteil zur Navigation auf dem See ist.

Reichweite und Missionsoptimierung Note 3, da es diverse Möglichkeiten gibt die Reichweite des Wasserfahrzeugs unter Berücksichtigung diverser Faktoren (Strömung, Wind, Uhrzeit, ...) zu verbessern. Peter erhielt von Prof. Zielinski Papers die mögliche Einsparungspotentiale behandeln.

Seetauglichkeit Note 2, da sichergestellt werden muss, das die Plattform bei einem Wassereintritt nicht ausfällt.

Sensormessung Note 3, da die Bereitstellung einer stabilen Sensorumgebung (Moon-Pool, ...), die das Hinzufügen von Sensoren ermöglicht, ein wichtiger Bestandteil des Projekts sein soll.

Stealth-Problematik Note 3-

Zeitaufwand Wird mit Benutzerfreundlichkeit zusammengetan.

TOP 7 – Interessen der Projektgruppe Die priorisierten Problemlösungen wurden durchgegangen und die verbliebenen Mitglieder meldeten sich für die Themen, an denen sie Interesse haben. Diese Zuweisung ist jedoch noch nicht verpflichtend und dient nur der Übersicht.

Antrieb Keine Interessenten

Ausfallsicherheit (Sicherheit) Keine direkten Interessenten, wurde eher als übergeordnetes Thema aufgefasst, welches andere Problemlösungen mit einschließt.

Benutzerfreundlichkeit Ebenfalls eher übergeordnet. Benutzerfreundlichkeit der Software explizit: Maximilian, Michael, Dennis)

Kollisionserkennung Björn, Dennis, Peter, Maximilian

Kollisionsvermeidung Peter, Zahra, Maximilian, Björn, Michael

Regelung Peter, Björn

Energieversorgung Zahra

Reichweite und Missionsoptimierung Björn, Dennis

Seetauglichkeit Björn, Dennis, Michael, Maximilian

Sensormessung Dennis, Michael, Björn, Peter, Zahra

Stealth-Problematik Maximilian

TOP 8 – Sprint Planung Aufbauend auf der Priorisierung der Problemlösungen der Stakeholder sollen im kommenden Sprint Speziallistengruppen erstellt, die für die jeweiligen Themengebiete Anforderungen erheben. Der Sprint beginnt mit dem 8.12.2015 und endet am 5.01.2016. Da nicht alle Mitglieder anwesend waren, wurde das Einteilen der Gruppen auf die Sitzung am 10.12.2015 verschoben, um die Interessen aller einzubeziehen.

B.1.11 Gruppensitzung am 10. Dezember 2015

Moderator: *Dennis Pilny*
Protokollant: *Peter Tank*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Dennis begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Gruppeneinteilung Die Gruppeneinteilung der Spezialisten wird besprochen, bzw. nach Vorschlägen zum Vorgehen gefragt. Die Gruppen Kollisionserkennung und Kollisionsvermeidung sind die Gruppen mit den meisten Eintragungen. Ziele sind es funktionale und nicht funkionale Anforderungen in den Gruppen auszuarbeiten. Vielleicht kann Björn eine Vorlage erstellen. Die Gruppe Benutzerfreundlichkeit soll Software als Schwerpunkt haben und wird in Software Usability umbenannt. Energieversorgung und Reichweite werden zusammengefasst. Seetauglichkeit und Stealthproblematik werden zusammengefasst. Kollisionsvermeidung und Regelung werden zusammengefasst. Zu den Gruppen werden Buzzwords und Schwerpunkte zur späteren Ausarbeitung identifiziert.

Software Usability
Onshore auf Usability testen
Simulation testen
Anzeigen von dynamischen Hindernissen
Ausblenden von bestimmten Manövern (Radien...)
Was kann openseamap noch liefern (z.B. Metadaten)
Generieren von automatischen Sperrzonen
Def. von Bereichen die nur kurz befahren werden sollen
Onboardsoftware dynamisch USB Port Erkennung
Auswertung von Sensordaten allg. in einer GUI
Anzeige von möglichen Kollisionen

Buzzwords werden von Konstantin in die Tasks eingefügt. In den Ausarbeitungen von den Anforderungen soll etwas Hintergrundwissen beigefügt werden. Dazu gefundene Paper werden im Git gespeichert. Es werden nur Anforderungen für Themen erhoben die auch durchgeführt werden. Es werden

Kollisionserkennung
siehe Björns Ausarbeitung
wie und womit können Objekte erkannt werden
Tracking von Objekten
Mustererkennung und Auswertung (SW oder HW)
Google Scholar hat Beispiele zur Kollisionserkennung mit z.B. Raspberry Pi

Kollisionsvermeidung
Erkennen von physikalischen Grenzfällen
Identifizieren von Störgrößen
Aufbau eines mathematischen Schiffsmodells
Auswahl von geeigneten Reglertypen
Berechnung von Ausweichkursen zur Bahnplanung

Reichweite & Missionsoptimierung
12 Stundeneinsatz über einen Tidenzyklus (Nutzen von Ebbe und Flut)
Lastprofile der Elektrik
Geschwindigkeitsprofile zum Missionspunkt
Datenanbindung, evtl. GSM/LTE-Netz nutzen

Seetauglichkeit
Was hält das Boot bisher aus, Stabilität gegen Wellen
Optimierung des Bootskörpers
Stealthproblematik: Anbringen von Signallichtern
Wie kann man auf sich aufmerksam machen bei Kollisionsbehandlung
Verfahren zur Wasserung optimieren
Kalibieren des Kompass
Usability des Boots (Vorbereitung / Nachbereitung / Wartung)
(LARS) Launch and Recovery System

konkret Personen zu den Gruppen zugeordnet. Der Gruppe Seetauglichkeit wird als optionales Ziel die Sensormessung zugeordnet.

Die Themen werden in einem evtl. verlängerten Sprint durchgeführt, bisher ist der 5. Januar 2016 als Sprintende geplant. Bewertungskriterien, Nachteile und Themen werden in der Projektvision angepasst, bzw. Begründung

Sensormessung
Sensorkorb, Anbringort der Sensoren, Moonpool
Sensoren in Betrieb nehmen
Filmen der Wasseroberflächen und Wolken
Flaschensystem zur Probenaufnahme

Schwerpunkt	*Gruppenmitglieder*
Seetauglichkeit	Henning, Eike
Software Usability	Dennis
Kollisionserkennung	Max, Björn, Konstantin
Kollisionsvermeidung	Zahra, Peter, Michael

wieso Themen nicht genommen wurden werden hinzugefügt. Max spricht Prof. Hahn an, ob die Projektvision angepasst oder korrigiert werden sollen.

B.1.12 Gruppensitzung am 15. Dezember 2015

Moderator: *Peter Tank*
Protokollant: *Michael Beering*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Peter begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Festlegen der Projektsprache Es stellte sich die Frage, ob die Projektsprache innerhalb der wöchentlichen Sitzungen auf Englisch festgelegt wird. Hierfür spricht, dass Mohammad den Sitzungen folgen könnte und die Mitglieder der Projektgruppe ihre Englischfähigkeiten verbessern könnten.

Nachteile sind, dass Zarah kein Englisch spricht, es aber verstehen kann. Des Weiteren würde das Erheben der Anforderungen durch die Eingewöhnungszeit möglicherweise ins Stocken geraten.

Es wurde beschlossen, dass die Projektsprache weiterhin deutsch bleibt. Eventuell wird dies zu einem späteren Zeitpunkt geändert.

TOP 3 – Stand der Anforderungserhebung Hier entstanden einige Fragen bzgl. der Seetauglichkeit. Es ist nicht notwendig das gesamte Boot noch einmal in allen Details zu erklären. Es ist ausreichend die für die Anforderungserhebung nötigen Dinge zu erklären.

TOP 4 – Jira Um einen in Jira bearbeiteten Task in resolved zu verschieben, ist es zwingend erforderlich diesen ausreichend zu dokumentieren. Die

Dokumentation hat dabei so zu erfolgen, dass der geschriebene Text direkt in den Zwischen- bzw. Abschlussbericht eingefügt werden kann.

Jeder Commit in Git wird ab sofort mit der entsprechenden Jira-Task-Nummer versehen. In Sourcetree besteht dann die Möglichkeit über Links direkt auf die entsprechende Jira-Seite zu gelangen. Für die Einrichtung in Sourcetree sind einige Einstellungen nötig. Hierfür schickt Dennis einige Screenshots per Slack rum.

TOP 5 – Mitglieder der vorherigen Projektgruppen Einige Mitglieder aus alten Projektgruppen haben noch immer einen Schlüssel für den Raum der Projektgruppe. Um alle Materialen sicher verstauen zu können, ist ein weiterer Schrank notwendig. Den Schlüssel dafür organisiert Sascha bis zum Anfang des neuen Jahres.

Die Schlüssel der alten Teilnehmer werden vorerst nicht für den Raum gesperrt. Weiterhin gilt für die gesamte Projektgruppe: Nach dem praktischen Arbeiten werden alle Werkzeuge wieder in den Schränken verstaut und der Raum wird ordentlich verlassen.

TOP 6 – Inventur Die Inventur dient dem Zweck alle nicht mehr benötigten Dingen im Gruppenraum zu entsorgen und einen Überblick über alle Materialen zu erhalten. Durchgeführt wird die Inventur an einem Donnerstag im nächsten Jahr. Hierzu sind alle Teammitglieder eingeladen und es werden Kaltgetränke serviert.

B.1.13 Gruppensitzung am 5. Januar 2016

Moderator: *Michael Beering*
Protokollant: *Peter Tank*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Michael begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Anforderungserhebung Es werden die Anforderungserhebungen der einzelnen Gruppen besprochen. Sascha schlägt vor dass die Anforderung erst nach der Zusammenführung ausformulieren. Die Form der Ausarbeitung sollte vereinheitlicht werden. Als Grundlage hierfür soll das Dokument aus Requirements Engineering gelten. diese Woche Donnerstag werden die Ausarbeitungen Seetauglichkeit und Software bearbeitet.

Software Usability Dennis stellt seine Anwendungsfälle vor und definiert daraus die Anforderungen. Die Formulierung „darf“ sollte nicht verwendet werden, da die Bedeutung nicht eindeutig definiert ist.

Kollisionsvermeidung Michael erklärt die Herangehensweise und erläutert die Anforderungen. Das Nomoto-Modell sollte als mathematisches Modell in Betracht gezogen werden.

Seetauglichkeit Eike und Henning stellen ihre Ausarbeitung vor.

Kollisionserkennung Konstantin und Max stellen die Ausarbeitung vor.

TOP 3 – Sprint Review Der Task der Ausformulierung wird in den nächsten Sprint übertragen. Die nächsten beiden Donnerstage sind als Termine zur Bearbeitung festgelegt worden. Der Status der bisherigen Tasks wird besprochen. Es werden die Inventur, die Seminarausarbeitungen und die Anforderungen als Tasks in den nächsten Sprint gestellt.

TOP 4 – Sprint Retrospektive Der bisherige Sprint wird aufgrund der Feiertage als negativ bewertet, da über die Feiertage weniger geschafft wurde als gewünscht.

TOP 5 – Sprint Planning Der folgende Sprint soll drei Wochen dauern. Sascha möchte eine Planung/Formulierung bzgl. der Meilensteine, dieses wird als Task hinterlegt. Es soll ein Gantt-Chart angelegt werden. Sascha wird von Dennis bzgl. Lizenzen für ein UML-Tool befragt. Des Weiteren fragt Dennis nochmals nach dem Abgabetermin für den Zwischenbericht. Ende des Sprints ist der 26. Januar 2016.

TOP 6 – Sonstiges Die Seminararbeiten werden nicht durch andere Gruppenmitglieder korrigiert. Für andere Texte werden Lektoren nach Bedarf bestimmt. Die Lektoren sollen nicht nur den Text, sondern auch den LaTeX-Code korrigieren. Der Termin für Donnerstag wird auf 18:00 Uhr gesetzt. Es wird demnächst eine Liste mit Begrifflichkeiten festgelegt.

B.1.14 Gruppensitzung am 14. Januar 2016

Moderator: *Konstantin Gebel*
Protokollant: *Eike Hagena*
Abwesende Gruppenmitglieder: *Peter Tank*

TOP 1 – Begrüßung Konstantin begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Einigung auf ein UML-Tool Als UML-Tools dürfen sowohl Visual Paradigm, als auch der Rational Software Architect verwendet werden. Die Lizenzen, welche von der Uni Oldenburg stammen, hat Konstantin besorgt. Eine Anleitung für die Installation befindet sich im Wiki unter den Punkt „HowTos“. Die Visual Paradigm Diagramme werden am Ende in RSA übertragen. Die Lizenzen für RSA besitzt Dennis.

TOP 3 – Planung der Meilensteine Das Projekt soll am 30. September 2016 von uns fertiggestellt werden. Bis dahin werden wir unser Projekt in vier Meilensteine aufteilen. Die jeweiligen Meilensteine sind in drei Sprints aufgegliedert, wobei ein Sprint drei Wochen dauert. In den Sprints werden dann die vorher besprochenen Aufgaben bearbeitet. Ein Beispiel für die Planung der Meilensteine ist in Abb. B.2 zu finden.

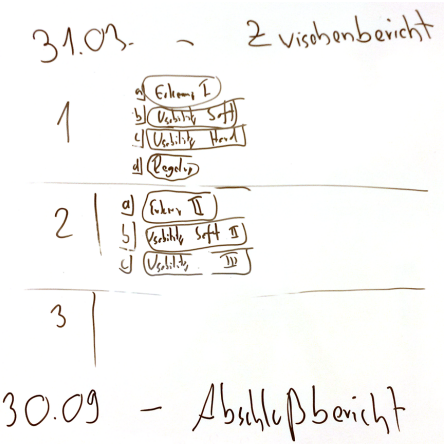


Abbildung B.2: Identifizierte Meilensteine

Die von uns vorgenommen Arbeitspakete für die Meilensteine und Sprints sind in der Abb. B.3 enthalten.

TOP 4 – Weiteres Vorgehen Die aufgestellten Anforderungen müssen nach den in Abb. B.4 dargestellten Konventionen benannt werden.

Ein Beispiel für eine funktionale Anforderung an die Software wäre:

F-SW-1.1.1

Für diesen Sprint müssen alle ihre Seminarausarbeitungen fertigstellen und in den Zwischenbericht einfügen. Außerdem müssen die Konventionen für die Anforderungen angepasst, die doppelten Anforderungen beseitigt und das Scrum verschriftlicht werden. Die geplante Inventur wird am nächsten Donnerstag, den 28. Januar 2016, durchgeführt.

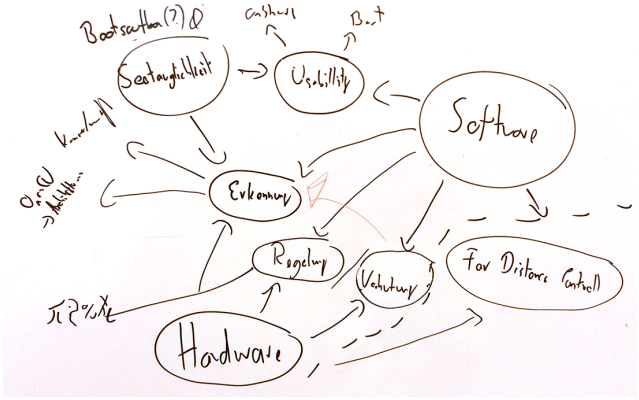


Abbildung B.3: Mögliche Arbeitspakete

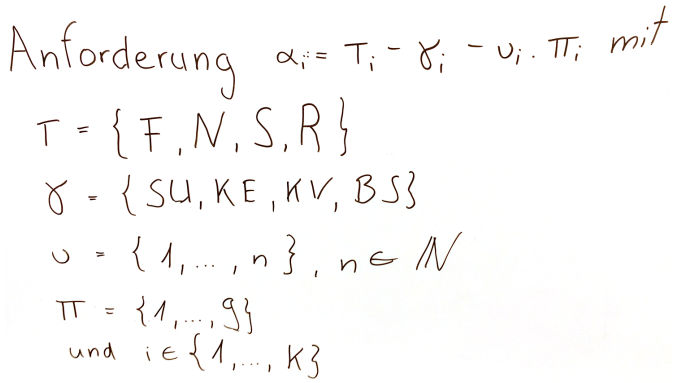


Abbildung B.4: Konventionen für Anforderungsbezeichner

B.1.15 Gruppensitzung am 19. Januar 2016

Moderator: *Eike Hagena*
Protokollant: *Björn Koopmann*
Abwesende Gruppenmitglieder: *Konstantin Gebel (bis 10:30 Uhr), Maximilian Hipp*

TOP 1 – Begrüßung Eike begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Vorstellung des Projektplans Eike präsentiert die bisherige Version des Projektplans und fasst die wesentlichen Inhalte der vier Meilensteine zusammen. Die Feingliederung wird in einem gesonderten Tagesordnungspunkt diskutiert.

Sascha bittet darum, ihm die finale Version des Projektplans über das Wiki zur Verfügung zu stellen. Zur Vereinfachung der Kommunikation mit Mohammad wird vereinbart, die Meilensteine neben deutsch- auch mit englischsprachigen Bezeichnungen zu versehen.

TOP 3 – Modellierung in RSA Dennis stellt die Verzeichnisstruktur für das Ablegen der im *Rational Software Architect* modellierten UML-Diagramme vor und erklärt den Workflow zum Anlegen eines neuen Diagramms. Björn bittet darum, die Vorgehensweise auf einer gesonderten Wiki-Seite zu beschreiben.

TOP 4 – Verfeinerung des Projektplans Dennis moderiert die Planung des ersten Meilensteins. Im Verlauf des Gesprächs werden die vier Arbeitsgruppen *Kollisionserkennung I*, *Usability Software I*, *Usability Hardware I* und *Regelung I* mit Inhalten versehen und erste Zieldefinitionen erarbeitet.

Kollisionserkennung I
(1) Abgrenzung von der Systemumgebung (2) Herleiten der funktionalen Architektur (3) Definition von Schnittstellen (4) Festlegung von Rahmenbedingungen (5) Treffen zentraler Designentscheidungen - Sensoren/Kameras (inkl. Auflösung, Schnittstellen, Preis) - Konzept zur Anbringung der Sensoren - Formulieren von Safety Goals und Sicherheitsanforderungen - Vorüberlegungen zur technischen Architektur - Ausblick auf weitere Realisierung
Verantwortliche: *Konstantin, Björn, Max*

Software I
(1) Validierung des IST-Zustands der Software - Trockenaufbau - Auslesen und Dokumentieren - Überprüfen verschiedener Testfälle - Software-Debugging (2) Umstellung des Build-Systems (Ant auf Maven) (3) Vorbereiten der Versionierung (4) Parsen von Kartendaten zur Gefahrenzonen-Detektion

(5) Parametrisierung der Hardware-Simulation (Wind, Strömung)
Verantwortliche: *Dennis, Michael*

Hardware I
(1) Validierung des IST-Zustands der Hardware - Trockenaufbau - Messen und Dokumentieren - Überprüfen verschiedener Testfälle (2) Überarbeiten und Vervollständigen der existierenden Handbücher (3) Erarbeiten eines Konzepts zur Sensoranbringung (Moonpool o.ä.)
Verantwortliche: *Henning, Eike*

Regelung I
(1) Evaluierung des Reglertyps (2) Definition eines geeigneten mathematischen Modells (3) Aufbau eines ersten Regelkreises (ohne Übertragungsfunktionen)
Verantwortliche: *Peter, Zahra*

B.1.16 Gruppensitzung am 26. Januar 2016

Moderator: *Maximilian Hipp*
Protokollant: *Björn Koopmann*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Max begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Abgabe des Zwischenberichts Die Abgabe des Zwischenberichts soll am 4. März 2016 bis 12:00 Uhr in digitaler Form erfolgen. Sascha verweist darauf, dass die Gliederung des Berichts an die der letzten Projektgruppe angelehnt werden kann. Der aktuelle Stand soll dargestellt und ein Ausblick auf das kommende Semester gegeben werden.

TOP 3 – Ergebnis der Inventur Max berichtet von der am 21. Januar durchgeführten Inventur. Peter erklärt sich bereit, das Amt des Inventurbeauftragten zu übernehmen und die Inventarliste fortlaufend zu aktualisieren.

TOP 4 – Stand der Anforderungserhebung Björn und Max stellen die Ergebnisse der Überarbeitung der Anforderungen vor. Mit Ausnahme der Sicherheitsanforderungen sind alle Anforderungen vollständig erfasst und können im Zwischenbericht eingesehen werden. Die korrekte Nummerierung der einzelnen Sätze wird in den kommenden Wochen nachgetragen.

TOP 5 – Sprint Review Im Verlauf des aktuellen Sprints konnte die Anforderungserhebung vollständig abgeschlossen werden. Alle Seminarausarbeitungen wurden in den Zwischenbericht übertragen. Die Beschreibung der Projektorganisation und die Erhebung der Sicherheitsanforderungen sind weiterhin ausstehend. Sie sollen in einem der nächsten Sprints durchgeführt werden.

TOP 6 – Sprint Retrospektive Michael lobt das konstante Vorantreiben der Dokumentation. Konstantin bezeichnet die ausformulierten Anforderungen als gute Basis für die weitere Arbeit. Während Henning das Arbeitsklima als sehr angenehm empfindet und die produktive Zusammenarbeit lobt, zeigt sich Eike von der gleichmäßigen Arbeitsverteilung beeindruckt und verweist auf die Vielzahl der abgeschlossenen Tasks. Björn bittet erneut darum, sich bei dem Schreiben von Texten für die Dokumentation an den zu Beginn der Projektgruppe vereinbarten Richtlinien zu orientieren. Max stimmt zu und schlägt vor, das gesamte Team durch gezielte Arbeitsanweisungen in die Optimierung des Zwischenberichts einzubinden.

TOP 7 – Planung des nächsten Sprints Dennis stellt die vorbereiteten Tasks für den ersten Meilenstein vor. Die anwesenden Gruppenmitglieder führen eine Grobplanung für die nächsten Wochen durch und wählen die zu bearbeitenden Tasks für den nächsten Sprint aus.

TOP 8 – Verschiedenes Björn schlägt vor, den Donnerstagstermin in den Ferien auszusetzen. Der Vorschlag wird einstimmig angenommen. Die reguläre Gruppensitzung am Dienstag wird weiterhin stattfinden.

B.1.17 Gruppensitzung am 2. Februar 2016

Moderator: *Björn Koopmann*
Protokollant: *Peter Tank*
Abwesende Gruppenmitglieder: *Henning Lawatsch*

TOP 1 – Begrüßung Björn begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Ausblick zur Betreuungssituation Sascha ist bis zum 31. März 16 an der Uni angestellt. Sein Nachfolger wird die Betreuung der Projektgruppe übernehmen, in der Zwischenzeit steht Mohammad als Ansprechpartner zur Verfügung.

TOP 3 – Hochschulinformationstag Am 3. Juni 2016 findet der Hochschulinformationstag statt. Es ist eventuell eine Ausstellung geplant. Einige Wochen vorher wird die Planung durchgeführt.

TOP 4 – Kurzberichte der Spezialistengruppen Es gibt vier Spezialistengruppen. Aufgrund der hohen Arbeitsbelastung durch die Klausurenphase werden die Kurzbericht der Teilgruppen auf die nächste Sitzung verschoben.

TOP 5 – Formulieren von Sicherheitsanforderungen Björn stellt die in Abb. B.5 dargestellte Systemarchitektur vor. Es wird eine zentrale Sicherheitseinrichtung geben, welche das System zurücksetzen und im Notfall die Position an die Onshore-Software senden kann. Des Weiteren soll die Batterie überwacht und die Daten dazu bereitgestellt werden. Die Resetleitungen zu den Komponenten sollten Low-Aktiv sein.

TOP 6 – Verschiedenes Björn wird Sascha eine Liste der ausformulierten Meilensteine per E-Mail zukommen lassen.

B.1.18 Gruppensitzung am 9. Februar 2016

Moderator: *Henning Lawatsch*
Protokollant: *Zahra Paya*
Abwesende Gruppenmitglieder: *Peter Tank, Maximilian Hipp*

TOP 1 – Begrüßung Henning begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Review des Einleitungstextes Der Einleitungstext für den Zwischenbericht wurde von Henning und Eike vorgestellt. Im Laufe der Zeit soll dieser noch um einen Ausblick auf das folgende Dokument hinzugefügt werden.

TOP 3 – Verschiedenes Prof. Dr. Hahn hat die Projektgruppe eingeladen am 22. Februar ein Forschungsschiff in Berne zu besichtigen. Dieses wird verwendet um geophysikalische und hydrographische Untersuchungen im Meer durchzuführen.

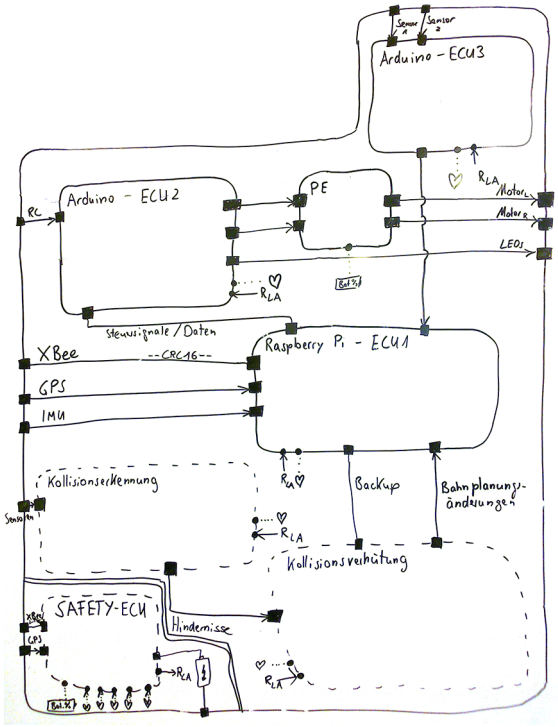


Abbildung B.5: Entwurf der technischen Systemarchitektur

B.1.19 Gruppensitzung am 16. Februar 2016

Moderator: *Zahra Paya*
Protokollant: *Dennis Pilny*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Zahra begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Sprint Review

Regelung: Grundlagen diverser Regelungsmethodiken wurden erarbeitet (PID-Regler, Neuronale Netze, ...), diese werden mithilfe von mathematischen Modellen in Matlab Simulink modelliert und evaluiert.

Bootsaufbau: Die bestehenden Handbücher wurden validiert, im Folgenden wird der IST-Zustand des Bootsaufbaus überprüft.

Kollisionserkennung: Eine grobe Systemabgrenzung wurde durchgeführt. Diese beinhaltet einen modularen Aufbau der Kollisionserkennung, so-

wie benachbarte Komponenten und die Schnittstellen zu denen. Sie dient als Grundlage für die detailliertere Spezifikation der Architektur.

Software: Der IST-Zustand der Onboard- und Onshore-Software wurde erhoben, dabei wurden einige Mängel an der Bedienbarkeit des Systems festgestellt. Für die Umstellung auf das Maven-Buildsystem wurde ein Nexus Repository auf dem Server eingerichtet. Zusätzlich wurden bereits Teile des Sourcecodes auf das Maven-Buildsystem umgestellt.

TOP 3 – Sprint Retrospektive

Positiv	*Negativ*
Das bereits vorhandene System konnte getestet werden	Aufgrund der Klausurenphase wurde wenig gemacht.

TOP 4 – Verschiedenes Für Phasen in denen aufgrund von Klausuren oder anderen Unitätigkeiten keine Zeit für die Arbeit an der Projektgruppe möglich ist, soll Urlaub genommen werden. Die weitere Meilensteinplanung soll zeitnah fertigestellt werden.

B.1.20 Gruppensitzung am 23. Februar 2016

Moderator: *Maximilian Hipp*
Protokollant: *Henning Lawatsch*
Abwesende Gruppenmitglieder: *Dennis Pilny*

TOP 1 – Begrüßung Max begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Review des bisherigen Zwischenberichts

- **Captions** ohne Punkt am Ende.
- **Bilder und Tabellen** müssen auf einheitliche Größe angepasst werden.
- **Quellen** überarbeiten!
- Das **Deckblatt** wird von Björn angepasst.
- Das **Inhaltsverzeichnis** ist soweit in Ordnung.
- Das **Abbildungsverzeichnis** wird von Eike überarbeitet (Quellen entfernen).
- Das **Tabellenverzeichnis und Referenzen** werden von Björn überarbeitet
- In der **Einführung** müssen einige Formalitäten korrigiert werden

- **Projektorganisation:** Der 2. und 3. Meilenstein muss noch überprüft werden. Die Umbrüche müssen optimiert werden.
- **Rollenverteilung:** In Ordnung, aber teilweise Korrektur notwendig
- **Infrastruktur:** In Ordnung.
- **Projektvision und Stakeholder:** In Ordnung.
- **Rahmenbedingungen:** Kollisionserkennung fehlt, Bei Bootsaufbau müssen Referenzen und Tabelle(keine caption) überarbeitet werden.
- **Anforderungen:** Es fehlen einige Fülltexte und die Nummerierung.
- **Sicherheitsanforderungen** wurden für den Zwischenbericht entfernt.
- **Aktuell eingesetztes System:** Einige Tippfehler vorhanden.
- **Theoretische Grundlagen:** Noch offen.
- **Konzept:** Jede Gruppe sollte zum Zwischenbericht einige Artikel schreiben.
- **Realisierung:** Noch offen.
- **Evaluierung:** Fällt weg.
- **Fazit:** Es muss noch ein Ausblick geschrieben werden.
- **Seminarausarbeitungen:** Kleinigkeiten müssen korrigiert werden (und Literatur).
- **Protokolle:** Korrektur von Kleinigkeiten.
- **Anhang C** eventuell auskommentieren und **Anhang D** rausschmeißen.
- **Glossar:** Jeder sollte aus seinen texten Definitionen einfügen.
- **Literaturverzeichnis:** Muss grundlegend überarbeitet werden und jeder muss seine Quellen mit Informationen füttern. Dies sollte bis Dienstag, den 1. März, abgeschlossen sein.

TOP 3 – Verschiedenes Es werden nächste Woche und in Zukunft zusätzliche Treffen nach Absprache stattfinden. Diese finden bevorzugt Nachmittags und Abends statt. Für nächste Woche sind die Treffen täglich um 18 Uhr angesetzt. Die offizielle Sitzung fällt nächste Woche deshalb aus. Am Montag wird ein Trockentest um 10 Uhr durchgeführt (freiwillige Teilnahme).

B.1.21 Gruppensitzung am 8. März 2016

Moderator: *Peter Tank*
Protokollant: *Michael Beering*
Abwesende Gruppenmitglieder: *Konstantin Gebel, Maximilian Hipp*

TOP 1 – Begrüßung Peter begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Ergebnis Zwischenbericht/Planung Abschlussbericht Es wird darüber diskutiert, ob an der Vorgehensweise zum Verfassen des Zwischenberichtes irgendwas schlecht gelaufen ist. Hierbei wird lediglich von Peter angemerkt, dass es einen Datei zum Ablegen von Code-Beispielen geben sollte. Nach einer kurzen Diskussion wird beschlossen, Code-Ausschnitte vorerst in „Weitere Anhänge“ zwischenzuspeichern.

Sascha hat noch keine Anmerkungen zum Bericht, da er diesen erst im Laufe der Woche durcharbeiten wird. Im Ordner des Zwischenberichts werden ab sofort keine Änderungen mehr vorgenommen, da der Ordner Abschlussbericht bereits existiert. Alles Weitere wird in diesem Ordner hinzugefügt oder geändert.

TOP 3 – Sprint Review Während des Sprint Reviews berichten die Kleingruppen von ihren bisher erreichten Zielen, die im Zwischenbericht dokumentiert wurden.

TOP 4 – Sprint Retrospektive Die Sprint Retrospektive hat ergeben, dass während des Sprints ein sehr gut organisierter Zwischenbericht entstanden ist. Außerdem hat sich der Slack-Channel „Doku-Pranger“ als nützlich erwiesen, um Leute auf Fehler innerhalb der Dokumentation hinzuweisen. Der Zwischenbericht bietet eine gute Grundlage für den Abschlussbericht und hat den Gruppenmitgliedern einen detaillierten Einstieg in das Projekt ermöglicht.

Negativ am vergangenen Sprint ist angemerkt worden, dass zwar viel Text produziert, allerdings noch wenig Greifbares implementiert wurde. Dies soll in den kommenden Wochen geändert werden. Der Ausfall der Gruppensitzung in der vergangenen „Doku-Woche“ hat sich als nicht zweckmäßig erwiesen. Daher sollen in der Zukunft keine weitere Gruppensitzungen ausfallen. Des Weiteren hat es beim Hochladen von Änderungen in das Git-Repository einige Probleme gegeben. Dies soll durch eine erhöhte Aufmerksamkeit der Gruppenmitglieder in Zukunft nicht erneut vorkommen. Außerdem kam es bei dem Einpflegen der Quellen in den Zwischenbericht zu einigen Missverständnissen, die auf unzureichende Absprache zurückzuführen sind.

TOP 5 – Termin- und Personalsuche für den MOPS-Frühjahrsputz Der Frühjahrsputz wird möglichst von allen Gruppenmitgliedern durchgeführt. Dennis wird zur Terminfindung eine Doodle-Umfrage erstellen. Zweck ist neben dem Reinigen des Wasserfahrzeugs auch das Anbringen von Schellen an die Batterie sowie das Herstellen eines geeigneten Ladekabels.

TOP 6 – Terminsuche für einen Praxistest am Tweelbäker See Der erste Praxistest soll Anfang April stattfinden. Vorzugsweise soll dieser innerhalb der Woche liegen, da Sascha oder Prof. Hahn anwesend sein müssen. Der Praxistest dient neben einem Test der Komponenten auch dem Sammeln von Testdaten für die Kollisionserkennung. Außerdem sollen nach Möglichkeit einige Tests bzgl. der Regelung durchgeführt werden. Die Terminfindung wird durch einer von Dennis erstellten Doodle-Umfrage organisiert.

TOP 7 – Besprechung der funktionalen Architektur Da beim Erstellen der funktionalen und technischen Architektur nicht alle Gruppenmitglieder anwesend waren, wird diese von Björn vorgestellt.

TOP 8 – Sprint Planning Der kommende Sprint läuft in der Zeit vom 8. bis zum 29. März. Die durchzuführenden Aufgaben können aus dem Agile-Board entnommen werden.

TOP 9 – Verschiedenes Im Sourcecode werden momentan sowohl Tabs als auch Leerzeichen für die syntaktische Einrückung von Codeteilen genutzt. Ab sofort werden hierfür nur noch Tabs verwendet.

Dennis hat die Einrichtung von Maven-Projekten in IntelliJ vorgestellt. Eine Anleitung findet sich außerdem im Doku-Wiki.

B.1.22 Gruppensitzung am 15. März 2016

Moderator: *Michael Beering*
Protokollant: *Konstantin Gebel*
Abwesende Gruppenmitglieder: *Peter Tank*

TOP 1 – Begrüßung Michael begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Stand des aktuellen Sprints

Kollisionserkennung

- Die Gruppe Kollisionserkennung hat sich mit OpenCV auseinandergesetzt und erste Algorithmen zur Bildverarbeitung und Objekterkennung ausprobiert.

- Die Gruppe Kollisionserkennung hat Vorüberlegung zu möglichen Technologien und deren Anbringungen am Boot angestellt.

Software

- Die Gruppe Software hat das alte Logsystem auf JLog umgestellt. Bei Exceptions kann nun einfach an die entsprechende Stelle im Code verwiesen werden.
- Die Gruppe Software hat Anpassungen an der Struktur des Codes vorgenommen.

Bootsaufbau

- Die Gruppe Bootsaufbau hat sich mit damit befasst, wie die Platinen am Boot befestigt werden können. Dabei kamen die Ideen auf, auf Schlaufen, Magnetband, Verschrauben oder Schienen zurückzugreifen.
- Die Gruppe Bootsaufbau hat sich Gedanken zur Verkabelung gemacht.

Sonstige Tasks

- Konstantin hat Wordpress auf dem Server installiert und erste Inhalte zum Webauftritt hinzugefügt.

TOP 3 – Technologieentscheidungen

- Maximilian stellt anhand von Beispielbildern die Funktionsweise von OpenCV vor.
- Die Gruppe einigt sich darauf, vorerst auf Kameras zur Erkennung von Kollisionen zurückzugreifen.
- Entscheidungen bezüglich einer Bootsabdeckung und des Moonpools werden auf den nächsten Sprint verschoben.

TOP 4 – Termin für den Praxistest Die Gruppe einigt sich darauf, den ersten Praxistest am Tweelbäker See am 7. oder 14. April 2016 durchzuführen.

B.1.23 Gruppensitzung am 22. März 2016

Moderator: *Konstantin Gebel*
Protokollant: *Henning Lawatsch*
Abwesende Gruppenmitglieder: *Peter Tank, Eike Hagena, Zahra Paya, Björn Koopmann, Maximilian Hipp*

TOP 1 – Begrüßung Konstantin begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Stand des aktuellen Sprints Die Software wurde grundlegend überarbeitet und das Zeitverhalten der einzelnen Komponenten optimiert. Durch das Herstellen eines neuen Ladekabels wurde das Aufladen der Batterie vereinfacht.

TOP 3 – Vorbereitung des Praxistests Vorher müssen folgende Dinge geklärt/ausgeführt werden:

- Aufbauen
- Trockentest durchführen
- Kamera für Aufnahmen organisieren
- Anhänger
- Werkzeug mitnehmen
- Verlängerungskabel (vom Yacht-Club?)
- USB-Hub muss besorgt sein

TOP 4 – Evaluation der Regelung Es muss noch die richtige Methode für die Evaluation gefunden werden.

B.1.24 Gruppensitzung am 29. März 2016

Moderator: *Maximilian Hipp*
Protokollant: *Dennis Pilny*
Abwesende Gruppenmitglieder: *Henning Lawatsch, Eike Hagena, Björn Koopmann, Zahra Paya, Konstantin Gebel*

TOP 1 – Begrüßung Maximilian begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Nächster Praxistest

- Der Termin für die Fahrt an den Tweelbäker See ist noch nicht sicher, es ist jedoch sehr wahrscheinlich das der Praxistest am 7. April stattfinden kann.
- Es ist nötig, im Vorfeld einen vollständigen Trockentest durchzuführen, bei dem alle zu beobachtenden Funktionen am See vorher getestet werden müssen.

- Michael hat die Möglichkeit, ein Auto und einen Anhänger für den Transport des Bootes zu besorgen.
- Dennis bringt eine Kabeltrommel sowie eine Steckerleiste mit.
- Benötigtes Werkzeug wird aus dem PG-Raum entnommen, gegebenenfalls bringt Michael zusätzlich benötigtes Werkzeug mit.
- Sascha hat es aufgrund der Urlaubszeit nicht geschafft eine Kamera zu organisieren. Es besteht die Möglichkeit, dass Michael eine Webcam mitbringt oder Konstantin eine RaspberryPi-Kamera.

TOP 3 – Sprint Review

Kollisionserkennung

- Es wurden diverse Algorithmen in OpenCV zur Objekterkennung mit Bildern ausgetestet. Diese liefen bisher jedoch nur auf lokalen Maschinen. Im nächsten Schritt sollen diese auf einem RaspberryPi durchgeführt werden.
- Es wurden zwei Kameramodelle ausgesucht, welche zur Kollisionserkennung verwendet werden können.

Software

- Abgearbeitet Missionspunkte werden in der Onshore-Software nun visuell dargestellt.
- Die XBee-Schnittstelle in der Onshore-Software kann nun konfiguriert werden.
- Log4j2 wurde als Standardlogger implementiert. Dies ermöglicht das Loggen spezifischer Events in seperaten Dateien.
- GPS-Daten werden in einer separaten Datei gespeichert. Dies soll der Evaluation für die Regler dienen.
- Die Generierung von Gefahrenzonen innerhalb der Onshore-Software ist zu einigen Teilen fertig.

Bootsaufbau

- Ein Not-Aus Kreis wurde erstellt, dieser schaltet das Gesamtsystem ab, wenn ein Schalter betätigt wurde.
- Die Verkabelung der Komponenten ist umständlich.

Regelung

- Alle Vorbereitung für die Evaluation/den Aufbau der neuen Reglerstruktur wurden durchgeführt. Nun fehlen Daten, die bei der Fahrt zum Tweelbäker See aufgenommen werden sollen.

TOP 4 – Sprint Retrospektive

Positiv	*Negativ*
Arbeiten an der Onshore-Software waren sehr effektiv.	Aufgrund der Urlaubsphase wurde wenig gemacht.

TOP 5 – Verschiedenes

- Die Not-Aus-Funktion könnte so implementiert werden, dass nur die Motoren bei der Betätigung ausgestellt werden. Dies würde die weitere Kommunikation mit der Onshore-Software ermöglichen.
- Eine Aufgabe für zukünftige Projektgruppen könnte die Implementierung einer Schalttafel sein, die alle nötigen Schalter beinhaltet und Informationen zum Zustand des Bootes beinhaltet.
- Für das kommende Sommersemester muss ein neuer Termin für die regelmäßigen Treffen der Gruppe gefunden werden, da sich der aktuelle Termin mit Vorlesungen überschneidet.

B.1.25 Gruppensitzung am 6. April 2016

Moderator: *Dennix Pilny*
Protokollant: *Michael Beering*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Dennis begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Terminfindung Der Sitzungstermin für das Sommersemester 2016 ist Mittwochs um 16:15 Uhr.

TOP 3 – Vorbereitungen Praxistest

- Es hat sich gezeigt, dass die Kommunikation zwischen Raspberry Pi und Motor-Arduino derzeit nicht funktioniert. Dies ist für den Praxistest nicht relevant.
- Die Inventurliste wird an Sascha geschickt.
- Der Praxistest beginnt um 8:30 Uhr beim im Projektgruppen-Raum. Michael bringt seinen Anhänger für den Transport mit.

TOP 4 – Ziele des Tests Die primären Ziele des ersten Praxistests bestehen darin, Bildmaterial für die Kollisionserkennung zu sammeln und Daten für die Regler-Evaluation aufzuzeichnen.

TOP 4 – Verschiedenes

- Max stellt ein Lidar-System als Alternative zur kamerabasierten Kollisionserkennung vor. Sascha prüft, ob finanzielle Mittel für ein solches System zur Verfügung gestellt werden können. Außerdem erkundigt sich Sascha, ob ein Laserscanner am OFFIS verfügbar ist und ob dieser für Tests verwendet werden kann.
- Die Erstattung des ausgelegten Geldes für Kabelbinder, Klebeband, Isolierband und Batterien wird über Manuela Wüstefeld organisiert. Die Kassenzettel werden von Sascha weitergereicht. Das Geld hierfür hat Michael ausgelegt.
- Ein Moonpool wird in dieser Projektgruppe nicht mehr verbaut. Hierfür wird höchstens ein Konzept erarbeitet. Die Rechtfertigung für das nicht Umsetzen des Moonpools wird von Eike und Henning notiert und in die Dokumentation eingearbeitet.
- Eike und Henning stellen ein Konzept für die Anbringung der Elektronikkomponenten in den Otterboxen vor. Hierfür sollen Holz und kleine Spanngurte eingesetzt werden.

B.1.26 Gruppensitzung am 13. April 2016

Moderator: *Michael Beering*
Protokollant: *Konstantin Gebel*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Michael begrüßt die anwesenden Gruppenmitglieder. Marius – unser zukünftiger Betreuer – stellt sich vor.

TOP 2 – Stand des aktuellen Sprints

Kollisionserkennung

- Die Gruppe Kollisionserkennung hat einzelne Videoausschnitte zu bestimmten Szenarien aus dem Praxistest gezogen und mit diversen Filtern bearbeitet.
- Die Gruppe Kollisionserkennung hat etwas zu dem Laserscanner aus dem OFFIS recherchiert. Es ist geplant diesen auszuleihen und Tests damit durchzuführen.

Software

- Die Gruppe Software hat das Parsen von Kartendaten realisiert und weitere Verbesserungen an der Codestruktur vorgenommen.

- Die Gruppe Software hat sich dazu entschlossen vorerst nicht weiter an der Software zu arbeiten, sondern an der Kollisionserkennung mitzuwirken.

Bootsaufbau

- Die Gruppe Bootsaufbau hat Schienen zur Befestigung der Platinen am Boot angebracht.
- Das elektrische Konzept soll überarbeitet werden. Peter hat die Probleme der jetzigen Lösung dargestellt und eine Mögliche Verbesserung vorgeschlagen. Dazu soll die Verkabelung erneuert und die Anordnung der Komponenten geändert werden. Dies soll im Endeffekt zu einer verbesserten Steuerung und Ausfallsicherheit führen.

Sonstige Tasks

- Björn hat am Sicherheitskonzept und der Übersicht der Systemkomponenten weitergearbeitet.

TOP 3 – Rückblick des Praxistests

- Für die Kollisionserkennung wurden mithilfe zweier Kameras Testdaten in Form von Videos aufgenommen. Diese zeigen verschiedene Szenarien auf dem See aus Sicht des Bootes.
- Weiterhin wurden Logdaten für die Regelung aufgenommen. Dazu wurden Geschwindigkeit, Position sowie ein Zeitstempel aufgezeichnet.
- Der Gesamteindruck vom Praxistest war gut.

TOP 4 – Schiffsaufbau und Elektronik

- Die bisherige Verteilung der Bootskomponenten ist nicht ideal. Sowohl der Schwerpunkt als auch der Drehpunkt verschulden eine unvorteilhafte Steuerung und Strömungsverhalten des Bootes.
- Die Leistungselektronik soll von den Steuerungskomponenten getrennt werden, um Spannungsinterferenzen zu minimieren oder auszuschließen. Es ist sogar in Überlegung eine eigene Stromversorgung für die Steuerungselektronik einzubinden.
- Es ist geplant die Elektronik bis zum nächsten Praxistest umzugestalten.

TOP 5 – Weiteres Vorgehen

- Es ist ein weiterer Praxistest in der zweiten Maiwoche geplant.
- Für den nächsten Praxistest ist geplant einem vorgeplantem Script mit Szenarien zu folgen.
- Es soll ein Laserscanner aus dem OFFIS organisiert werden, um dessen Tauglichkeit für die Kollisionserkennung zu erproben.

B.1.27 Gruppensitzung am 20. April 2016

Moderator: *Konstantin Gebel*
Protokollant: *Eike Hagena*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Konstantin begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Aktueller Stand Da Sascha nicht an der Sitzung teilnehmen konnte, haben wir diesen Tagespunkt verschoben.

TOP 3 – Schiffsdynamik und elektrisches System Alle anwesenden Gruppenmitglieder haben für den Umbau des Bootes zugestimmt. Auch die damit verbunden Investitionen wurden von allen Teilnehmern abgesegnet. Es fehlt lediglich die Zustimmung von Sascha. Laut Peter betragen die Kosten für die benötigten Teile um die 140,00 Euro.

TOP 4 – Nächster Praxistest Der zweite Praxistest findet wahrscheinlich in der zweiten Maiwoche am 9. Mai 2016 statt. Konstantin, Dennis und Max haben eine Liste mit möglichen Szenarien erstellt, die wir noch mit der Kamera aufnehmen sollten. Diese Liste kann über das Etherpad noch erweitert werden. Daher sollen sich alle Gruppenmitglieder noch mögliche Szenarien überlegen. Beim nächsten Praxistest sollen die Punkte auf der Liste dann abgearbeitet werden.

TOP 5 – Sprint Review Die Kleingruppen erzählen was sie in den letzten Wochen gemacht haben und welche Ziele erreicht wurden. Außerdem zeigt Michael ein Video mit entfernten Weitwinkel.

TOP 6 – Sprint Retrospektive Vor allem der erste Praxistest wurde sehr gelobt und als gut empfunden. Für den nächsten Praxistest soll aber ein Testprotokoll angefertigt werden. Bemängelt wurde, dass nicht ganz so viel geschafft wurde, wie wir uns vorgenommen haben. Das lag auch daran,

dass einige Entscheidungen viel Zeit benötigt haben. Dafür wurden aber auch Ziele umgesetzt, die für diesen Sprint nicht eingeplant waren. Dazu gehören der geplante Umbau des Bootes und die Besorgung der Laserkamera.

TOP 7 – Sprint Planning Die Tasks für den neuen Sprint können im Jira eingesehen werden. Der Sprint läuft in der Zeit vom 20. April bis zum 11. Mai 2016.

B.1.28 Gruppensitzung am 27. April 2016

Moderator: *Eike Hagena*
Protokollant: *Maximilian Hipp*
Abwesende Gruppenmitglieder: *Dennis Pilny*

TOP 1 – Begrüßung Eike begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Videosequenzen Eike und Henning haben die ausgewählten Sequenzen aus dem Rohmaterial geschnitten.

TOP 3 – Schiffsdynamik und elektrisches System Die Otterboxen sollen am Dienstag um 16:00 Uhr an ihre neuen Positionen versetzt werden. Peter lädt alle Gruppenmitglieder dazu ein, sich an den Umbaumaßnahmen zu beteiligen.

TOP 4 – Horizonterkennung und Laserscanner Die Horizontlinienerkennung von Michael kann, neben dem eigentlichen Zweck, zusätzlich als Backup-Lösung für die Schiffsneigung herangezogen werden. Der Laserscanner kann über eine IP-Adresse angesprochen werden. Diese kann bei Max erfragt werden. Im weiteren Schritt sollen die Daten des Laserscanners exportiert und anschließend erneut eingelesen werden können. Weiterhin sollen Anpassungen bei der Horizonterkennung vorgenommen werden, sodass die Uferbereiche zuverlässiger erkannt werden.

TOP 5 – Sonstiges Wenn etwas bestellt werden muss, soll ein möglichst bekannter Shop herausgesucht werden und an Sascha weitergeleitet werden. Weiterhin merkt Sascha an, dass mehr Ressourcen für die Horizonterkennung und den Laserscanner zugeteilt werden sollten.

B.1.29 Gruppensitzung am 4. Mai 2016

Moderator: *Maximilian Hipp*
Protokollant: *Björn Koopmann*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Max begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Schiffsdynamik und elektrisches System Die Teile wurden in der vergangenen Woche durch Sascha Hornauer bestellt. Bisher gibt es noch keinen bestätigten Liefertermin. Die Durchführung des Praxistests am kommenden Montag scheint fraglich.

TOP 3 – Sicherheitskonzept Björn präsentiert das von ihm erarbeitete Sicherheitskonzept. Einen inhaltlichen Einblick in die Thematik gebend, werden die einzelnen Kapitel schrittweise vorgestellt. Die Fertigstellung der theoretischen Betrachtungen wird in den nächsten Tagen erfolgen.

TOP 4 – Laserscanner Max berichtet von den durchgeführten Arbeiten zur Vorbereitung des Laserscanners. In einem selbst geschriebenen Tool können die Sensordaten aufgezeichnet und zu einem späteren Zeitpunkt abgerufen werden.

TOP 5 – Horizonterkennung Michael stellt die Ergebnisse der Arbeitsgruppe *Kollisionserkennung* vor. Die Algorithmen zur Horizonterkennung wurden weiter optimiert. Durch die Bestimmung und den Vergleich des Histogramms verschiedener Bildausschnitte könnte ein Verfahren zur Kollisionserkennung entwickelt werden.

TOP 6 – Verschiedenes Die anwesenden Gruppenmitglieder einigen sich darauf, den für den 9. Mai angesetzten Praxistest auf unbestimmte Zeit zu verschieben, da die Durchführung ohne die benötigten Bauteile wenig zielführend ist.

B.1.30 Gruppensitzung am 11. Mai 2016

Moderator: *Björn Koopmann*
Protokollant: *Henning Lawatsch*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Björn begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Schiffsdynamik und elektrisches System Mit dem Umbau wurde bereits begonnen. Noch nicht alle Teile wurden geliefert. Für die Plexiglasscheibe des Bedienpanels ist ein Rahmen notwendig.

TOP 3 – Kollisionserkennung und Sensorik Michael ist es gelungen den Himmel aus dem Videomaterial herauszuschneiden und die Reflexionen größtenteils entfernt. Der Laserscanner ist kurzzeitig unterwasserfähig. Beim Abholen des Laserscanners soll direkt gefragt werden, wie wir ihn einsetzen dürfen.

TOP 4 – Sprint Review Eine Entscheidung, welche Sensoren angewandt werden, wurde noch nicht getroffen. Im nächsten Sprint müssen daher konkrete Entscheidungen getroffen werden. Die Horizonterkennung ist größtenteils fertig, kann aber noch weiterentwickelt werden. Das Befassen mit dem Nomoto-Modells wird nach hinten verschoben. Das Sicherheitskonzept ist fertig und Björn hat ein Template für die Praxistests erstellt.

TOP 5 – Sprint Retrospektive Es war ein sehr produktiver Sprint in dem besonders die *Kollisionserkennung* vorangetrieben wurde.

TOP 6 – Planung des nächsten Sprints Für die *Kollisionserkennung* ist eine konkrete Idee für die Objekterkennung ist dringend erforderlich. Dazu sind alle Mitglieder gebeten, sich zu informieren, Literatur heranzuziehen und ggf. zu testen. Es werden ein neues Raspberry Pi und Hardware für die Safety-ECU benötigt. Wegen des Bootsumbaus ist eine Anpassung der Bedienungsanleitung notwendig. Einen Schaltplan hat Peter bereits erstellt. Das Thema Wasserfestigkeit wird zunächst nach hinten verschoben. Im Bedienanel sollen Status-LEDs angebracht werden.

TOP 7 – Vorbereitung des Praxistests Die einzelnen Gruppen sollen sich Testfälle überlegen. Der Praxistest wird voraussichtlich am 27. Mai ab 8:30 Uhr stattfinden.

TOP 8 – Vorbereitung des Praxistests Am 23. Juni findet das Boule-Turnier statt. Bis zur nächsten Sitzung sollen sich alle Gruppenmitglieder mögliche Teamnamen ausdenken.

B.1.31 Gruppensitzung am 18. Mai 2016

Moderator: *Henning Lawatsch*
Protokollant: *Zahra Paya*
Abwesende Gruppenmitglieder: *Maximilian Hipp*

TOP 1 – Begrüßung Henning begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Schiffsdynamik und elektrisches System Peter berichtet, dass die Bedienelemente bis jetzt nicht befestigt werden können, da zuerst die Verkabelung fertig sein soll. Der Steuerkreis fehlt zurzeit noch, soll aber spätestens am Samstag fertiggestellt werden.

TOP 3 – Kollisionserkennung Michael berichtet, dass es ein Histogrammbasierter Ansatz verfolgt wurde und viele Sachen mit OpenCV fertig sind. Es gibt Paper über Hintergrunderkennung, die hilfreich sein können. Dennis und Konstantin präsentieren die Ergebnisse ihrer Literaturrecherche. Die entsprechenden Dokumente sind im Git unter `trunk/Literatur/Kollisionserkennung` zu finden.

Marius fügt hinzu, dass als Methode zur Kollisionserkennung ein fertiger Algorithmus, das sogenannte „Grabcut"-Verfahren, verwendet werden könnte.

TOP 4 – Feinplanung Praxistest Dennis berichtet, dass beim Praxistest ein Plan abgearbeitet wird. Deswegen soll jede Gruppe über Testfälle nachdenken. Es sollen Daten geloggt, weitere Testaufnahme für die Kollisionserkennung gemacht und Daten für die Regelung aufgenommen werden. Die Szenarien sind im Git unter `trunk/Testfahrten` einsehbar.

Peter ist der Meinung, dass entweder am Montag oder am Dienstag alles ausprobiert und ein erster Test des vollständigen Systems durchgeführt werden kann, wenn es keine Probleme gibt.

Michael ist der Meinung, dass es für die Kameraaufnahme sehr wichtig ist, dass die Hindernisse größer als letztes mal sein sollen.

TOP 5 – Boule-Turnier Die Teilnehmer haben den Namen „Moptimus Prime" für das Boule-Team gewählt. Eike wird sich in die Spielregeln einarbeiten und den anderen Gruppenmitgliedern während der nächsten Sitzung einen kurzen Überblick geben.

TOP 6 – Verschiedenes Bei dem Thema Realisierung in der Dokumentation sind Konstantin und Michael der Meinung, dass es zu viele Unterpunkte gibt, was die Zuordnung der Inhalte schwierig macht.

Björn und Marius sind der Meinung, dass alle Themen, die zu den Grundlagen gehören, in das Kapitel *Theoretische Grundlagen* verschoben werden sollten. Dies gilt auch analog für die Kapitel *Konzept* und *Realisierung*.

B.1.32 Gruppensitzung am 25. Mai 2016

Moderator: *Zahra Paya*
Protokollant: *Dennis Pilny*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Zahra begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Kollisionserkennung Die OpenCV-Projekte wurden objektorientiert umstrukturiert. Dadurch soll es in Zukunft einfacher gemacht werden, verschiedene Algorithmen zu evaluieren.

Weiterhin wurde sich darauf geeinigt, eine Webcam für die Kollisionserkennung zu verwenden. Diese soll per USB angesteuert werden und somit einfach austauschbar sein. Verwendet werden soll eine Logitech C920HD. Über einen Polfilter könnten zusätzlich Reflexionen aus den Bildern der Kamera herausgefiltert werden, dies könnte Rechenzeit einsparen, wenn die Reflexionen nicht mehr über die Software herausgerechnet werden müssen.

TOP 3 – Nächster Praxistest Für die am 27. Mai geplante Testfahrt muss das Datenlogging optimiert werden. Eike übernimmt die Protokollierung des Praxistests. Für die Kollisionserkennung sollen weitere Szenarien aufgenommen und Aufnahmen mit einem Polfilter vorgenommen werden. Zusätzlich sollen Beschleunigungs und Kursdaten aufgenommen werden, die der Bestimmung der Reglerdaten dienen soll.

TOP 4 – Boule-Turnier Eike stellt die Spielregeln und wichtige Begriffe vor. Eine Zusammenfassung ist im Git-Repository zu finden.

TOP 5 – Verschiedenes Ein Kassenbuch wurde in das Git hochgeladen. Eike und Henning übernehmen die Aufgabe dies zu verwalten und tragen bereits getätigte Ausgaben nach.

B.1.33 Gruppensitzung am 1. Juni 2016

Moderator: *Dennis Pilny*
Protokollant: *Peter Tank*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Dennis begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Weiteres Vorgehen/Praxistest Der Praxistest ist auf den 13. Juni um 8:00 Uhr verschoben. Peter und Michael bringen das elektrische System in Ordnung. Dennis integriert den RPi3. Die Safety-ECU wird vorerst, aufgrund von Zeitmangel, aus dem Projektvorhaben gestrichen. Im Abschlussbericht soll für die folgenden Gruppen ein zeitlicher Ablaufplan integriert werden.

TOP 3 – Sprint Retrospektive Die Tasks wurden auf Vollständigkeit im JIRA überprüft und aktualisiert.

TOP 4 – Sprint Review Keine Auffälligkeiten.

TOP 5 – Sprint Planning Der Laserscanner wird mit einer kardanischen Aufhängung am Bootsrand angebracht. Die Scanneinheit soll dabei nach unten zeigen.

TOP 6 – Verschiedenes Sascha bittet darum, vor Bestellungen kurz informiert zu werden.

B.1.34 Gruppensitzung am 8. Juni 2016

Moderator: *Michael Beering*
Protokollant: *Eike Hagena*
Abwesende Gruppenmitglieder: *Peter Tank, Dennis Pilny, Konstantin Gebel*

TOP 1 – Begrüßung Michael begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Weiteres Vorgehen/Praxistest Die Hardwarekomponenten vom Boot wurden von Michael getestet und funktionieren so weit. Lediglich mit dem Arduino gibt es noch ein kleines Problem, welches aber bis zur Testfahrt behoben sein sollte.

Die von uns überlegten Testfelder sollen abgearbeitet werden, etwa das selbständige Abfahren einer geplanten Strecke. Auch möchten wir die Regelung der alten Projektgruppe mit unserer Regelung vergleichen. Außerdem soll der Laserscanner und die Kamera möglichst über den Raspberry Pi laufen. Treffen für den Praxistest am 12. Juni um 8:00 Uhr.

TOP 3 – Elektrisches System/Software Die elektrischen Komponenten wurden komplett aus dem Boot geholt und nochmal komplett überarbeitet. Dadurch wurde erreicht, dass die Stromversorgung jetzt stabil läuft.

TOP 4 – Bootsaufbau: Status Kamera-/Laserscannerhalter Die Halterung für die Kamera soll über einen 3D-Druck realisiert werden. Für den Laserscanner gibt es bereits eine Halterung, Marius möchte sich hierüber erkundigen.

TOP 5 – Kollisionserkennung: Bildverarbeitung/Laserscanner Björn gibt einen Ausblick über die Filteroperationen und möchte diese in den nächsten Tagen testen. Darüber hinaus hat Michael einen Frame-Controller geschrieben.

TOP 6 – Verschiedenes Marius stellt ein paar Fragen bezüglich der Objekterkennung und gibt Tipps für mögliche Lösungen. Über die Vorschläge von Marius wird innerhalb der Gruppe diskutiert. Außerdem erläutert die Gruppe die Probleme die vor allem bezüglich der Reflexionen auf der Wasseroberfläche auftreten.

B.1.35 Gruppensitzung am 15. Juni 2016

Moderator: *Peter Tank*
Protokollant: *Michael Beering*
Abwesende Gruppenmitglieder: *Henning Lawatsch, Eike Hagena*

TOP 1 – Nachbesprechung Praxistest Der Praxistest wird als ein voller Erfolg gewertet. Es konnten neue Videosequenzen für die Kollisionserkennung aufgezeichnet werden, auf denen dieses Mal ein weiteres Boot zu sehen ist. Außerdem war das Wasser bei diesem Test wesentlich ruhiger, was auf den Videos deutlich zu erkennen ist. Des Weiteren konnte festgestellt werden, dass die XBee-Verbindung stabiler lief als bisher und das es keine Stromausfälle an Bord mehr gab. Dies ist auf die überarbeitete Elektronik des Bootes zurückzuführen. Außerdem wurde der Ablauf mit vorheriger Planung der Testszenarien positiv empfunden. Das Aufzeichnen der Beschleunigungsdaten für die Regelung ist ebenfalls geglückt, sodass hieraus Reglerparameter ermittelt werden können.

TOP 2 – Status Kamera-/Laserscannerhalter Die kardanische Aufhängung zum Ausgleichen von Schwankungen an der Laserscannerplattform wird vorerst nicht berücksichtigt. Außerdem wird zur Zeit ein Bauteil für die Kamerahalterung im 3D-Drucker gedruckt.

TOP 3 – Wasserdichtigkeit Eine Plane zum Regenschutz wird vorerst nicht realisiert, da die Themen Kollisionserkennung und -vermeidung sowie Regelung höhere Priorität haben. Das Wasserfahrzeug mit der vorhandenen Folien gegen Wasser zu schützen ist vorerst ausreichend.

TOP 4 – Elektrisches System/Software Das elektrische System wird bzgl. Stromversorgung um keine weiteren Komponenten erweitert, da derzeit ein sehr stabiler und zuverlässiger Zustand vorliegt. Des Weiteren steht für einige Komponenten und Verkabelungen noch der finale Einbau an.

TOP 6 – Verschiedenes Für das Bouletermin wird aufgrund von Zeitmangel vermutlich kein Trainingstermin angeboten.

B.1.36 Gruppensitzung am 22. Juni 2016

Moderator: *Konstantin Gebel*
Protokollant: *Björn Koopmann*
Abwesende Gruppenmitglieder: *Maximilian Hipp, Peter Tank*

TOP 1 – Begrüßung Konstantin begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Aktueller Stand Henning und Eike präsentieren die gedruckten Halterungen für den Laserscanner und die Kamera. Zu der Implementierung der Algorithmen zur Kollisionserkennung gibt es nichts zu berichten.

TOP 3 – Sprint Review Dennis fasst die Ergebnisse des zehnten Sprints zusammen. Es wurden wichtige Fortschritte gemacht, die eine gute Grundlage für die Erledigung der nachfolgenden Aufgaben bilden.

TOP 4 – Sprint Retrospektive Die Stabilität des Systems konnte nachhaltig verbessert werden. Während des zweiten Praxistest konnten weiteren Szenarien aufgezeichnet werden, die eine höhere Testabdeckung ermöglichen. Das Arbeitsklima innerhalb der Gruppe ist weiterhin sehr produktiv.

TOP 5 – Planung des nächsten Sprints Um die Kommunikation der einzelnen Komponenten zu erleichtern, soll ein Wireless Access Point auf dem Boot montiert und in das bestehende System integriert werden.

TOP 6 – Verschiedenes Dennis schlägt vor, den Zeitslot für die regelmäßigen Gruppensitzungen am Mittwoch um eine Stunde von 16:15 Uhr auf 15:15 Uhr vorzuverlegen. Der Vorschlag wird einstimmig angenommen.

B.1.37 Gruppensitzung am 29. Juni 2016

Moderator: *Björn Koopmann*
Protokollant: *Dennis Pilny*
Abwesende Gruppenmitglieder: *Peter Tank, Maximilian Hipp, Konstantin Gebel, Zahra Paya*

TOP 1 – Begrüßung Björn begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Schiffsdynamik und elektrisches System Um die Benutzbarkeit des elektrischen Systems zu verbessern, wurden Hebelklemmen gekauft, die in die Otterboxen getan werden und die Elektronik versorgen. Henning und Eike präsentieren die angebrachte Halterung für die Kamera. Die Halterung für den Laser wird zur nächsten Woche fertig gestellt und angebracht.

Da sich nun auf dem Boot zwei Raspbery Pis befinden, wurde entschieden, einen zentralen Accesspoint auf dem Boot zu installieren, mit dem sich die Raspberrys verbinden. Dadurch wird zum Einem die Kommunikation zwischen den Geräten ermöglicht und zum Anderen das Debugging vereinfacht, da sich alle Geräte innerhalb eines Netzwerkes befinden.

TOP 3 – Kollisionserkennung und Sensorik Michael präsentiert seine Fortschritte bei der Implementierung des Canny Edge Algorithmus zur Erkennung von Hindernissen. Eindeutige Hindernisse werden bereits gut erkannt, es besteht jedoch Potential den Algorithmus zu verbessern. Außerdem besteht die Möglichkeit den Algorithmus durch den frequenzbasierten Ansatz von Björn zu verbessern.

TOP 4 – Nächster Praxistest Für den nächsten Praxistest wurden die Folgenden Ziele ermittelt:

- Testen der Kommunikation der Teilkomponenten untereinander
  - Onboard zu Kollisionserkennung zur Kollisionsvermeidung (Netzwerkkommunikation)
  - Verbindung zwischen Kamera und Laserdaten
- Testen des Regelungssystems
  - Anpassung der PID-Parameter des bestehenden Systems und Evaluierung
  - Inbetriebnahme des neuen Regelungssystems

Aufgrund der kommenden Prüfungsphase wird der nächste Praxistest zeitlich Anfang August stattfinden. Ein genauer Termin wird in den folgenden Sitzungen festgelegt.

TOP 5 – Verschiedenes Marius übernimmt die Betreuung der Projektgruppe, da Sascha die Universität verlassen hat. Die Projektgruppe hat bei Boule-Turnier den 7. Platz erreicht.

B.1.38 Gruppensitzung am 6. Juli 2016

Moderator: *Dennis*
Protokollant: *Konstantin Gebel*
Abwesende Gruppenmitglieder:

TOP 1 – Begrüßung Dennis begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Schiffsdynamik und elektrisches System Die Gruppe stellt die Veränderungen am elektrischen System vor. Eine Halterung für die Kamera wurde angebracht. Eike und Henning haben damit begonnen an einer Halterung für das Lidar zu arbeiten. Ein Accesspoint wurde eingerichtet.

Es wurden allerdings weitere Mängel entdeckt. Zum Beispiel stürzen Arduino und Raspberry immer noch ab und zu ab. Auch sind nach mehreren Neustarts keine Daten mehr vom IMU lesbar.

TOP 3 – Kollisionserkennung und Sensorik Björn stellt den frequenzbasierten Ansatz vor. Allerdings sind wir der Ansicht, diesen Ansatz nicht weiter zu verfolgen und stattdessen nur im Abschlussbericht zu erwähnen.

Die Halterung für die Kamera wurde angebracht. Für die Lidarhalterung fehlen allerdings noch Teile.

TOP 4 – Nächster Praxistest Als Termin für den nächsten Praxistest wurde der 4. August 2016 ab 8:30 Uhr festgelegt.

TOP 5 – Verschiedenes Es wurde angemerkt, dass die Position des Lidars eventuell im Kamerasichtfeld stören würde. Hier müssen eventuell noch Anpassungen getätigt werden.

Weiterhin kam die Frage auf, wann der Abschlussbericht abgegeben werden muss. Marius wird sich darum kümmern und uns darüber informieren.

B.1.39 Gruppensitzung am 13. Juli 2016

Moderator: *Eike Hagena*
Protokollant: *Maximilian Hipp*
Abwesende Gruppenmitglieder: *Björn Koopmann*

TOP 1 – Begrüßung Eike begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Retrospektive und Sprintplanung Positiv war, dass die Sprintziele größtenteils eingehalten werden konnten. Die Halterungen sind gelungen und der Algorithmus zur Horizonterkennung funktioniert. Negativ ist, dass es weiterhin kleinere Stabilitätsprobleme gibt. Die Zusammenarbeit im Team ist gut.

TOP 3 – Projektabschluss Die Abgabe des Projektes wird auf den 31. September datiert. Die Präsentation im Oberseminar soll am 29. September gehalten werden.

TOP 4 – Elektrisches System und Bootsaufbau Als nächste Aufgabe wird die Kommunikation zwischen den zwei Raspberry Pis avisiert. Die Halterungen am Boot müssen noch verfeinert werden.

TOP 5 – Objekterkennung Der Lidarcode soll in das bestehende Projekt integriert werden.

TOP 6 – Software Die Kommunikation zwischen den beiden Raspberry Pis soll in den nächsten Wochen realisiert werden.

B.1.40 Gruppensitzung am 20. Juli 2016

Moderator: *Henning Lawatsch*
Protokollant: *Zahra Paya*
Abwesende Gruppenmitglieder: *Peter Tank, Michael Beering, Konstantin Gebel*

TOP 1 – Begrüßung Henning begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Stabilität elektrisches System Dennis präsentiert, dass das Onboard-System immer noch ab und zu abstürzt.

TOP 3 – Kollisionsvermeidung Es soll eine Kommunikation zum Laserscanner aufgebaut werden, deren Entwurf in Abb. B.6 dargestellt ist. Auf der linke Seite sind der Raspberry, Algorithmen, Webcam und Laser angeschlossen. Die beiden Komponenten Kamera und Lidar speisen Algorithmen. Das Ergebnis ist eine AreaToAvoid-Datenstruktur. Der Client schickt an den Server über Json Daten, die anschließend integriert werden.

Björn ist der Meinung, dass aus dem Lidar wichtige Informationen rauskommen sollten, wie die Größe des Objekts und die gemessene Entfernung. Das Onboard-System versucht, den Einträgen der Liste das GPS-Zeichen zuzuordnen, dann können Informationen verknüpft werden und zu der Position des Bootes der Messzeitpunkt, der Winkel, der Abstand und die Größe bestimmt werden.

TOP 4 – Bootsaufbau Das Konzept ist fertig. Max hat über Strom für Laser vorgestellt. Eike werde nächste Woche den Stecker abholen. Kabel wird Peter besorgen. Die Halterung ist vorbereitet und muss einkleben werden.

TOP 5 – Verschiedenes Im September sollte kein Urlaub gebucht werden.

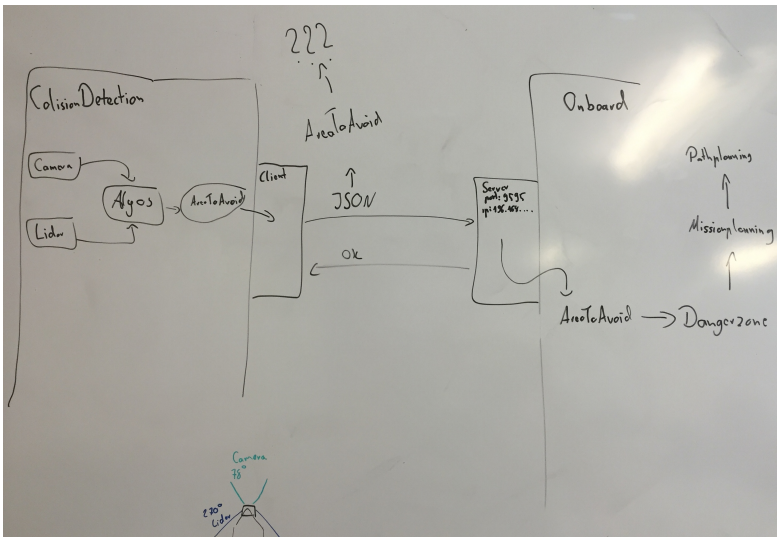


Abbildung B.6: Entwurf der Onboard-Kommunikation

B.1.41 Gruppensitzung am 27. Juli 2016

Moderator: *Zahra Paya*
Protokollant: *Peter Tank*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Zahra begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Schiffsdynamik und elektrisches System Der Laserscanner muss noch ins elektrische System integriert werden. Das elektrische System stürzt unperiodisch ab. Der Fehler wird in den nächsten Tagen behoben.

TOP 3 – Kollisionserkennung und Sensorik Hindernisse die vom Laserscanner erkannt werden, werden als Punktwolke dargestellt. Diese werden in Quadrate eingeteilt. Quadrate mit vielen Punkten haben eine hohe Wahrscheinlichkeit das ein Hindernis vorhanden ist. Diese werden bereitgestellt.

TOP 4 – Bootsaufbau Die Halterung des Laserscanners wird weiter nach hinten geschoben. Bis nächste Woche wird dieses abgeschlossen.

B.1.42 Gruppensitzung am 3. August 2016

Moderator: *Peter Tank*
Protokollant: *Maximilian Hipp*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Peter begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Praxistest Das Treffen beginnt offiziell um 8:30 Uhr. Einige Gruppenmitglieder treffen sich bereits vorher, um notwendige Vorbereitungsmaßnahmen zu erledigen.

TOP 3 – Tests am See Es sollen Parameter für die Regelung erfasst werden. Weiterhin sollen die ersten Ergebnisse der Bilderkennung getestet und ausgewertet werden. Auch der Laserscanner soll Daten aufzeichnen, die im Anschluss ausgewertet werden können.

TOP 4 – Retrospektive Generell ist der Sprint als erfolgreich bewertet worden. Als kontraproduktiv wurde die Klausurenzeit für die Gruppenmitglieder empfunden. Weiterhin ist viel Zeit für die Stabilisierung des elektrischen Systems aufgewandt worden. Lange Beschaffungsprozesse hinderten den Fortschritt am Neuaufbau des elektrischen Systems immens.

TOP 5 – Planung des nächsten Sprints Der Fokus des nächsten Sprints liegt auf der Kollisionsvermeidung. Eine Onshore-Anzeige sowie das Anzeigen von dynamischen Dangerzones wird als nächstes Ziel gesetzt. Auch müssen die Scannerdaten gefiltert werden, um die Ergebnisse des Scanners verwerten zu können. Die Regelparameter nach dem Umbau des Systems sollen aufgezeichnet und implementiert werden.

B.1.43 Gruppensitzung am 10. August 2016

Moderator: *Eike Hagena*
Protokollant: *Michael Beering*
Abwesende Gruppenmitglieder: -/-

TOP 1 – Begrüßung Eike begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Kollisionserkennung und Sensorik Im letzten Praxistest hat sich gezeigt, dass die kamerabasierte Kollisionserkennung bereits Obstacles generieren kann und diese Informationen können ebenfalls per JSON und über TCP an die Onshore-Software geschickt werden. Die Anbindung des Laserscanners funktionierte während des Tests nicht. Es hat sich gezeigt, dass sich MRPT nicht auf dem Raspberry Pi kompilieren lässt. Michael hat eine Implementierung von ROS in C++ migriert, die eine stabile Anbindung ermöglicht. Björn implementiert die noch fehlende Obstacle-Generierung für die laserscannerbasierte Kollisionserkennung.

TOP 3 – Bootsaufbau Die Halterungen sowohl für die Kamera als auch für den Laserscanner haben sich als äußerst stabil und gut einsetzbar rausgestellt.

TOP 4 – Nächster Praxistest Der nächste Praxistest wird am kommenden Donnerstag, den 18. August, durchgeführt. Das Protokoll für den Test wird von Eike geschrieben.

B.1.44 Gruppensitzung am 17. August 2016

Moderator: *Björn Koopmann*
Protokollant: *Konstantin Gebel*
Abwesende Gruppenmitglieder: *Peter Tank*

TOP 1 – Begrüßung Björn begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Bootsaufbau und elektrisches System Der Accesspoint wurde vom Mast des Bootes entfernt und stattdessen an der Halterung für den Laserscanner und die Kamera angebracht. Auch bekommt die Kamera nun über eine externe Stromquelle versorgt.

TOP 3 – Kollisionserkennung und Sensorik Das Lidarsystem wurde nun in das Gesamtsystem eingebunden. Es werden Hindernis-Objekte aus den Rohdaten erzeugt, die anschließend weiterverarbeitet werden können. Der Scanbereich des Laserscanners wird in Zellen eingeteilt.

Die erfassten Lidardaten können nun mitsamt den generierten Obstacles in einem externen Tool „ScannerViewer“ angezeigt werden. Dieser soll für die Visualisierung in späteren Praxistests eingesetzt werden. Weiterhin wurde von Dennis darauf hingewiesen, dass die Hindernisse in der Onshore-Software angezeigt und durch den Bahnplanungsalgorithmus berücksichtigt werden.

TOP 4 – Kollisionsverhütung Es wurde überlegt den bisherigen Algorithmus zur Bahnplanung zu optimieren. Allerdings wurde auch angemerkt, dass der eingesetzte A*-Algorithmus aufgrund der Größe des resultierenden Zustandsraums schlecht skalierbar ist und nicht immer den kürzesten sondern lediglich einen validen Weg erzeugt.

Es wurde festgestellt, dass die zur Verfügung stehende Hardware eher ungeeignet für eine schnelle und zuverlässige Kollisionserkennung und -verhütung ist. Daher wurde die Anschaffung von leistungsfähigerer Hardware in den Ausblick gestellt.

TOP 5 – Vorbereitung des Praxistests Der nächste Praxistest wird am kommenden Donnerstag, den 18. August, durchgeführt. Das Protokoll für den Test wird von Eike geschrieben. An diesem Praxistest sollen Lidarsystem und die Hinderniserkennung via Kamera getestet werden. Anschließend werden für die Regelung neue Parameter aufgenommen.

TOP 6 – Feinplanung des Projektendspurts Es wurden vorläufig drei weitere Termine für die nächsten Praxistests festgelegt. Diese sind der 1., 9. und 15. September, wobei der letzte Termin für das Aufnehmen von Videos und Bilder für die Abschlusspräsentation gedacht ist. Außerdem soll schon jetzt mit dem Schreiben eines Ausblicks angefangen werden.

B.1.45 Gruppensitzung am 24. August 2016

Moderator: *Zahra Paya*
Protokollant: *Eike Hagena*
Abwesende Gruppenmitglieder: *Henning Lawatsch, Maximilian Hipp, Michael Beering*

TOP 1 – Begrüßung Zahra begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Praxistest Während des letzten Praxistests wurden erfolgreiche Ergebnisse beim Lidar, der Kamera und bei der Regelung erzielt. Die Lidardaten sind geloggt und mit der Kamera wurden Gegenstände erkannt. Es müssen aber eventuell noch Filter für entstehende Reflexionen hinzugefügt werden. Bei der Regelung hat sich gezeigt, dass diese gegenüber die er Vorgängergruppe wesentlich besser läuft. Für die Dokumentation soll dies mittels einer Gegenüberstellung gezeigt werden.

TOP 3 – Elektrisches System/Software Die TCP-Kommunikation läuft noch nicht ganz stabil. Die Probleme mit dem Lidarsystem sind behoben. Das Problem war die unterschiedliche Implementierung, beispielsweise wurden unterschiedliche Entfernungsparameter angegeben.

TOP 4 – Bootsaufbau Es gibt noch Kleinigkeiten zur Befestigung die noch erledigt werden müssen. Auch die Steckverbindungen sind noch nicht Dokumentiert, dies wird im nächsten Sprint erledigt.

TOP 5 – Sprint Review Alle anwesenden Gruppenmitglieder waren der Meinung, dass der Sprint sehr positiv war. Besonders die Regelungsgruppe war sehr zufrieden und erfolgreich.

TOP 6 – Sprint Retrospektive Es müssen noch Polfilter getestet werden.

TOP 7 – Planung des nächsten Sprints Der nächste Sprint wurde von der Gruppe geplant, die einzelnen Tasks wurden im Jira festgehalten.

TOP 8 – Nächster Praxistest Der nächste Praxistest findet am 1. September statt. Hier sollen Fotos und Videos für die Dokumentation und die Abschlusspräsentation gemacht werden. Marius ist an diesem Termin leider verhindert.

B.1.46 Gruppensitzung am 31. August 2016

Moderator: *Peter Tank*
Protokollant: *Henning Lawatsch*
Abwesende Gruppenmitglieder: *Maximilian Hipp*

TOP 1 – Aktueller Praxistest Treffen für den Praxistest ist am 1. September um 8:00 Uhr. Konstantin bringt ein Verlängerungskabel mit. Michael stellt seinen Anhänger für den Transport zur Verfügung. Für die Abschlusspräsentationen sollen schon einmal Videos gedreht werden.

TOP 2 – Nächster Praxistest Der nächste Praxistest findet am 15. September statt. Dies soll zugleich der letzte geplante Test sein.

TOP 3 – Abschlussdokumentation Jede Gruppe muss fehlende Überschriften zu ihren Themen hinzufügen, um den noch vorhandenen Arbeitsaufwand einzuschätzen. Die Texte müssen nun fertiggestellt werden. Die Deadline ist ungefähr Mitte des Monats September, darauf soll die gesamte Dokumentation in einer Gruppe korrigiert werden.

B.1.47 Gruppensitzung am 7. September 2016

Moderator: *Henning*
Protokollant: *Zahra*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Henning begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Letzter Praxistest Bei dem vergangenen Praxistest wurden verschiedene Videos aufgezeichnet, in denen das Ruderboot *Petzi* und ein Kajak als Hindernis dienten. Es wurde gezeigt, dass die Kollisionserkennung mit dem Laserscanner gut funktioniert. Die Kollisionsverhütung hat nicht so gut geklappt, wegen des Problems mit dem Raspberry Pi, auf dem die Missionsplanung läuft.

TOP 3 – Nächster Praxistest Der nächste Praxistest findet am Donnerstag, den 15. September, statt. Treffen ist um 8:30 Uhr. Es werden die Kollisionserkennung getestet, Videos aufgenommen und Bilder für die Abschlusspräsentation gemacht.

TOP 4 – Abschlussdokument Eike und Henning haben angefangen, die Erweiterungen des Systems z.B für die Wasserfestigkeit oder die Motoren zu dokumentieren. Bei den Szenarien in dem Regelungsteil wäre es besser, manche Bilder als Tabelle zu setzen. Die Regelung soll von Realisierungskapitel unter Datenkomponenten verschoben werden.

Im Realisierungskapitel wurde der Abschnitt Bootsaufbau und Seetauglichkeit als Software umbenannt. Es wurde auch ein Unterkapitel Architektur bei Hardware hinzugefügt, um ganz grob das Zusammenspiel des Accesspoints, der Raspberry Pis, usw. zu erklären.

Bei dem Systemumbau wäre es besser, wenn man sagt, welche Bauteile im Vergleich zu Mops III neu dazu gekommen sind. Die Testprotokolle sollen in den Anhang verschoben und dazu ein Text hinzugefügt werden.

TOP 5 – Abschlusspräsentation Für die Präsentation wäre es eine Idee, zu beschreiben, wie die Onshore-Software, das Onboard-System und der Bootsaufbau sich im Vergleich zu Mops III verändert haben. Die aufgenommenen Videos sollen geschnitten werden.

Max präsentiert seine Vorlage für die Abschlusspräsentation. Es wäre besser, echte Fotos anstelle von schematischen Darstellungen zu benutzen. Die Meilensteine sollen als grafische Abbildung gezeigt werden. Zu viel Information für Zuschauer muss vermieden werden.

TOP 6 – Verschiedenes Marius kümmert sich um das Eintragen einer Prüfung für die Projektgruppe. Der PG-Raum soll nach der PG-Zeit durch die Gruppenmitglieder aufgeräumt werden.

B.1.48 Gruppensitzung am 14. September 2016

Moderator: *Michael*
Protokollant: *Konstantin*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Michael begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Nächster Praxistest Das Ziel des nächsten Praxistests wurde besprochen. Im Vordergrund soll dabei das Sammeln von Bildmaterial und Videos für die kommende Präsentation und die Abschlussdokumentation stehen. Wie üblich wird sich die Gruppe um 8:30 Uhr im PG-Raum treffen.

TOP 3 – Abschlussbericht Das Kapitel Konzept ist weitesgehend fertiggestellt. Auch die Dokumentation der Kollisionsverhütung wurde fertiggestellt. Es wurden weitere Ergänzungen an den Ausarbeitungen zur Onshore- und Onboard-Software vorgenommen.

Aufgrund eines Hinweises von Marius soll es kein eigenes Kapitel für die Entwicklertests geben. Die Tests werden stattdessen im jeweiligen Abschnitt der Realisierung eingebunden.

Weiterhin soll ein Performance-Test durchgeführt werden, um zu belegen, dass die Hardware des Raspberrys unzureichend für eine erfolgreiche Kollisionsverhütung ist. Hierbei werden Routenplanung, Bildverarbeitung und Laserscanner parallel auf dem Raspberry und dem Macbook getestet.

Verschiedenes Dennis stellt die simulative Kollisionsverhütung vor. Diese belegt, dass das Boot einem detektierten Hindernis unter Idealbedingungen mit einem ausreichenden Sicherheitsabstand ausweichen kann.

B.1.49 Gruppensitzung am 21. September 2016

Moderator: *Konstantin Gebel*
Protokollant: *Eike Hagena*
Abwesende Gruppenmitglieder: –

TOP 1 – Begrüßung Konstantin begrüßt die anwesenden Gruppenmitglieder.

TOP 2 – Dokumentation Michael sammelt die verbleibenden Aufgaben an der Tafel und erklärt, was bei ihrer Bearbeitung zu beachten ist.

TOP 3 – Präsentation Konstantin zeigt den aktuellen Stand der Präsentation. Die noch fehlenden Bereiche werden von ihm angesprochen. Außerdem zeigt Michael noch Videos, die in die Präsentation integriert werden sollen.

TOP 3 – Sprint Review/Retrospektive Dennis merkt an, dass die Kollisionsverhütung während des letzten Praxistests erfolgreich getestet werden konnte. Auch die Dokumentation ist schon sehr weit fortgeschritten, es müssen aber noch einige Aufgaben erledigt werden. Es waren sich alle einig, dass der Sprint sehr gut lief.

TOP 4 – Verschiedenes Marius weißt drauf hin, dass wir in der Präsentation deutlich machen sollten, dass die Kollisionsverhütung nur in der Simulation vernünftig läuft, da das Onboard-System über eine zu geringe Rechenleistung verfügt. Nächste Woche ist noch einmal ein Treffen für die Präsentation. Außerdem wollen wir die Präsentation üben, es sollen am Donnerstag dann maximal drei bis vier Leute vortragen.

B.2 Interviewprotokolle

B.2.1 Interview am 3. Dezember 2015

Befragter: *Prof. Dr. Axel Hahn*
Interviewer: *Dennis Pilny*
Protokollant: *Björn Koopmann*

Einführung Prof. Dr. Axel Hahn begrüßt die anwesenden Gruppenmitglieder und lobt die Projektgruppe ausdrücklich für ihre bisherige Arbeit. Im Anschluss an eine kurze Vorstellungsrunde fasst er die Aufgabenstellung des Projektvorhabens erneut in eigenen Worten zusammen und gibt einen Überblick über den geplanten Verlauf der Veranstaltung. Aufgrund anderer Verpflichtungen entschuldigt er sich für seine Abwesenheit bei den Gruppensitzungen, versichert aber, sich regelmäßig bei Sascha über den Projektfortschritt zu informieren.

Erfahrungen aus den vorherigen Projektgruppen Prof. Dr. Axel Hahn berichtet von der Arbeit der vorherigen Projektgruppen. Mops i hatte die Aufgabe, ein prototypisches System zum Ausprobieren der ersten Ideen zu entwickeln.

Die darauffolgende Gruppe Mops ii konnte die aktuell genutzte Infrastruktur durch ein neues Architekturkonzept nachhaltig prägen und signifikante Verbesserungen in den Bereichen Ausfallsicherheit und Zuverlässigkeit vornehmen.

Die bisher letzte Projektgruppe Mops iii beschäftigte sich vor allem mit der Robustheit der Hard- und Softwarekomponenten sowie mit der Homogenisierung der eingesetzten Bauteile.

Besprechung der Projektziele Im Anschluss an die einleitenden Worte von Prof. Dr. Axel Hahn werden der aktuelle Stand sowie die im Vorfeld erarbeitete Projektvision präsentiert. Eine konstruktive Diskussion über die vorgesehenen Inhalte anregend, moderiert Dennis die schrittweise Besprechung der einzelnen Ziele. Zu den genannten Punkten äußert sich Prof. Dr. Axel Hahn wie folgt:

Ausfallsicherheit/Zuverlässigkeit Das Thema „Ausfallsicherheit und Zuverlässigkeit“ soll im Rahmen des Projektvorhabens Mops iv nur eingeschränkt betrachtet werden, da sich in diesem Bereich mit vergleichsweise geringem Aufwand schnelle Erfolge erzielen lassen und die Herausforderungen durch die vorherigen Projektgruppen bereits gut gelöst wurden. Die entsprechenden Aspekte sollen unter dem Stichwort „Sicherheit“ zusammengefasst und mit einer geringen Priorität in die weiteren Betrachtungen einbezogen werden.

Seetauglichkeit Die Verbesserung der Seetauglichkeit sieht Prof. Dr. Axel Hahn als wichtigen Punkt an. Mit einem heißen Draht könnten große Styroporblöcke zugeschnitten werden, um mögliche Freiräume für eine Erhöhung des Auftriebs zu nutzen. Zudem könnte der Bootskörper mit einer Abdeckung aus LKW-Plane versehen werden, um das Innenleben vor Spritzwasser und Niederschlägen zu schützen.

Reichweite/Antrieb/Energieversorgung Die aktuelle Geschwindigkeit des Bootes beträgt ca. 3 kn. Da in den Mündungsgebieten größerer Flüsse mit deutlich höheren Strömungsgeschwindigkeiten zu rechnen ist, ist die Optimierung des Antriebskonzepts prinzipiell interessant. Da der Wissenszuwachs bei der Anschaffung neuer Motoren als vergleichsweise gering einzuschätzen ist, ist die Umsetzung im Rahmen der Projektgruppe MOPS IV allerdings eher weniger sinnvoll. Auch die Möglichkeit, weitere Batterien oder Solarmodule in das System zu integrieren, erweisen sich im Verlauf des Gesprächs als kostspielige und einfache Aufgaben, die sich allein durch einen hohen finanziellen Aufwand realisieren lassen.

Regelung/Control and Modeling Die Überarbeitung der Regelung der Motoren, eine Bewertung der Seetauglichkeit und der Ausfallsicherheit sowie die Beschreibung der Eigenschaften des Systems in einem mathematischen Modell schreibt Prof. Dr. Axel Hahn eine hohe Priorität zu. Eine umfassende Evaluierung des dynamischen Modells sowie des Stabilitätsverhaltens empfindet er als äußerst spannend.

Sensormessung Das Nutzlastmanagement bezeichnet Prof. Dr. Axel Hahn als vergleichsweise leicht zu lösende Aufgabe. Die Projektgruppe sollte sich unter anderem darüber Gedanken machen, ob es nicht möglich ist, einen Moonpool in das System zu integrieren. Von dem bisher eingesetzten Sensorkorb ist er wenig überzeugt und rät von der weiteren Verwendung ausdrücklich ab.

Rechtliche Rahmenbedingungen Die rechtlichen Rahmenbedingungen sollen – mit Ausnahme der entsprechenden Seminarausarbeitung – im Rahmen dieser Projektgruppe nicht weiter betrachtet werden. Während der gesamten Bearbeitungszeit kann davon ausgegangen werden, dass ein „Boot an einer virtuellen Leine“ entwickelt wird.

Kollisionserkennung und -verhütung Den Einsatz eines der vorhandenen RADAR-Geräte bezeichnet Prof. Dr. Axel Hahn als große Herausforderung, da die Systeme sehr hoch angebracht werden müssten und dies negative Auswirkungen auf den Schwerpunkt der Plattform hätte. Auch die möglichen Gefahren durch die hohe Strahlung und die Komplexität der automatischen

Zieldetektion sieht er als wichtige Gründe gegen den Einsatz eines RADAR-Systems an.

Die Möglichkeit, optische Verfahren zur Detektion und Identifikation von Hindernissen einzusetzen, empfindet er als deutlich ansprechender. Im Nahbereich könnten dadurch sogar Entfernungen abgeschätzt werden. Die Umsetzung würde sich durch den Einsatz von mehreren Kameras bei einer ausreichenden Lösungsqualität deutlich vereinfachen. Unter Umständen kann ein „kleines“ RADAR-System als Ergänzung für die optische Erkennung dienen.

Anmerkungen zur agilen Vorgehensweise Prof. Dr. Axel Hahn berichtet von seinen Erfahrungen mit dem Einsatz von Scrum als Vorgehensmodell. Die agile Vorgehensweise hält er persönlich zwar für die Entwicklung von „SAP-Oberflächen super geeignet“, sieht die Umsetzung eines systematischen Exploration des Entwurfsraums im agilen Kontext allerdings als große Herausforderung an.

Er begrüßt die Entscheidung zur Verwendung des Vorgehensmodells, rät aber gleichzeitig dazu, diese bis zum Ende der Projektgruppe keinesfalls zu widerrufen und das „Experiment durchzuziehen“. Die Einrichtung von Freiräumen für kreative Ansätze oder umfassendere Aufgaben erachtet er als sinnvolle Ergänzung.

Fazit und Zusammenfassung Insgesamt ist für ihn die Kernfrage „Was interessiert uns als Projektgruppe“ von zentraler Bedeutung. Die Visionen und Vorstellungen der Betreuer spielen seiner Aussage nach nur eine eingeschränkte Rolle. Den weiteren Verlauf der Projektgruppe Mops iv macht er von den persönlichen Interessen der einbezogenen Gruppenmitglieder abhängig.

Für den weiteren Verlauf hält Prof. Dr. Axel Hahn eine Aufteilung der Gruppenmitglieder in zwei Arbeitsgruppen für sinnvoll. Neben einer „Hardware-Truppe“ für die Einbindung der Sensorik, das Erweitern des Nutzlastmanagements, die Gestaltung des Bootsaufbaus sowie der Verbesserung der Wasserfestigkeit und der Seetauglichkeit sieht er eine „Hindernis-Truppe“ für die Erkennung von Hindernissen, einer Optimierung der Bahnplanung und Regelung sowie der Definition von Ausweichregeln als zielführende Aufteilung.

Finanzieller Rahmen Der finanzielle Rahmen kann flexibel gestaltet werden. Im Vordergrund stehen hierbei die Ideen der Gruppenmitglieder sowie ihre Bedeutung für das Projektvorhaben. Über Ausgaben in beliebiger Höhe kann gerne diskutiert werden, solange die Ideen den Projektzielen dienen und einen Wissenszuwachs mit sich bringen.

Sicherheitshinweise Im weiteren Verlauf werden die anwesenden Gruppenmitglieder über die einzuhaltenden Sicherheitsmaßnahmen informiert. Wäh-

rend der praktischen Tests müssen alle Beteiligten eine Rettungsweste tragen. Die Plattform darf nur dann ferngesteuert werden, wenn niemand am Boot arbeitet. Die Stromversorgung ist umgehend zu trennen, sobald sich ein Gruppenmitglied in der unmittelbaren Nähe des Bootes aufhält.

Abschluss des Interviews Prof. Dr. Axel Hahn schließt seine Ausführungen ab und erklärt sich bereit, für die Beantwortung von Fragen jederzeit gerne persönlich zur Verfügung zu stehen. Er verabschiedet sich von den anwesenden Gruppenmitgliedern und wünscht allen viel Erfolg für die kommenden zwei Semester.

B.2.2 Interview am 8. Dezember 2015

Befragter: *Prof. Dr. Oliver Zielinski*
Interviewer: *Peter Tank*
Protokollant: *Michael Beering*

Einführung Zum Einstieg in das Interview wurde ein kurzer Statusbericht über die laufende Projektgruppe gegeben. Es wurde vorgestellt, dass eine Projektvision erstellt wurde, in der die Nachteile des aktuell eingesetzten Systems aufgeführt und Verbesserungsvorschläge gemacht wurden. Außerdem wurde der zeitliche Rahmen (10.10.2015 - 30.09.2016) für das Projekt erwähnt.

Des Weiteren erläuterte Herr Zielinski kurz seinen Forschungsschwerpunkt und die Aufgabenfelder seiner Abteilung (vgl. weiterführende Informationen).

Ziel der Projektgruppe Herr Zielinksi sieht als primäres Ziel den Ausbau der vorhandenen Plattform. Die Konstruktion einer robusten und stabilen Plattform spielt in den rauen Einsatzgebieten eines solchen Wasserfahrzeugs eine besondere Rolle. Auf einer gut umgesetzten Plattform lassen sich im Nachhinein leicht verschiedene Sensoren anbringen.

Zum Ausbau der vorhandenen Plattform zählt Herr Zielinski darüber hinaus die Punkte situational awareness und collision avoidance, denn hierdurch kann die Autonomie des Bootes deutlich erhöht werden. Um dies umzusetzen weist Herr Zielinski auf eine gute Modellbildung und Regelung hin, die dafür essentiell ist. Das Einbeziehen der Umweltbedingungen (Wind, Strömungen,...) in die Regelung ist außerdem ein interessanter Ansatz.

Nachdem sich die Vorgängergruppen viel mit der Basistechnologie des Wasserfahrzeugs auseinandergesetzt haben, sieht Herr Zielinksi das Verbesserungspotential besonders im Bereich der Regelung und der Komforterhöhung.

Langfristige Ziele Die langfristigen Ziele bestehen für Herrn Zielinski darin, Flusssysteme (Weser, Elbe,...) oder den Bereich des Jadebusen zu erkunden. Er weist allerdings daraufhin, dass die nicht vorhandene Schiffserkennung

und rechtliche Probleme dies verhindern werden. Außerdem ist es nicht ratsam die rechtlichen Aspekte eines solchen Projektes zu verfolgen, da dies bereits ausführlich von den Vorgängergruppen getan wurde. Weiterhin spielen starke Strömungen in Flüssen und Küstengebieten eine Rolle, wofür der Motor des bestehenden Systems nicht ausgelegt ist. Dies könnte durch finanzielle Mittel leicht gelöst werden, muss allerdings nicht in naher Zukunft passieren.

Ein weiteres langfristiges Ziel sieht Herr Zielinski im Ausnutzen der Tide bei der Fortbewegung. Beispielsweise könnte das Boot mit der Tide zu einem bestimmten Messbereich fahren, dort Messungen durchführen und mit rückläufiger Tide zu seinem Ursprungsort zurückkehren. Darüber hinaus ist das Ausnutzen von Strömungen bei der Fortbewegung ein interessantes Thema.

Missionsumfang Die mögliche Dauer einer Mission schätzt Herr Zielinski auf etwa zwölf Stunden. Ein Szenario könnte hierbei so aussehen: Ein Mitarbeiter setzt die Plattform morgens aus, diese führt über den Tag verteilt an verschiedenen Orten Messungen durch. Abends kehrt das Wasserfahrzeug zu seinem Ursprungsort zurück, um dort von einem Mitarbeiter eingesammelt zu werden.

Bei dem Einsatzgebiet soll es sich immer um regionale Gebiete (z.B. Tweelbäker See) handeln, die dicht an der Zivilisation liegen.

Einschränkungen bzgl. des Wetters sollen nicht existieren. Denn häufig sind Messungen in Stürmen oder bei schlechtem Wetter von besonderem Interesse. Dennoch wäre es wünschenswert, wenn eine Intelligenz verfügbar wäre, die für das Boot ungünstige Wetterbedingungen erkennt und die Mission gegebenenfalls abbricht. Auf diese Weise soll versucht werden die sogenannte Schönwetterforschung, die im maritimen Umfeld häufig betrieben wird, zu umgehen.

Messungen/Sensorik Was das Durchführen von Messungen betrifft, ist für Herrn Zielinski neben den Messwerten auch die Farbe der Wasseroberfläche und des Himmels von Interesse. Hierfür könnten Kameras entsprechend positioniert werden. Die Ausrichtung der Kamera auf die Wasseroberfläche ist in einem Winkel von ca. 30° ideal. Aus diesen Bildern können im Anschluss weitere Umweltparameter bestimmt werden.

Die Ansteuerung bestimmter Missionspunkte sollte eine Toleranz von $5 - 10$m aufweisen. Diese Werte ergeben sich daraus, dass ein erfahrener Kapitän nur ähnlich gute Toleranzen einhalten kann. Nach Möglichkeit sollte sich die Genauigkeit im System einstellen lassen und je nach Wetterlage oder Strömungen variieren. Der Einsatz von GPS liefert hierfür ausreichend gute Ergebnisse und muss nicht weiter verbessert werden.

Beim Durchführen der Messungen mittels Sensorkorb sind Wassertiefen von maximal 20m wünschenswert. Hier sind Ergebnisse um 10m allerdings schon ausreichend. Die Bachelorarbeit, die den Sensorkorb entwickelt hat,

wurde bis auf die Designphase von Herrn Zielinskis Abteilung nicht weiter verfolgt.

Bodenproben Die Entnahme von Bodenproben ist für verschiedene Arbeitsgruppen in Herrn Zielinskis Umfeld ein interessantes Thema. Hierfür gibt es verschiedene Greiftechniken, die besonders geeignet sind. Herr Zielinski legt allerdings größeres Interesse auf die Entnahme von Wasserproben in verschiedenen Tiefen (z.B. 1,2 und 3m). Hierfür ist eine Genauigkeit des Probenentnahmesystems von etwa 20cm erforderlich. Ein einfaches System könnte dabei wie folgt aussehen: An einem geeigneten Vorrichtung werden sechs Flaschen angebracht und über eine Pumpe mit Wasser gefüllt. Der Ansaugschlauch muss hierfür unter dem Rumpf befestigt sein und verschiedene Tiefen erreichen können.

Weiterführende Informationen Im Rahmen des Interviews wies Herr Zielinski auf einige weiterführende Informationen hin. Diese werden im Folgenden kurz erläutert.

Abteilung Prof. Zielinski Die Abteilung Marine Sensorsysteme unter der Leitung von Prof. Zielinski beschäftigt sich mit der Entwicklung innovativer Messverfahren und Sensoren, die im Meer einsetzbar sind. Fokus wird hierbei auf Verfahren gelegt, die auf Optik oder Akkustik basieren. http://www.icbm.de/marine-sensorsysteme/.

www.citclops.eu Mit der App Citclops ist es möglich auf Basis von Bilddaten, die von einem Gewässer aufgenommen wurden, Umweltparameter zu bestimmen. Diese werden in eine Karte eingetragen und der Community zur Verfügung gestellt. Die Idee ist, jedem die Möglichkeit zu schaffen sich an der Beobachtung von Umweltparametern zu beteiligen und so in kurzer Zeit einen großen Datenpool aufzubauen.

www.eyeonwater.org EyeOnWater ist eine Ergänzung zu Citclops und zeigt in einer Karte die aufgenommen Bilder und weitere Informationen an.

www.nexosproject.eu Das Projekt NeXOS verfolgt das Ziel multifunktionale Sensorsysteme zu entwickeln, die sowohl auf mobilen als auch auf stationären Plattformen montiert werden können. Der Fokus bezieht sich auf Plattformen aus der Meeres- und Umweltforschung.

http://news.science360.gov/obj/video/959f41af-9b56-45e9-afb3-a615f4 43ebae/jetyak?auid=15233125 Video eines umgebauten Kajaks, welches zur Messung von Wasserparametern eingesetzt wird. Der primäre Einsatzbereich sind Regionen, die Menschen schlecht erreichen können.

On Active Current Selection for Lagrangian Profilers In diesem Artikel, der unter anderem von Prof. Zielinski verfasst wurde, geht es um das Ausnutzen tidaler Strömungen bei der Fortbewegung von Wasserfahrzeugen.

B.3 Testprotokolle

B.3.1 Praxistest am 7. April 2016

Während des ersten Praxistests am 7. April 2016 sollten die Funktionsfähigkeit des aktuell eingesetzten Systems unter realen Einsatzbedingungen validiert und Testdaten für die Entwicklung einer effizienten und effektiven Kollisionserkennung gesammelt werden.

Datum, Uhrzeit	7. April, 9:00 bis 12:30 Uhr
Gewässer	Tweelbäker See
Standort	Am Tweelbäker See 1 26135 Oldenburg
Wetter	Leicht bewölkt, schwacher Wind, vereinzelt starker Sonnenschein
Tester	Michael Beering Konstantin Gebel Eike Hagena Maximilian Hipp Björn Koopmann Henning Lawatsch Zahra Paya Dennis Pilny Peter Tank
Testfälle	Durchfahren vordefinierten Testszenarien, Aufzeichnung von Videosequenzen für die *Kollisionserkennung*, Aufzeichnung von Positionsdaten und Motorwerten für die *Regelung*

Aufgrund eines bereits im Vorfeld aufgetretenen Defekts des aktiven USB-Hubs und damit entstandenen Instabilitäten der Stromversorgung der verwendeten Raspberry Pis, konnte das Onboard-System im Rahmen dieses Praxistests nicht vollumfänglich validiert werden. Mithilfe der integrierten RC-Fernsteuerung konnten die nachfolgend beschriebenen Testfälle dennoch

durchgeführt und die erhaltenen Testdaten im Nachgang zur Optimierung des Systems verwendet werden.

Kollisionserkennung

Bereits im Vorfeld des Tests wurde das bestehende System durch die Befestigung zweier GoPro-Kameras am Bug modifiziert. Während das Boot mithilfe der RC-Fernbedienung durch verschiedene, vordefinierte Szenarien manövriert wurde, konnten die Videosequenzen auf den internen Speichern hinterlegt und nach der Rückkehr ins OFFIS ausgewertet werden.

Im Verlauf des Praxistests konnten die nachfolgend aufgeführten Szenarien erfolgreich abgefahren und durch die Aufzeichnung geeigneter Videosequenzen in die Evaluierung einbezogen werden:

Horizont Wasser mit Horizont

Wellen Starker/schwacher Wellengang, Wellen quer zum Boot

Ufer Fahrt entlang eines Ufers, Fahrt in Richtung eines Ufers

Statische Hindernisse Steg

Dynamisches Hindernisse Blaue Kiste als „Dummy"-Hindernis

Die weiterhin genannten Szenarien konnten während des ersten Praxistests nicht vollständig abgearbeitet werden. Eine Aufzeichnung der entsprechenden Videosequenzen ist für den zweiten Praxistest geplant.

Statische Hindernisse Poller, Bojen/Tonnen, festgemachte Boote, Bäume, weitere kaum/schlecht sichtbare Hindernisse, Schwimmer

Dynamische Hindernisse Segel- und Motorboote, Schwimmer

Regelung

Um die verwendeten Regler möglichst exakt an die tatsächlichen Systemparameter anpassen zu können, werden genaue Informationen über die Eigendynamik des Bootes und sein Verhalten auf der Wasseroberfläche benötigt. Das systemeigene Sensorboard einbeziehend, konnten die gewünschten Werte während der Durchführung der beschriebenen Testfälle aufgezeichnet werden.

Testfall 1: Zick-Zack-Manöver

Nr. P1-S1	Durchführung von Kurswechseln zur Aufzeichnung des tatsächlichen Systemverhaltens
Testobjekt	Onboard-System (I1)

Eingabedaten	Mission: `testmission1.mission` Fernsteuerung: inaktiv Geschwindigkeit: $0 \leq v \leq 4$ Hindernisse: keine
Vorbedingung	Das System ist inaktiv.
Erwartete Reaktion	Folgen des vorgesehenen Zick-Zack-Kurses unter Einhaltung der festgelegten Geschwindigkeit
Nachbedingung	Das System ist inaktiv.
Bemerkungen	–

Nr. P1-S1-A	Auswertung des Testfalls *P1-S1* (Systemtest)
Datum, Uhrzeit	7. April 2016, 11:00 bis 11:15 Uhr
Vorbedingung	erfüllt
Tatsächliche Reaktion	Zeitverzögertes Folgen des vorgesehenen Zick-Zack-Kurses unter Einhaltung der festgelegten Geschwindigkeit, strömungsabhängige Abweichungen der Positionsdaten
Nachbedingung	erfüllt
Artefakte	Gespeicherte Positionsdaten und Messwerte der Navigationssensorik zur Identifikation der Reglerparameter
Bemerkungen	–

Testfall 2: Beschleunigungsfahrt

Nr. P1-S2	Beschleunigung des Bootes aus dem Stillstand auf die maximale Geschwindigkeit zur Aufzeichnung des tatsächlichen Systemverhaltens
Testobjekt	Onboard-System (I1)
Eingabedaten	Mission: `testmission1.mission` Fernsteuerung: inaktiv Geschwindigkeit: $0 \leq v \leq 4$ Hindernisse: keine

Vorbedingung	Das System ist inaktiv.
Erwartete Reaktion	Beschleunigen des Systems unter Einhaltung der festgelegten Geschwindigkeit
Nachbedingung	Das System ist inaktiv.
Bemerkungen	–

Nr. P1-S1-A	Auswertung des Testfalls *P1-S1* (Systemtest)
Datum, Uhrzeit	7. April 2016, 11:15 bis 11:30 Uhr
Vorbedingung	erfüllt
Tatsächliche Reaktion	Zeitverzögertes Beschleunigen des Systems unter Einhaltung der festgelegten Geschwindigkeit, strömungsabhängige Abweichungen der Positionsdaten
Nachbedingung	erfüllt
Artefakte	Gespeicherte Positionsdaten und Messwerte der Navigationssensorik zur Identifikation der Reglerparameter
Bemerkungen	–

B.3.2 Praxistest am 13. Juni 2016

Während des zweiten Praxistests am 13. Juni 2016 sollten die bereits vorgenommenen Änderungen am elektrischen System unter realen Einsatzbedingungen validiert und weitere Videosequenzen für die Entwicklung der Kollisionserkennung aufgezeichnet werden. Im weiteren Verlauf wurden zudem zusätzliche Datensätze für die Parameteridentifikation aufgezeichnet.

Datum, Uhrzeit	13. Juni, 8:00 bis 14:00 Uhr
Gewässer	Tweelbäker See
Standort	Am Tweelbäker See 1 26135 Oldenburg
Wetter	Leicht bewölkt, schwacher Wind, vereinzelt starker Sonnenschein

Tester	Michael Beering Konstantin Gebel Eike Hagena Maximilian Hipp Björn Koopmann Henning Lawatsch Zahra Paya Dennis Pilny Peter Tank
Testfälle	Durchfahren vordefinierten Testszenarien, Aufzeichnung von Videosequenzen für die *Kollisionserkennung*, Aufzeichnung von Positionsdaten und Motorwerten für die *Regelung*

Kollisionserkennung

Mithilfe der im Zuge des vorherigen Praxistests montierten GoPro-Kameras konnten weitere Videosequenzen aufgenommen werden, aus denen nach der Rückkehr ins OFFIS zusätzliche Szenarien extrahiert wurden.

Regelung

Wie auch schon während des vorherigen Praxistests am 7. April wurden bei dieser Testfahrt Datensätze aufgezeichnet, mithilfe derer die verwendeten Regler an das tatsächliche Systemverhalten angepasst werden konnten. Das systemeigene Sensorboard und die Sensoren eines zusätzlich montierten Tablets einbeziehend, konnten die benötigten Werte während der Durchführung der in Abschnitt B.3.1 beschriebenen Testfälle aufgezeichnet werden.

B.3.3 Praxistest am 4. August 2016

Während des dritten Praxistests am 8. August 2016 wurde die Kollisionserkennung getestet. Hierbei wurden das erste mal die Webcam und der Laserscanner auf dem Boot montiert. Weiterhin wurde an diesem Tag die bestehende Regelung evaluiert und neue Parameter aufgenommen.

Datum, Uhrzeit	7. April, 9:00 bis 12:30 Uhr
Gewässer	Tweelbäker See
Standort	Am Tweelbäker See 1 26135 Oldenburg

Wetter	Leicht bewölkt, starker Wind, vereinzelt starker Sonnenschein
Tester	Michael Beering Konstantin Gebel Eike Hagena Maximilian Hipp Björn Koopmann Henning Lawatsch Zahra Paya Dennis Pilny Peter Tank
Testfälle	Durchfahren einer Mission zur Evaluierung der Reglerparameter, Testen der bisherigen Kollisionserkennung

Aufgrund des starken Windes und Wellengangs schien es zunächst schwer das Boot unter Kontrolle zu halten. Dies zeigt, dass die eingebauten Motoren für einen zuverlässigen Betrieb nicht ausreichend sind. Allerdings war es dennoch möglich in den nachfolgend aufgezählten Bereichen Tests durchzuführen.

Kollisionserkennung

Anders als in den vorher durchgeführten Praxistests, wurden kein Videomaterial gesammelt. Stattdessen sollte die bestehende Kollisionserkennung getestet werden. Es wurde auch kein bestimmtes Szenario abgefahren, sondern lediglich eine Mission über die Onshore-Software abgefahren. Leider konnte die Kollisionserkennung per Laserscanner nicht überprüft werden, aber über die Bildverarbeitung wurden schon die ersten Hindernisse erkannt.

Regelung

Um die Reglerparameter zu testen, wurde eine Mission über die Onshore-Software abgefahren. Es stellte sich heraus, dass der Kurs des Bootes mit den aktuellen Parametern im Vergleich zu vorherigen Testfahrten deutlich besser eingehalten wurde. Dennoch könnten noch weitere Feinabstimmungen in den Reglerparametern durchgeführt werden.

B.3.4 Praxistest am 18. August 2016

Für den vierten Praxistest am 18. August 2016 wurde ein weiteres Mal die Kollisionserkennung getestet. Dazu wurden sowohl Aufnahmen mit der Ka-

mera, als auch mit dem Laserscanner gemacht. Auch für die Regelung sollten weitere Tests gemacht werden.

Datum, Uhrzeit	18. April, 9:00 bis 12:30 Uhr
Gewässer	Tweelbäker See
Standort	Am Tweelbäker See 1 26135 Oldenburg
Wetter	Starker Sonnenschein, vereinzelt ein paar Wolken, Windstille
Tester	Michael Beering Konstantin Gebel Eike Hagena Maximilian Hipp Björn Koopmann Henning Lawatsch Zahra Paya Dennis Pilny Peter Tank
Testfälle	Testen der Kamera und des Laserscanners für die *Kollisionserkennung*, Durchfahren einer Mission zur Evaluierung der Regelparameter

Kollisionserkennung

Für diesen Praxistest wurden – wie auch schon bei der vorherigen Testfahrt – sowohl die Kamera als auch der Laserscanner im vorderen Bereich des Bootes angebracht. Es sollten verschiedene Hindernisse mit den Systemen aufgezeichnet und erkannt werden. Um ein dynamisches Hindernis zu simulieren, sind wir mit einem weiteren Boot auf dem See gefahren.

Während des Praxistests wurde das folgende Szenario wiederholt durchfahren und sowohl das Videomaterial als auch die LIDAR-Rohdaten für ihre spätere Auswertung aufgezeichnet:

Kamerabasierte Kollisionserkennung

Horizont Wasser mit Horizont

Wellen Keine Wellen

Statische Hindernisse –

Dynamisches Hindernisse Ruderboot „Petzi“, blaue Kiste als Erhöhung

Ergebnis Erkennung des Bootes „Petzi“ erfolgreich

Laserbasierte Kollisionserkennung

Horizont Wasser mit Horizont

Wellen Keine Wellen

Statische Hindernisse Ufer und Steg

Dynamisches Hindernisse Ruderboot „Petzi“, blaue Kiste als Erhöhung

Ergebnis Erkennung des Bootes „Petzi“ erfolgreich, auch das Erkennen von Bewegungen, sowohl aus kurzer und weiter Distanz, Erkennung von Ufer und Steg ebenfalls erfolgreich

Regelung

Das Testen der Regelung musste leider abgebrochen werden, da es Verzögerungen zwischen der Berechnung und der Umsetzung der Motordaten gab.

B.3.5 Praxistest am 1. September 2016

Bei diesem Praxistest soll der Fokus auf der Kollisionserkennung und Vermeidung liegen. Hierbei werden dem Laserscanner Hindernisse präsentiert, die das Wasserfahrzeug eigenständig erkennen und anschließend angemessen reagieren soll.

Datum, Uhrzeit	1. September, 9:00 bis 12:30 Uhr
Gewässer	Tweelbäker See
Standort	Am Tweelbäker See 1 26135 Oldenburg
Wetter	Sonnenschein, vereinzelt ein paar Wolken, Windstille
Tester	Michael Beering Konstantin Gebel Eike Hagena Maximilian Hipp Björn Koopmann Zahra Paya Dennis Pilny

Testfälle	Testen der Kamera und des Laserscanners für die *Kollisionserkennung*, Aufnahme von Bildern und Videos für die Abschlusspräsentation

Kollisionserkennung

Der Praxistest wurde, wie im vorherigen Test, mit der Kamera und dem Laserscanner am vorderen Bereich des Bootes durchgeführt. Es sollten verschiedene Hindernisse durch die Systeme aufgezeichnet und erkannt werden. Um ein dynamisches Hindernis zu simulieren, wurde ein weiteres Boot genutzt.

Während des Praxistests wurden neben einigen weiteren Testfällen insbesondere das folgende Szenario durchgeführt:

Testfall 1: Einige Hindernisse bei mittlerer Geschwindigkeiten

Nr. P1-S1	Detektion verschiedener Hindernisse
Testobjekt	Onboard-System (I1)
Eingabedaten	Mission: `testmission1.mission` Fernsteuerung: inaktiv Geschwindigkeit: $0 \leq v \leq 4$ Hindernisse: keine
Vorbedingung	Das System ist inaktiv.
Erwartete Reaktion	Das Gesamtsystem erkennt die bereitgestellten Hindernisse und weicht ihnen eigenständig aus.
Nachbedingung	Das System ist inaktiv.
Bemerkungen	10 m-Sichtweite des Lidar-Systems

Nr. P1-S1-A	Auswertung des Testfalls *P1-S1* (Systemtest)
Datum, Uhrzeit	1. September 2016, 9:40 bis 10:45 Uhr
Vorbedingung	erfüllt
Tatsächliche Reaktion	Das Gesamtsystem erkennt die Hindernisse und weicht ihnen eigenständig aus.
Nachbedingung	erfüllt
Artefakte	–

Bemerkungen	Erhöhung der Sichtweite auf 20 m

B.3.6 Praxistest am 15. September 2016

Bei dem sechsten und vorerst letzten Praxistest am 15. September sollte der Fokus auf die Kollisionserkennung und -vermeidung gelegt werden. Hierbei wurden den Sensorsystemen verschiedene Hindernisse präsentiert, die selbstständig erkannt und umfahren werden sollten. Eine ausführliche Beschreibung der einzelnen Testfälle ist im Kapitel 8 enthalten.

Datum, Uhrzeit	15. September, 9:00 bis 11:15 Uhr
Gewässer	Tweelbäker See
Standort	Am Tweelbäker See 1 26135 Oldenburg
Wetter	Sonnenschein, vereinzelt ein paar Wolken, mäßiger Wind
Tester	Michael Beering Konstantin Gebel Eike Hagena Maximilian Hipp Björn Koopmann Zahra Paya Dennis Pilny Peter Tank
Testfälle	Testen der Kamera und des Laserscanners für die *Kollisionserkennung*, Aufnahme von Bildern und Videos für die Abschlusspräsentation

Anhang C

Handbücher

C.1 Handbuch für die Onshore-Software

Die Onshore-Software dient der Planung und Durchführung einer Mission mit der Bootsplattform. Im Folgenden wird die Benutzeroberfläche der Bedienungssoftware beschrieben.

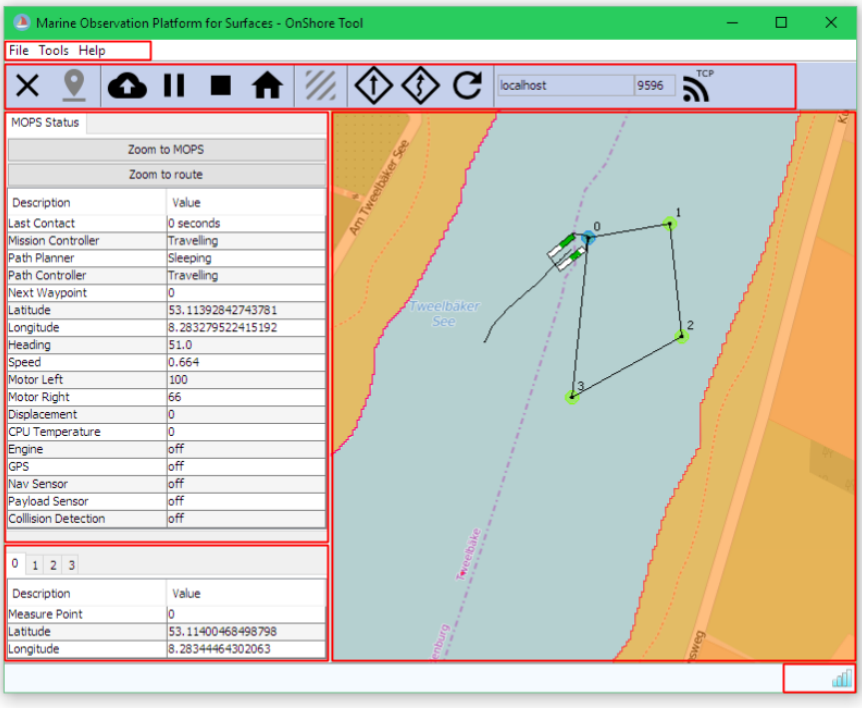


Abbildung C.1: Onshore-Software in der Gesamtansicht

C.1.1 Menü

Das Menü der Onshore-Software fasst die wichtigsten Funktionen zusammen, die für die Verwaltung und die Konfiguration von Missionen zuständig sind. In diesem Abschnitt werden die einzelnen Bestandteile der in Abb. C.1 dargestellten Benutzeroberfläche schrittweise erläutert.

File

Unter diesem Menüpunkt sind die Funktionen zur Verwaltung von Missionen aufgelistet.

New Erstellen einer neuen Mission. Nicht gespeicherte Änderungen gehen verloren.

Open Über einen Dialog lässt sich eine bereits gespeicherte Mission wieder öffnen.

Save Speichern einer Mission, um später an dieser weiterzuarbeiten. Über einen Dialog lässt sich Speicherort und Dateiname festlegen.

Export Danger Zones Über einen Dialog können generierte Gefahrenzonen exportiert werden.

Import Danger Zones Über einen Dialog lassen sich Gefahrenzonen in die aktuelle Mission importieren.

Exit Verlassen der Onshore-Software. Nicht gespeicherte Änderungen gehen verloren.

Tools

Unter dem Menüpunkt Tools lassen sich Werkzeuge für die Konfiguration einer Mission finden. Im Folgenden werden diese anhand von Abbildungen näher erläutert.

Compass Calibration Mithilfe dieses Werkzeugs ist es möglich das tatsächliche Heading des Bootes in die Onshore-Software zu übertragen. Dies ist nötig um die Ausrichtung des Bootes in die Missionsplanung einzubeziehen. Abbildung C.2 zeigt den zugehörigen Dialog. Nachdem das Boot entsprechend der Himmelsrichtung ausgerichtet wurde, kann dies über den jeweiligen Button in der Onshore-Software bestätigt werden. Nachdem dieser Vorgang für jede der vier Himmelsrichtungen wiederholt wurde ist die Kompasskalibrierung abgeschlossen.

Configuration Abbildung C.3 zeigt die verschiedenen Einstellungsmöglichkeiten der Onshore-Software bezüglich der Regelungparameter. Aber

auch andere Einstellungen, die die Onshore-Software betreffen können hier vorgenommen werden.

Configuration Xbee Da die Kommunikation über XBee funktioniert, muss ein entsprechendes Empfangsgerät verfügbar sein. Über diesen Dialog kann der zu nutzende Port für eine XBee-Antenne festgelegt werden. Abbildung C.4 zeigt, wie dieser ausgewählt und gespeichert werden kann.

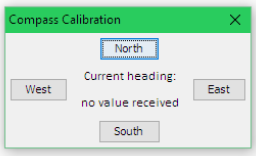


Abbildung C.2: Kalibrierungsdialog für den Kompass

Help

Unter diesem Menüpunkt können allgemeine Informationen zur Onshore-Software nachgelesen werden. Die beiden Auswahlmöglichkeiten werden im Folgenden näher beschrieben.

Hilfe Unter dieser Auswahl ist ein Dialog zu finden, der einen Verweis auf diese Dokumentation enthält und eine Emailadresse für eventuelle Fragen bereitstellt.

About Hier lassen sich Informationen zu dem Zweck und den Verantwortlichen der Onshore-Software finden.

C.1.2 Toolbar

Die Toolbar enthält verschiedene Befehle die sowohl der Bedienung der Plattform dienen, in Abb. C.5 gezeigt.

XBee Verbindung aufbauen/abbrechen Aufbau einer Verbindung zum Boot per XBee. Falls bereits eine Verbindung besteht, kann über dieses Icon die Verbindung wieder abgebrochen werden.

Auswahl Home Position Mithilfe dieses Befehls kann ein ausgewählter Messpunkt zur Home Position umgewandelt werden.

Mission hochladen Besteht eine Verbindung zum Boot, so kann über diesen Befehl die aktuelle Mission auf die Plattform geladen werden.

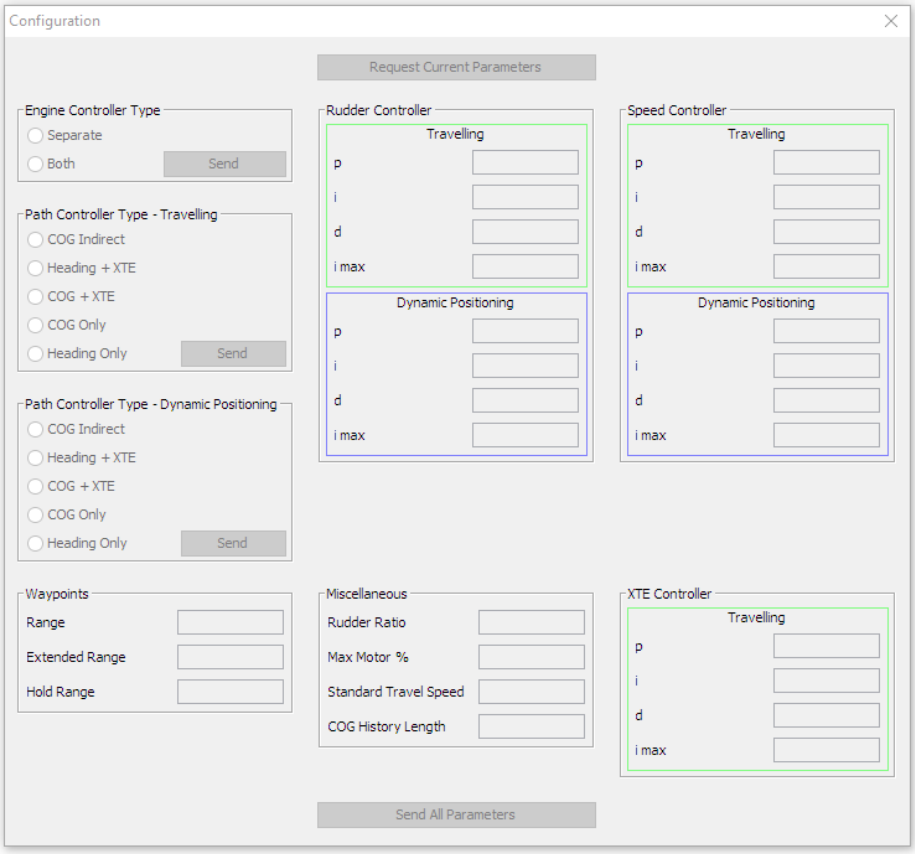


Abbildung C.3: Konfigurationsdialog für die Systemparameter

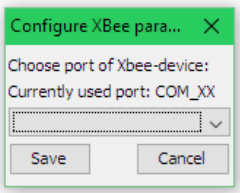


Abbildung C.4: Konfigurationsdialog für die XBee-Konfiguration

Play/Pause Besteht eine Verbindung zum Boot und wurde eine Mission hochgeladen, so kann über diesen Befehl eine Mission gestartet werden, oder falls eine Mission läuft diese pausieren.

Abbildung C.5: Toolbar der Onshore-Software

Stop Eine bereits laufende Mission kann über dieses Icon gestoppt werden.

Return To Home Während einer laufenden Mission wird diese abgebrochen und das Boot fährt zurück zur Home Position.

Dangerzones generieren Für den aktuell ausgweählten Kartenausschnitt werden die Dangerzones generiert. Da die Generierung eine längere Zeit in Anspruch nimmt wird das Icon während der Berechnung ausgeblendet.

Start/Stop Geradeausfahrt Eine Entwickleroption die bei Betätigung dem Boot signalisiert, dass eine Geradeausfahrt begonnen hat bzw. beendet wurde. Diese Funktion wurde für die Einstellung der Reglerparameter verwendet.

Start/Stop Kurvenfahrt Eine Entwickleroption, die bei Betätigung dem Boot signalisiert, dass eine Kurvenfahrt begonnen hat bzw. beendet wurde. Diese Funktion wurde für die Einstellung der Reglerparameter verwendet.

Neuladen der Systemparameter Bei Betätigung werden die Systemparameter in der Missionsplanungsofware auf dem Boot neu geladen.

Host Eingabefeld In das Eingabefeld wird die Hostadresse für die Kollisionserkennungssoftware eingestellt.

Port Eingabefeld In das Eingabefeld wird der Port für die Kollisionserkennungssoftware eingestellt.

TCP-Verbindung Aufbau einer TCP-Verbindung zur Kollisionserkennungssoftware. Bei einer bestehenden Verbindung werden die empfangenen Hindernisse auf der Karte dargestellt.

C.1.3 Statusinformationen

Besteht eine Verbindung zur Plattform so werden in der Tabelle links neben der Karte dargestellt. In Tabelle C.1 werden die verfügbaren Informationen aufgelistet und kurz beschrieben.

Name	*Beschreibung*
Last Contact	Anzahl der Sekunden seit dem letztem XBee-Kontakt mit dem Boot
Mission Controller	Aktueller Zustand des Missioncontrollers
Path Planner	Aktueller Zustand des Path Planners
Path Controller	Aktueller Zustand des Path Controllers
Next Waypoint	ID des nächsten Waypoints der angefahren werden soll
Latitude	Geographische Breitenposition des Bootes
Longitude	Geographische Längenposition des Bootes
Heading	Ausrichtung des Bootes
Speed	Geschwindigkeit in km/h
Motor Left	Relativer Motorwert für den linken Motor (−100 bis 100)
Motor Right	Relativer Motorwert für den rechten Motor (−100 bis 100)
Engine	Zustand der Motoransteuerung der Missionsplanung
GPS	Zustand des GPS-Moduls des Bootes
Nav Sensor	Zustand des Navigationssensors
Payload Sensor	Zustand der Payload Sensor Ansteuerung der Missionsplanung
Collision Detection	Verbindungszustand der Kollisionserkennung mit der Missionsplanung

Tabelle C.1: Statusinformationen in der Onshore-Software

C.1.4 Missionsübersicht

Ist mindestens ein Missionspunkt vorhanden, können diese in der Onshore-Software eingesehen werden. Über die Tableiste kann zwischen den einzelnen Messpunkten gewechselt werden. Tabbelle C.2 listet die verfügbaren Einträge mit der dazugehörigen Erläuterung auf.

Name	*Beschreibung*
Measure Point	Nummer des Missionspunkts
Latitude	Angabe des Breitengrades des Missionspunkts
Longitude	Angabe des Längengrades des Missionspunkts

Tabelle C.2: Missionsplanung in der Onshore-Software

C.1.5 Bedienung der Karte

Über die Karte der Onshore-Software können Missionen geplant werden, zusätzlich wird die aktuelle Position und die abgefahrene Route der Plattform dargestellt. Abb. C.6 zeigt die Simulation einer Mission. Im Folgenden werden die Bedienelemente der Karte beschrieben.

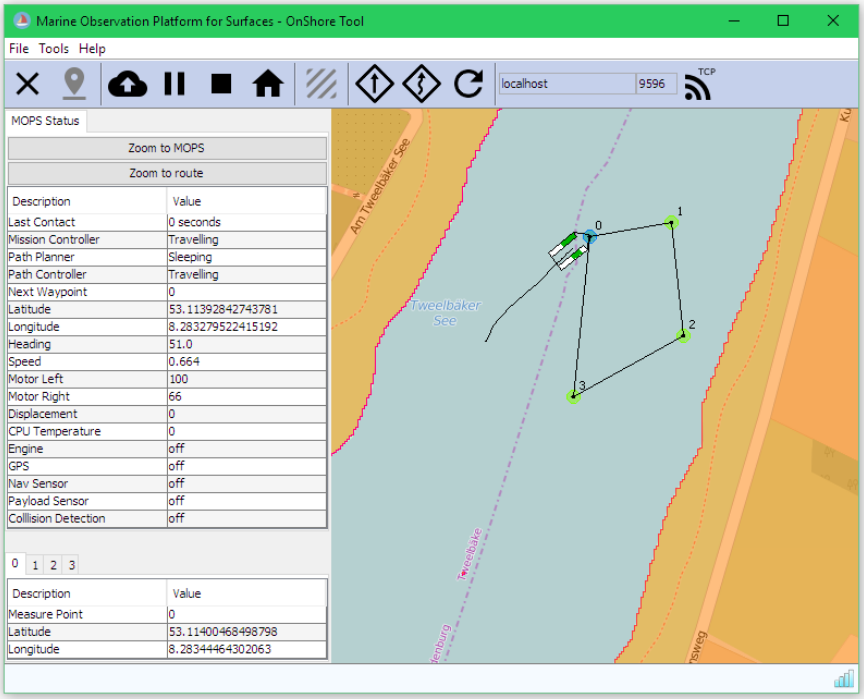


Abbildung C.6: Laufende Simulation mit generierten Gefahrenzonen

Navigation Mithilfe der Maus kann über die Karte durch drücken und ziehen des Cursors die Ansicht der Karte verschoben werden. Über das Mausrad kann das Zoomtiefe eingestellt werden.

Erstellen von Messpunkten Bei gedrückter STRG-Taste und Klick auf die Karte können Messpunkte gesetzt werden. Der erste gesetzte Messpunkt wird automatisch als Home Position definiert.

Erstellen von Gefahrenzonen Bei gedrückter SHIFT-Taste und Klick auf die Karte wird ein Eckpunkt für eine Gefahrenzone erstellt. Sobald die

SHIFT-Taste losgelassen wird, wird aus den Eckpunkten eine Gefahrenzone generiert und dargestellt.

C.2 Bisheriges Handbuch für den Bootsaufbau

Das folgende Handbuch beschreibt den Aufbau des Bootes. Es wurde aus der Dokumentation der Projektgruppe Mops ii *entnommen [53, S. 84 ff.]. Verweise auf das alte SVN-Repository wurden gestrichen.*

Dieser Abschnitt befasst sich mit der Inbetriebnahme des Mops ii. Im folgenden finden Sie die nötigen Schritte in einer Bauanleitung.

Aufladen der Batterie Zuerst sollte überprüft werden, ob die Batterie voll aufgeladen ist, damit die maximale Fahrzeit des Mops ii ausgeschöpft werden kann. Die Batterie zeigt über die runde Kontrollanzeige den Ladezustand an. Dabei umfasst die Batterie drei Status:

- Green Eye - Charge over 65 % , BATTERY PROPERLY CHARGED
- Black Eye - Charge under 65 % , DISCHARGED BATTERY NEEDS TO BE RECHARGED
- Electrolyte level too low, BATTERY MUST BE REPLACED

Beim Aufladen der Batterie sollte die Funktion Pb aufladen und 5A Stromstärke eingestellt werden. Das Ladegeräte sollte im Regelfall die Batterie erkennen und automatisch die Spannung einstellen. Es muss dennoch beachtet werden, dass die Batterie nicht mit mehr als 14.4 V geladen wird, da aus diesem Vorgang Beschädigungen der Batterie resultieren.

Es sollte bedacht werden, dass der Ladevorgang mehrere Stunden in Anspruch nimmt. Es empfiehlt sich die Batterie einen Tag vorher zu kontrollieren und aufzuladen. Beim Entladen der Batterie muss darauf geachtet werden, dass die Spannung niemals unter 11 V fällt.

Zusammenbau des Bugsegments *Liste der benötigten Teile:*

- 1× Batteriebox (blau)
- 1× obere Abdeckung
- 1× unteres Gehäuse
- 4× Gummihalterungen
- 1× Batteriesteckverbindung mit zwei Klemmen

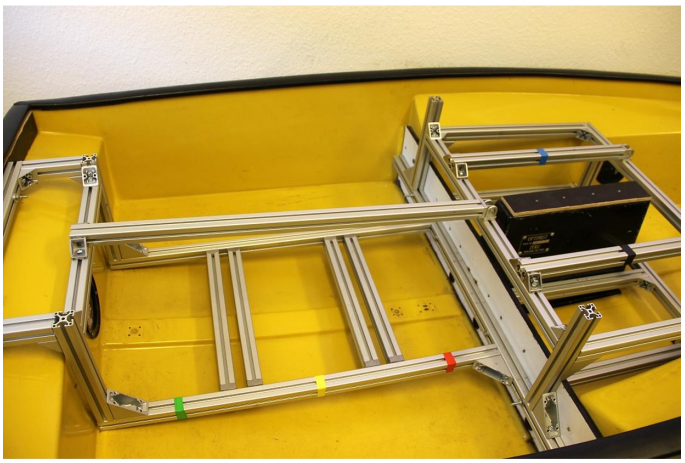


Abbildung C.7: Alukonstruktion

- 1× Vetus Marine Batterie 12 V 108 AH
- 1× M6 Sechskantmutter DIN934, A2
- 1× Sechskantschraube DIN933, A2 Edelstahl, M6 × 60 mm

Der erste Schritt beim Zusammenbau des Bugsegments ist die Zusammensetzung der Batteriebox (blau). Dazu wird die Batterie in das untere Gehäuse (mit der blauen Markierung) eingesetzt. Sollte die Kontrollanzeige der Batterie nicht grün sein, empfehlen wir das Aufladen der Batterie (siehe Kapitel Aufladen der Batterie).

Als nächstes wird die rote Klemme der Steckverbindung an den Pluspol und die schwarze Klemme an den Minuspol der Batterie geklemmt. Danach wird die obere Abdeckung auf das untere Gehäuse gesteckt. Es muss darauf geachtet werden, dass das obere Gehäuse alle Seiten abdeckt und richtig befestigt wird. Dazu müssen die 4 Gummihalterungen an den Haken des unteren Gehäuses und den Noppen des oberen Gehäusen angebracht werden.

Nach dem korrekten Schließen der Box kann diese vorne rechts bei der blauen Markierung eingesetzt werden. Danach wird die Querstange über die Batteriebox umgeklappt. Die Stange befindet sich nun auf der Rahmenstange in der Mitte des Rumpfs. Hier wird nun die M6 Sechskantschraube durch die Öffnung der Querstange und die Winkel an der Rahmenstange geführt und mit einer Sechskantmutter M6 verschraubt.

Als letztes wird die Batteriesteckverbindung durch die Öffnung im Brett in der Mitte des Rumpfs gezogen, um diese Steckverbindung später mit der Box 1 Leistungselektronik (rot) zu verbinden.

Abbildung C.8: Blaue Batteriebox

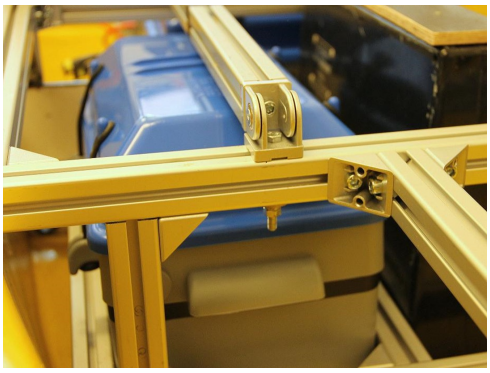


Abbildung C.9: Schwarze Otterbox mit Gewichten

Zusammenbau des Hecksegments *Liste der benötigten Teile:*

- 1× Box 1 Leistungselektronik (rot)
- 1× Motorcontroller Sabertooth 2×60
- 1× Relais
- 1× Steckverbindung mit Klemmen
- 2× Steckverbindung
- 1× Box 2 Raspberry Pi (gelb)
- 1× Arduino Uno
- 1× Raspberry Pi
- 1× Box 3 BeagleBone (grün)

- 1× BeagleBone Black
- 1× Cape für das BeagleBone
- 1× XBee Pro 868
- 1× Sensorboard 9DOF Razer IMU
- 1× GPS Navilock NL-652ETTL
- 1× M6 Sechskantmutter DIN934, A2
- 1× Sechskantschraube DIN933, A2 Edelstahl, M6 × 60 mm

In Box 1 Leistungselektronik (rot) befinden sich die benötigten Teile für die Steuerung der Motoren einschließlich der Not-Aus-Komponente. Über die Eingänge M1a, M1b und M2a, M2b sind die Motoren angeschlossen. Der Not-Aus liegt direkt an den Eingängen B+ und B-. Über die Eingänge S1 und S2 werden die PWM-Werte geschrieben. Box 1 Leistungselektronik (rot) muss nun an die rot markierte Stelle im MOPS angebracht werden.

Abbildung C.10: Rote Elektronikbox

In Box 2 Raspberry (gelb) befinden sich die Komponenten, um die Sensordaten aufzunehmen und zu verarbeiten. Das Raspberry steuert das Arduino Uno über den USB-Eingang und speichert die Sensordaten. Über das Ethernet ist das Raspberry mit dem BeagleBone verbunden, darüber hinaus stellt das Raspberry das WLAN-Signal. Es verfügt über verschiedene Kontroll-LEDs.

Kürzel	*Diode*	*Farbe*	*Beschreibung*
ACT	D5	grün	SD Card Access
PWR	D6	rot	3.3 V Power present
FDX	D7	grün	Full Duplex, LAN connected
LNK	D8	grün	Link Activity
100	D9	gelb	100 M/bit Lan connected

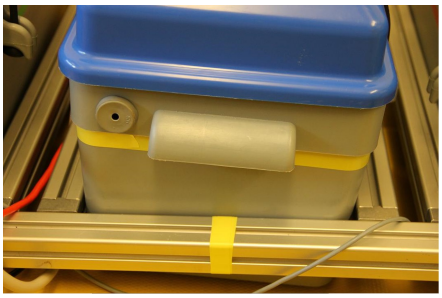


Abbildung C.11: Gelbe Elektronikbox

Das Arduino Uno stellt die Anfragen an die Sensoren über das Bus-System. Auf dem Arduino Uno befindet sich ein Cape mit dem die Stromversorgung geregelt wird. Es verfügt darüber hinaus über verschiedene Kontroll-LEDs. Die Box kann nun geschlossen und an der vorgesehenen Stelle (gelb) platziert werden.

LED	*Verwendung*
LED 1 RED	Stromversorgung
LED 2 RED	
LED 3 BLUE	Befehl empfangen, Serial / USB (aufblinken)
LED 4 WHITE	
LED 5 BLUE	Antwort empfangen, SDI-12 (toogle)
LED 6 WHITE	

In Box 3 BeagleBone (grün) befindet sich die Hardware auf der die Onboard-Software läuft und den MOPS II mit einem geeigneten Algorithmus autonom manövrieren lässt. Mit Hilfe des aufgesteckten Capes lassen sich das GPS, das Sensorboard und das XBee verwenden. Die Antenne ist mit einem RP-SMA Stecker mit dem XBee-Modul verbunden. Hierüber werden per Funk Steuersignale wie z.B. Missionspunkte zum MOPS 2 geschickt. Das Sensorboard von Sparkfun steckt über die folgende Pin-Verbindung auf dem BeagleBone.

Anschluss	*Kabel*
x1	braun
o1	kein Kabel
x2	weiß
x3	grün
x4	gelb
o2	kein Kabel

Außerdem ist das GPS-Modul Navilock NL-652ETTL mit dem BeagleBone verbunden. Die Box sollte nun geschlossen und an der grünen Stelle gestellt werden.

Abbildung C.12: Grüne Elektronikbox

Anbringen der Motoren *Liste der benötigten Teile:*

- 2× Rhino-VX-34
- 1× Stabilisierungsprofil
- 2× Schiffsschrauben
- 1× Schlüssel für die Schiffsschrauben

Die Rhino-VX-34 werden am Heck des MOPS II an der Markierung von außen fest angeschraubt. Bevor die Motoren nun in der Höhe verstellt werden, müssen die Schiffsschrauben angebracht bzw. kontrolliert werden. Dazu wird die Schiffsschraube mit der Einkerbung nach außen auf die Motoren gesteckt und mit Hilfe des Schlüssels befestigt. Sind die Schiffsschrauben bereits verbaut, sollte dennoch kontrolliert werden, ob diese fest sitzen.

Nun werden die Motoren in ihrer Höhe verstellt. Dazu muss die Schraube an der Stange gelöst werden und bis auf die Markierung eingestellt werden. Nachdem beide Motoren in ihrer Höhe verstellt und befestigt sind. Wird die Ausrichtung der Motoren mit Hilfe des roten Druckknopfs auf 90 Grad zum MOPS II eingestellt (unterste Rastereinstellung).

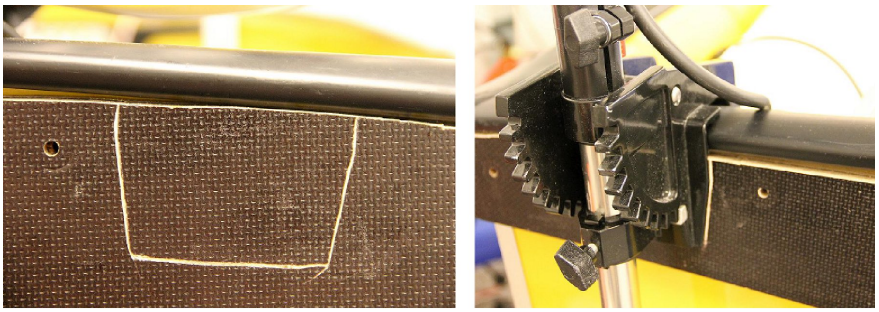


Abbildung C.13: Befestigung der Motoren

Verkabelung der Boxen *Liste der benötigten Teile:*

- 2× M6 Sechskantschraube DIN912, A2
- 2× 2x Nutenstein N6 mit Federkugel, stahlverzinkt
- 1× Stromkabel blau - rot

Um den MOPS II einsatzbereit zu machen, werden nun die Motoren mit den Elektronikboxen und der Batteriebox verbunden. Dazu wurde für jede

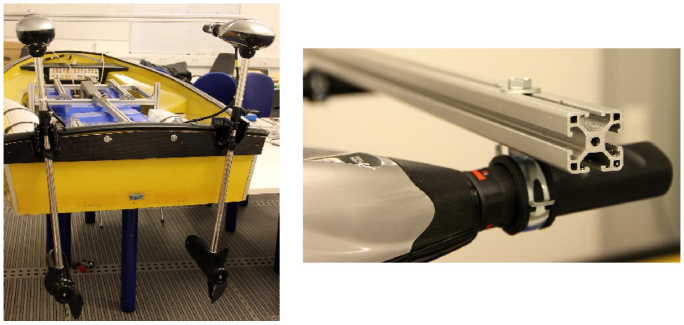


Abbildung C.14: Befestigung des Stabilisierungsprofils

Steckverbindung eine eindeutige Farbe gewählt, damit keine Störungen auftreten können. Als erstes sollte nun der Not-Aus-Schalter gezogen werden, um die Gefahr eines laufenden Motors abzuwenden. Dazu wird einmal an der Schnur des Schalters gezogen.

Komponente 1	*Komponente 2*	*Steckverbindung*
Motor gelb	Box 1 Leistungselektronik (rot)	gelb - gelb
Motor grün	Box 1 Leistungselektronik (grün)	grün - grün
Batterie blau	Kabel blau	blau - blau
Kabel rot	Box 1 Leistungselektronik (rot)	rot - rot

Der Not-Aus-Schalter wird nun an dem Aluminiumprofil zur Sicherung der Boxen 1, 2 und 3 mit Hilfe von zwei M6 Sechskantschrauben und zwei Nutensteine N6 von oben befestigt. Nachdem alle Boxen verkabelt sind, kann nun der Not-Aus-Schalter wieder betätigt werden, um den Mops II scharf zu stellen. Ab nun könnte der Mops II theoretisch Missionen entgegennehmen.

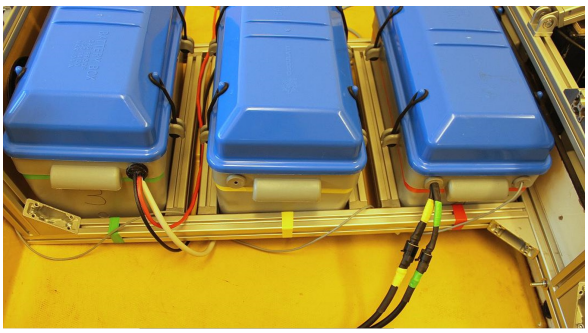


Abbildung C.15: Verkabelung Steuerbord

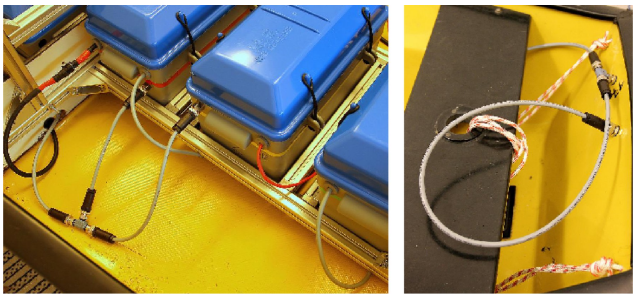


Abbildung C.16: Verkabelung Backbord

Anbringen der Auftriebskörper *Liste der benötigten Teile:*

- 4× Befestigungsgurte
- 2× Auftriebskörper

Zum Befestigen der Auftriebskörper werden beide Befestigungsgurte benötigt. Hierzu werden die Gurte durch den unteren Rahmen der Bugkonstruktion direkt zwischen zwei Aluminiumstreben und durch Halterungen an der Innenseite im Rumpf geführt. Anschließend werden die Gurte zusammengesteckt wie in der unteren Abb. C.17 zu sehen.

Abbildung C.17: Befestigung der Schwimmkörper

Befestigung des Sensorkäfigs *Liste der benötigten Teile:*

- 1× Sensorkäfig
- 2× Seile

Zum Befestigen des Sensorkäfigs werden die Seile durch die Öffnungen am Heck des Rumpfs gezogen und mit einem Knoten befestigt. Danach werden diese Seile durch die Öffnungen am oberen Rand der Profile des Sensorkäfigs geführt und mit einem Knoten befestigt. Durch die Anordnung der beiden Seile wird ein Rotieren des Sensorkäfigs und damit ein Abreißen dessen verhindert.

Anbringen der Plane *Liste der benötigten Teile:*

- 1× Plane
- 1× Seil
- 1× Spannhaken

Zunächst wird die Plane mit dem Loch zur Heckseite ausgelegt und dann mit dem Loch zuerst über die Antenne gestülpt. Danach kann die Plane komplett über den Rumpf gelegt werden.

Um die Plane zu fixieren wird das Seil von außen um die Plane gelegt und mit einem Spannhaken zusammengehalten (siehe Abb C.18).

Abbildung C.18: Befestigung der Plane

C.3 Bisheriges Handbuch für die Inbetriebnahme

An dieser Stelle wird das Handbuch für die Inbetriebnahme aufgeführt, welches in der Dokumentation der Projektgruppe MOPS III *[31, S 80ff] zu finden ist.*

Hier soll kurz erklärt werden, wie man die Plattform in den betriebsbereiten Zustand versetzt, d.h. welche Einstellungen vorgenommen und welche Verbindungen gesteckt werden müssen. Dabei wird dem Verlauf der Spannungsversorgung entsprechend vorgegangen, d.h. von der Leistungselektronik (12 V), über die Arduino-Box (12 V → 5 V), zum Raspberry Pi (5 V). Danach

wird die Systemelektronik angeschaltet, die Plattform eventuell kalibriert und zu Wasser gelassen.

Für den (unwahrscheinlichen) Fall das die Alu-Rahmenkonstruktion des Bootes noch zusammengebaut werden muss, sollte sich hier an der Anleitung in der Dokumentation der Projektgruppe MOPS II gehalten werden (Kapitel C.2). Außerdem kann die Belegung der Stecker/Kabel im Notfall in Kapitel B nachgesehen werden. In der folgenden Anleitung wird davon ausgegangen, das sich bereits alle Komponenten (bspw. die Motoren) im bzw. am Boot befinden.

Leistungselekronik-Box

1. Zunächst ist zu prüfen, ob die Switches der Box deaktiviert sind, dazu muss sich der Main-Switch in OFF-Position befinden und die Kappe des Kill-Switch muss abgezogen sein (vgl. Abb. C.28, links oben).

2. Danach muss geprüft werden, ob die Stecker der Platine mit den entsprechenden Buchsen verbunden sind, dazu wurden Stecker und Buchse jeweils mit einem Etikett versehen. Außerdem sollte ein Relais in dem entsprechenden Sockel stecken (vgl. Abb. C.28, links unten).

3. Nun ist zu prüfen, ob der Sabertooth ordnungsgemäß angeschlossen ist und die DIP-Switches in der richtigen Position stehen (siehe Steckverbindung Sabertooth). Außerdem sollte der Sabertooth im Zweifelsfall mit der notwendigen Konfiguration versehen werden (siehe Konfiguration Sabertooth).

4. Weiterhin sollte geprüft werden, ob der Stecker für den NMEA-Bus ordnungsgemäß angebracht ist. Nach befolgen der bisherigen Schritte sollte es in der Box nun wie im rechten Bildausschnitt aussehen.

5. Nun kann auch der Motorstecker geschlossen werden.

Arduino-Box

1. Es ist zunächst zu prüfen, ob der NMEA-Bus ordnungsgemäß verbunden ist.

2. Weiterhin sollten beim Sensor Arduino die Spannungsversorgung (BAT) und der USB-Stecker für den Raspberry Pi gesteckt sein. Der Arduino sollte im Zweifelsfall mit der Software `ArduinoPayload.ino` bespielt werden.

3. Beim Motor Arduino sind die Stecker für die 12 V-Spannungsversorgung (BAT), die 5 V-Spannungsversorgung für den HUB/RPi (HUB),

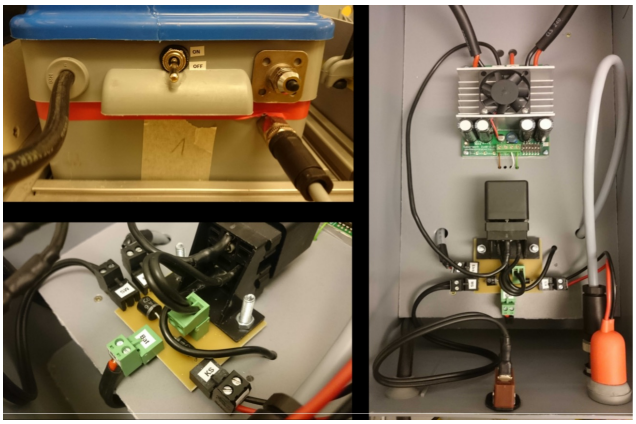


Abbildung C.19: Leistungselektronik

Abbildung C.20: Arduino-Box

die Debug-LEDs (LEDs), die Motor-Steuersignale (Motor), die Steuersignale der RC-Fernbedienung (RC-Receiver) sowie der USB-Stecker für den Raspberry Pi zu prüfen. Der Arduino sollte im Zweifelsfall mit der Software `ArduinoMotor.ino` bespielt werden.

4. Außerdem sollte geprüft werden, ob der RC-Receiver ordnungsgemäß verbunden ist.

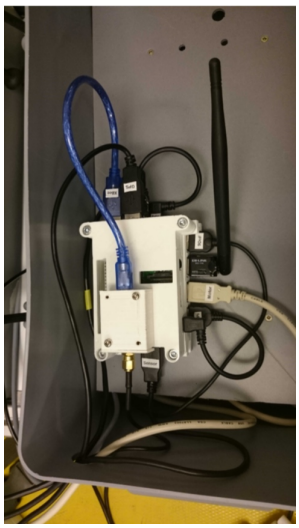


Abbildung C.21: Raspberry-Box

Rasperry-Box

1. Es ist zunächst zu prüfen, ob die Spannungsversorgung (BAT) im Power-Port des HUBs steckt.

2. Als nächstes ist zu überprüfen, ob beim Raspberry Pi am Power-Port eine USB-Verbindung zum HUB besteht. Außerdem sollte noch eine Verbindung von einem RPi USB-Port zum Master-Port des HUBs bestehen.

3. Nun ist zu prüfen, ob die folgenden USB-Stecker in Ports des HUBs oder RPi stecken:

   - GPS
   - XBee
   - 9DoF
   - WLAN
   - Motor
   - Sensor

4. Im Zweifelsfall sollte die Konfiguration des Raspberry durchgeführt werden, dazu wie in Raspberry Pi Konfiguration beschrieben vorgehen.

Abbildung C.22: Kontrollleuchten

Anschalten

1. Wurden die bisherigen Schritte pflichtbewusst befolgt, kann nun ohne Bedenken die Batterie angeschlossen werden, sowie die Systemelektronik mit Hilfe des Main-Switch angeschaltet werden.

2. Nun sollten die Status-LEDs innerhalb der Boxen überprüft werden, diese sollten wie in nebenstehendem Bild (rote Kreise) aussehen. Außerdem sollten die Debug-LEDs zunächst Blau und nach ungefähr 2-3 Minuten Grün leuchten.

Kalibrierung Die Kalibrierung ist persistent. Falls dennoch eine Neukalibrierung erwünscht ist:

1. MOPS Richtung Norden ausrichten und in Onshore-Software im Kalibrierungsdialog den Nord-Button drücken

2. MOPS Richtung Osten ausrichten und in Onshore-Software im Kalibrierungsdialog den East-Button drücken

3. MOPS Richtung Süden ausrichten und in Onshore-Software im Kalibrierungsdialog den South-Button drücken

4. MOPS Richtung Westen ausrichten und in Onshore-Software im Kalibrierungsdialog den West-Button drücken

Stapellauf

1. Für den Stapellauf ist zunächst die Stellung der Motorengriffe zu überprüfen, diese geben an wie viel Leistung den Motoren zur Verfügung steht, hierbei ist Position 5 (volle Leistung) zu wählen. Die Griffe müssen (mehr oder weniger gut) gegen verdrehen gesichert werden, in dem

sie mit Hilfe eines Aluprofils an den beiden Manschetten verbunden werden.

2. Die Motoren müssen zum Transport mit dem Slipwagen zunächst hochgeklappt sein und sollten im Zweifelsfall auch erst im Wasser wieder abgesenkt werden. Je nach dem wie zu Wasser gelassen wird, könnte es sonst zu Beschädigungen an den Motoren kommen.

3. Beim zu Wasser ist man stark von den Gegenbenheiten vor Ort abhängig. Am einfachsten ist die Prozedur, wenn am Einsatzort eine Slipanlage vorhanden ist, dann kann der MOPS relativ einfach mit dem Slipwagen ins Wasser gefahren werden. Bei genügend Tiefe löst der MOPS sich aus dem Slipwagen und wäre somit für die weiteren Schritte bereit. Sollte keine Slipanlage vorhanden sein, kann der MOPS auch ohne weiteres ins Wasser getragen werden, hierbei ist es allerdings unbedingt darauf zu achten, dass es zu keinen Beschädigungen kommt.

4. Befindet der MOPS sich nun im Wasser, müssen zunächst die Motoren in ihre Einsatzposition gebracht werden. Außerdem muss jetzt die Motorregelung noch mit Hilfe des Kill-Switch aktiviert werden, dazu einfach die Kappe aufstecken.

5. Damit der MOPS nun auch manövrierfähig wird, muss die RC-Fernbedienung nun zumindest einmal AN und wieder AUS geschaltet werden. Es bietet sich jedoch an, den MOPS zunächst ferngesteuert ein Stück von der Landungszone zu entfernen, um mögliche Gefahren zu vermeiden.

C.4 Neues Handbuch für den Bootsaufbau

Folgend wird der neue Aufbau des Bootes beschrieben. Alle Änderungen, die vorgenommen wurden, werden an dieser Stelle beschrieben. Der Schiffskörper, die Anbringung der Motoren, sowie die Aluprofilkonstruktion sind zum Großteil gleich geblieben (vgl.C.2) und werden daher nicht weiter erläutert.

C.4.1 Batterie

Die Batterie kann für das Aufladen im Boot bleiben, da die Kabelverbindung für die Klemmen verlängert wurden. Dafür muss lediglich das Ladegerät (Waeco PerfectCharge) an die dafür vorgesehenen Anschlüsse angebracht werden. Dazu müssen die Klemmen für die Stromversorgung der Hardware des Bootes abgenommen werden. Die Pole der Batterie sind eindeutig markiert.

Die Batterie sollte bei jedem Praxistest vollständig aufgeladen sein. Der Aufladevorgang kann mehrere Stunden dauern. Die verschiedenen Zustände der LED, die den Ladestatus anzeigt, können in der Tabelle C.7 abgelesen werden. Leuchtet die LED dauerhaft grün, ist die Batterie voll geladen.

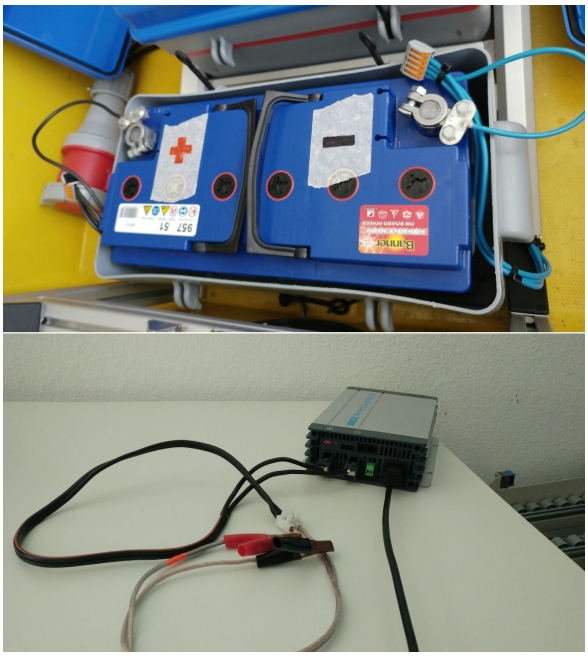


Abbildung C.23: Batterie

Phase 1	Schnelles Blinken (Orange)
Phase 2	Langsames Blinken (Orange)
Phase 3	Dauerleuchten (Orange)
Phase 4	Dauerleuchten (Grün)

Tabelle C.7: Phasen des Ladestatus

C.4.2 Anordnung der Otterboxen

Im Folgenden ist die Anordnung der Otterboxen vom Heck bis zum Bug dargestellt:

- In der blauen Box befindet sich die Batterie
- In der roten Box ist die Leistungselektronik
- In der gelben box ist das Raspberry Pi für die Kamera und das Lidar
- In der grünen Box befinden sich ein weiteres Raspberry Pi, ein Usb-Hub, das Arduino und das XBee-Modul

- Das GPS-Modul befindet sich nicht mehr in einer Box, sondern weiter oberhalb (siehe C.24)

Die Abbildung C.25 zeigt die farblich markierten Otterboxen, während sich das Heck auf der Rechten Seite des Bildes befindet.

Abbildung C.24: GPS-Mouse

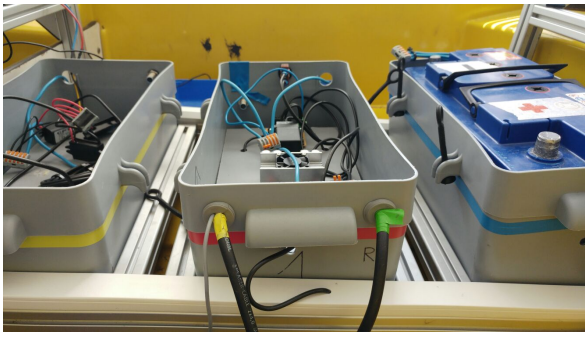


Abbildung C.25: Anordnung der Otterboxen

C.4.3 Veränderungen am Schiffskörper

Um die Kabel der blauen roten und gelben Box zu schützen, wurden Kabelkanäle angebracht. Die Deckel können mit etwas Kraft abgehoben und wieder aufgesteckt werden. Für die Kamera und den Scanner wurde ein Mast an das Gerüst angebracht. Um diesen in einer stabilen Position zu halten, wurde er mehrfach am Schiff befestigt. Der An- und Ausschalter aus Abbildung C.26 ist nur für Lidar und für die grüne Box. Ein Notaus-Schalter sorgt für das Lahmlegen des Bootes im Notfall. Dieser ist im Bild C.27 zu sehen.

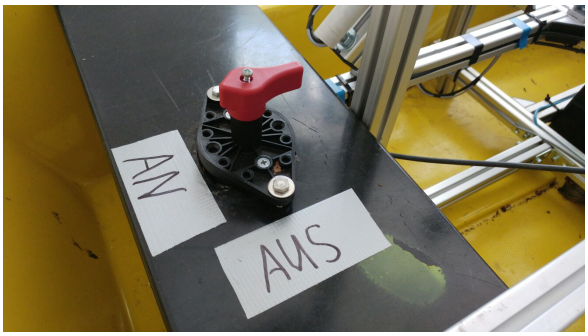


Abbildung C.26: An-Aus-Schalter

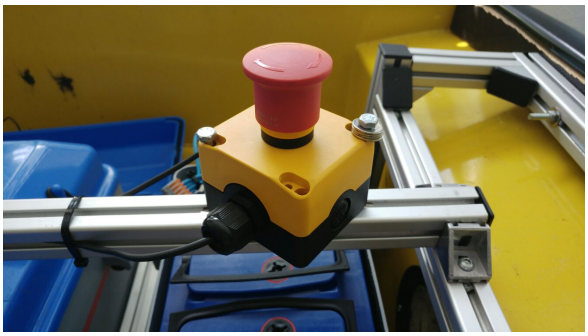


Abbildung C.27: Notaus-Schalter

C.5 Neues Handbuch für die Inbetriebnahme

Im zweiten Abschnitt soll erklärt werden, wie das Boot in den betriebsbereiten Zustand versetzt wird. Hierbei ist auf die richtige Verkabelung und Anbringung der Hardware zu achten.

C.5.1 Batterie

Die Batterie sollte vor jedem Gebrauch aufgeladen werden. Danach können die Batterieklemmen vom Boot angebracht werden.

C.5.2 Leistungselektronik

Die Leistungselektronik ist so verbaut, das an dieser keine Änderungen und Vorbereitungen vorgenommen werden müssen. Der Main-Switch auf der Rückseite kann dauerhaft auf OFF stehen, da dieser von uns deaktiviert wurde. Auch der Kill-Switch aus der vorherigen Gruppe wurde beseitigt und durch

einen Not-Aus-Schalter ersetzt. Die Stecker der Platinen und Buchsen sollten wie im nachfolgenden Bild C.28 aussehen.

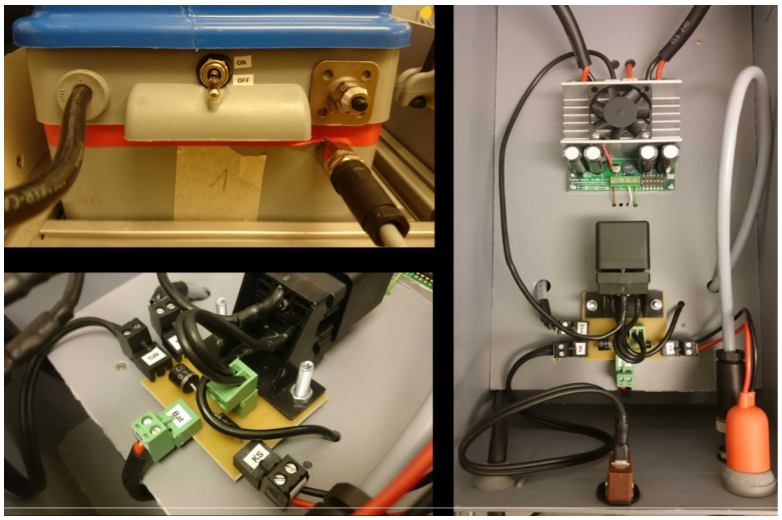


Abbildung C.28: Leistungselektronik

C.5.3 Kamera, Lidar und Router

Die Kamera und das Lidar sollten vor jedem Transport abmontiert werden. Hierfür reicht es bei der Kamera die Schraube unterhalb der Halterung zu lösen. Beim Lidar sollten nicht die Schrauben am Scanner gelöst werden, sondern die Verbindungsstücke zum Mast. Dies stellte sich als erheblich einfach bei der Montage heraus. Beim Router reicht es aus wenn die Antennen für den Transport umgeklappt werden.

C.5.4 Grüne Hardwarebox

Die grüne Box sollte wie auf dem Bild C.29 zu sehen ist, verkabelt sein. Die einzelnen Steckverbindungen werden im Kapitel C.6 näher erläutert.

C.5.5 Gelbe Hardwarebox

Auch die gelbe Box sollte wie auf folgenden Bild C.30 zu sehen ist, verkabelt sein. Die einzelnen Steckverbindungen sind im Kapitel C.6 dargestellt.

C.5.6 Anschalten

Ist alles im Boot ordnungsgemäß verkabelt und die Hardware fest angebracht, ist das Boot betriebsbereit. Der Schalter am Bug kann nun eingeschaltet wer-

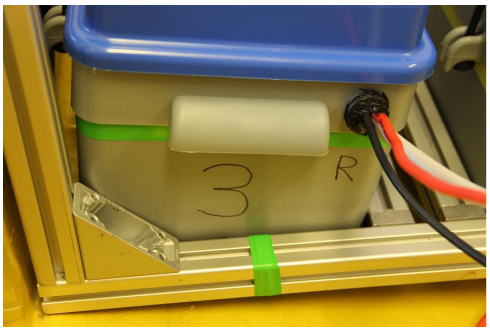


Abbildung C.29: Grüne Hardwarebox

Abbildung C.30: Gelbe Hardwarebox

den, der Not-Aus darf nicht gedrückt sein. Danach sollten die LEDs blinken und die Debug-LEDs am Mast nach kurzer Zeit von Blau auf Grün springen.

C.5.7 Kalibrierung

Sollte eine neue Kalibrierung erforderlich oder erwünscht sein, funktioniert dies wie schon im alten Handbuch Kapitel C.3 unter dem Abschnitt Kalibrierung beschrieben.

C.5.8 Stapellauf

Auch der Stapellauf hat sich gegenüber der vorherigen Gruppe nicht verändert und kann im alten Handbuch Kapitel C.3 unter dem Abschnitt Stapellauf nachgelesen werden.

C.6 Steckverbindungen

C.6.1 Rote Box

In der roten Box C.31 wurde am Sabertooth nichts geändert, die Steckverbindungen können also von der dritten Projektgruppe übernommen werden. Das Relaisboard besitzt acht Steckverbindungen, beschrieben von links nach rechts.

- 1 und 2 sind für die Batterie, wobei 1 Ground ist und 2 12 Volt besitzt.
- 3 und 4 sind für den Kill-Switch, beide sind nicht angeschlossen, sondern werden gebrückt.
- 5 und 6 sind für den Masterswitch, 5 ist dabei In und 6 ist Out.
- 7 und 8 sind für die Versorgung des Relais zuständig. 7 ist dabei V+ und 8 ist V-

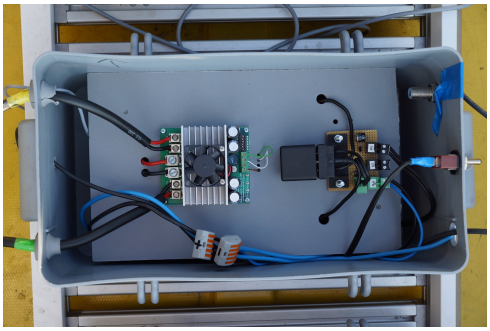


Abbildung C.31: Innenansicht der roten Hardwarebox

C.6.2 Gelbe Box

In der gelben Box C.32 befindet sich eine Ethernet-Verbindung zum Lidar. Außerdem ist die Kamera über ein Mini-USB-Port verbunden.

C.6.3 Grüne Box

In der grünen Box C.33 befindet sich ein weiteres Raspberry welches mit verschiedener Hardware verbunden ist.

- Das XBee ist über Mini-USB angeschlossen und geht auf den HUB
- das ArduinoShield ist über USB-Typ-B angeschlossen

Abbildung C.32: Innenansicht der gelben Hardwarebox

- GPS ist über USB angeschlossen
- 9DoF ist auch über USB angeschlossen

Das Arduino hat das ArduinoShield angeschlossen welches 15 Steckverbindungen besitzt. Wieder betrachtet von links nach rechts.

- 1,2 und 3 sind für den Motor und besitzen die gleiche Belegung, wie bei vorherigen Gruppe.
- Die Verbindungen 4 bis 8 sind für den RC-Receiver, auch hier ist die gleiche Belegung wie vorher.
- 9 und 10 sind für die Batterie und nicht angeklemmt.
- 11 und 12 sind für den Hub und auch nicht angeklemmt
- 13 - 15 sind für die LED und besitzen ebenfalls die Belegung von der vorherigen Projektgruppe

C.6.4 Access-Point

Das Raspberry, sowie die Onshore-Software sind über Wlan mit dm Access-Point verbunden.

C.7 Status-LED

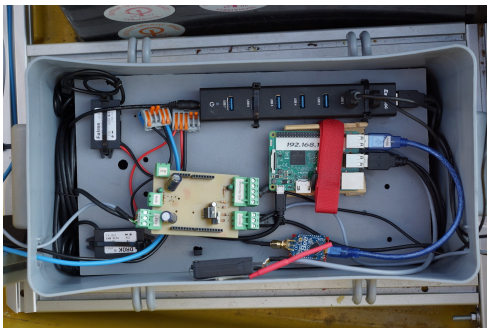


Abbildung C.33: Innenansicht der grünen Hardwarebox

Farbe	Bedeutung
gelb	XBee Verbindung unterbrochen
magenta	Debug Signal
cyan	Wegpunkt erreicht
grün	System bereit

Tabelle C.8: Zustände der Status-LED

Anhang D

Weitere Anhänge

Time [s]	**Speed [m/s]**
	start straight forward
0	0
1	2.9632
2	4.2596
3	5.1856
5	5.3708
6	5.7412
7	5.7412
8	5.7412
9	6
10	5.556
11	5.556
	stop straight forward

Tabelle D.1: Geschwindigkeitswerte der ersten Testfahrt (1)

Time [s]	**Speed [m/s]**
	start straight forward
0	0
1	0.12
2	0.61
3	1.13
5	2.9632
6	4.2596
7	5.1856
8	5.3708
9	5.7412
10	5.7412
11	5.7412
12	5.7
13	5.556
14	5.556
	stop straight forward

Tabelle D.2: Geschwindigkeitswerte der ersten Testfahrt (2)

Time [s]	**Course [°]**
0	181.2
1	180
2	170
3	159.2
5	147.1
6	136.5
7	126.4
8	126.4

Tabelle D.3: Kurswerte der ersten Testfahrt (1)

Abbildung D.1: Meilensteinplanung

Time [s]	Course [°]
0	181.2
1	180
2	178.7
3	170
4	159.2
5	147.1
6	136.5
7	130.5
8	128.8
9	126.4
10	126.4

Tabelle D.4: Kurswerte der ersten Testfahrt (2)

T_G	T_U	K_S	λ	K_R(P-Regler)	K_R(PI-Regler)	K_R(PID-Regler)	T_N(PI-Regler)	T_N(PID-Regler)	T_V(PID-Regler)	
2.04	0.4	55.50	0.09	0.09	0.08	0.11027	1.33	0.80	0.20	forward
5	1	233.6	0.02	0.02	0.02	0.02568	3.33	2.00	0.50	start curved

Tabelle D.5: Einstellregeln nach Ziegler-Nichols-Test (1)

T_G	T_U	K_S	λ	K_R(P-Regler)	K_R(PI-Regler)	K_R(PID-Regler)	T_N(PI-Regler)	T_N(PID-Regler)	T_V(PID-Regler)	
5.00	2	241.10	0.01	0.01	0.01	0.01	6.66	4.00	1.00	reached waypoint 0
6.00	1.00	199.40	0.03	0.03	0.03	0.04	3.33	2.00	0.50	reached waypoint 1
7.00	1.00	273.70	0.03	0.03	0.02	0.03	3.33	2.00	0.50	reached waypoint 0
7.00	2.00	272.40	0.01	0.01	0.01	0.02	6.66	4.00	1.00	reached waypoint 1
7.00	1.00	305.20	0.02	0.02	0.02	0.03	3.33	2.00	0.50	reached waypoint 2
3.00	1.00	342.90	0.01	0.01	0.01	0.01	3.33	2.00	0.50	reached waypoint 3
1.00	1.00	132.90	0.01	0.01	0.01	0.01	3.33	2.00	0.50	start straight forward
5.00	1.00	340.00	0.01	0.01	0.01	0.02	3.33	2.00	0.50	start straight forward

Tabelle D.6: Einstellregeln nach Ziegler-Nichols-Test (2)

1464801959612 start curved								
a_x	a_y	a_z	Δt	t	m_x	Σ_x	m_y	Σ_y
7.17	3.07	218.11	0.000	0.000	0.000	0.000	0.00	0.00
6.14	3.07	217.09	0.064	0.064	0.393	0.393	0.20	0.20
6.14	4.10	217.09	0.289	0.353	1.775	2.168	1.18	1.38
5.12	2.05	217.09	0.058	0.411	0.296	2.465	0.12	1.50
7.17	2.05	216.06	0.289	0.700	2.071	4.537	0.59	2.09
6.14	2.05	218.11	0.063	0.763	0.387	4.924	0.13	2.22
1464801960605 stop curved								

Tabelle D.7: Geschwindigkeitswerte der zweiten Testfahrt (1)

1464801963135 start curved								
a_x	a_y	a_z	Δt	t	m_x	Σ_x	m_y	Σ_y
7.17	4.09	217.09	0.000	0.000	0.000	0.000	0.00	0.00
7.17	2.05	217.09	0.061	0.061	0.437	0.437	0.12	0.12
7.17	2.05	217.09	0.286	0.347	2.050	2.487	0.59	0.71
6.14	3.07	218.11	0.062	0.409	0.380	2.868	0.19	1.90
5.12	3.07	216.06	0.291	0.700	1.489	4.358	0.89	1.80
6.14	3.07	217.09	0.060	0.760	0.368	4.726	0.18	1.98
6.14	4.10	219.09	0.287	0.047	1.763	5.490	1.18	3.15
1464801964335 stop curved								

Tabelle D.8: Geschwindigkeitswerte der zweiten Testfahrt (2)

1464801965710 start curved								
a_x	a_y	a_z	Δt	t	m_x	Σ_x	m_y	Σ_y
7.17	3.07	215.04	0.000	0.000	0.000	0.000	0.00	0.00
6.14	3.07	218.11	0.285	0.285	1.751	1.751	0.88	0.88
6.14	3.07	218.11	0.062	0.347	0.380	2.131	0.19	1.07
7.17	2.05	218.11	0.029	0.637	2.078	4.210	0.59	1.66
7.17	3.07	218.11	0.057	0.694	0.408	4.619	0.18	1.84
6.14	3.07	218.11	0.291	0.985	1.787	6.407	0.89	2.73
4.10	3.07	215.04	0.062	1.047	0.253	6.661	0.19	2.92
5.12	3.07	218.11	0.285	1.332	1.459	0.120	0.88	3.79
6.14	2.05	217.09	0.063	1.395	0.387	8.507	0.13	3.92
6.14	3.07	217.09	0.289	1.684	1.775	10.28	0.89	4.81
7.17	3.07	216.06	0.58	1.742	0.415	10.698	0.18	4.99
5.12	2.05	217.09	0.29	2.032	1.484	12.183	0.59	5.58
6.14	3.07	216, 06	0.058	2.090	0.356	12.539	0.18	5.76
6.14	3.07	217.09	2.379	2.379	1.775	14.315	0.89	6.65
5.12	2.05	217.09	0.61	2.440	0.312	14.627	0.12	6.77
6.14	2.05	217.09	0.286	2.726	1.757	16.385	0.59	7.36
6.14	3.07	216.06	0.062	2.788	0.380	16.765	0.19	7.55
6.14	2.05	216.06	0.291	3.079	1.787	18.553	0.60	8.15
4.10	2.05	219.14	0.62	3.141	0.253	18.807	0.13	8, 27
1464801969010 stop curved								

Tabelle D.9: Geschwindigkeitswerte der zweiten Testfahrt (3)

1464801971546 start curved								
a_x	a_y	a_z	Δt	t	m_x	Σ_x	m_y	Σ_y
5.12	3.07	216.06	0.000	0.000	0.000	0.000	0.00	0.00
6.14	2.05	217.09	0.057	0.057	0.350	0.350	0.12	0.12
6.14	3.07	219.14	0.291	0.348	1.787	2.138	0.89	1.01
7.16	3.07	216.06	0.061	0.409	0.436	2.575	0.19	1.20
5.12	3.07	217.09	0.286	0.695	1.464	4.039	0.88	2.08
5.12	2.05	218.11	0.063	0.758	0.322	4.361	0.13	2.21
5.12	5.12	218.11	0.285	1.043	1.459	5.821	1.46	3.66
5.12	2.05	217.09	0.063	1.106	0.322	6.143	0.13	3.79
7.17	3.07	215.04	0.284	1.390	2.035	8.179	0.87	4.67
5.12	2.05	217.09	0.063	1.453	0.322	8.501	0.13	4.80
5.12	4.10	217.09	0.291	1.744	1.489	9.991	1.19	5.99
5.12	2.05	216.06	0.062	1.806	0.317	10.309	0.13	6.11
5.12	4.10	217.09	0.285000	2.091	1.459	11.768	1.17	7.28
6.14	2.05	218.11	0.063	2.154	0.387	12.155	0.13	7.41
6.14	3.07	219.14	0.285	2.439	1.751	13.906	0.88	8.29
1464801974886 stop curved								

Tabelle D.10: Geschwindigkeitswerte der zweiten Testfahrt (4)

T_G	T_U	K_S	λ	K_R(P-Regler)	K_R(PI-Regler)	K_R(PID-Regler)	T_N(PI-Regler)	T_N(PID-Regler)	T_V(PID-Regler)	
1.50	0.50	0.04	75.00	75.00	67.50	90.00	1.67	1.00	0.25	91410-91430
1.00	1.00	0.04	25.00	25.00	22.50	30.00	3.33	2.00	0.50	91811-91830
1.50	0.50	0.20	15.00	15.00	13.50	18.00	1.67	1.00	0.25	355501-355520
4.00	0.50	0.18	44.44	44.44	40.00	53.33	1.6	71.00	0.25	424040-429068
1.00	0.50	0.29	6.90	6.90	6.21	8.28	1.67	1.00	0.25	486980-487000
2.50	0.50	0.23	21.74	21.74	19.57	26.09	1.67	1.00	0.25	522630-522650

Tabelle D.11: Einstellregeln nach Ziegler-Nichols-Test (3)

Glossar

Abteilung Systemanalyse und -optimierung Die ∼ gehört der Universität Oldenburg an. Die Leitung dieser Abteilung wird von Prof. Dr. Axel Hahn durchgeführt. 1, 16

Ant ∼ ist ein Build-Werkzeug, welches die automatische Erzeugung von ausführbaren Programmen ermöglicht. Die Basis hierfür ist häufig in Java vorliegender Quelltext. 7, 111, 195, 309

Basis Die ∼ ist der Ausgangs- und Endpunkt einer Mission. Im Falle eines Abbruchs bewegt sich das Boot zurück zur Basis. 28, 41, 183, 185

COM-Port Ein ∼ ist eine Schnittstelle zu einem seriellen Port. Dieser kann auch virtuell in Form eines USB zu Seriell Adapters vorliegen. 49, 117

Deadlock Ein ∼ ist ein Zustand in dem sich zwei Prozesse gegenseitig blockieren. Häufig tritt dies durch Sperren von Betriebsmitteln auf. 49

Fuzzy-Regelung Unter dem Begriff der ∼ wird die Regelung mithilfe eines Fuzzy-Reglers verstanden. 56, 95, 96

Fuzzy-Regler Ein ∼ ist eine spezielle Methode der Regelung. Hierbei wird, anders als z.B. bei einem PID-Regler, die Eingangsgröße durch festgelegte Regeln mit der Ausgangsgröße verknüpft. Diese Regeln fragen die Eingangsgröße ab, überprüfen, ob sie in einem festgelegten Wertebereich liegt, und anhand der Regeln wird dann der Wert der Ausgangsgröße festgelegt. 54, 56–58, 96, 155, 156, 163

Gefahrenzone Eine ∼ beschreibt einen Bereich, welcher nicht durch das Boot passiert werden darf. Gründe für Gefahrenzonen können Landflächen, flaches Wasser oder andere sein. 7, 26, 40, 72, 80, 85–87, 112, 113, 118–120, 122, 123, 149, 176, 177, 181, 195, 309, 320, 360, 365, 366

Git ∼ ist ein System zum Verwalten von Dateiversionen. Das System ermöglicht kollaboratives Arbeiten an Software. 12, 14, 195, 291, 294, 300, 302, 305, 316, 328, 329

GPS ∼ steht für „Global Positioning System“ und ist ein auf Satelliten basierendes Navigationssystem zur Positionsbestimmung. 18, 32, 44, 48–52, 66–68, 80, 87, 92, 94, 99, 116–118, 149, 153, 158, 162, 193, 197, 206, 207, 212, 219, 230, 236–239, 280, 282, 320, 335, 347, 364, 369, 371, 378, 382, 387

HSV-Farbraum ∼ kennzeichnet den Farbraum der durch den Farbwert (H), die Sättigung (S) und die Helligkeit (V) beschrieben wird. 64, 65, 142

ICBM ∼ steht für „Institut für Chemie und Biologie des Meeres“ und ist ein in Oldenburg ansässiges Institut. 1, 13, 16, 22, 23, 289, 295

Image-Datei Eine ∼ ist eine exakte Abbildung eines Datenträgers. 49

IMU ∼ steht für „inertiale Messeinheit“ und ist eine System zur Erfassung der räumlichen Ausrichtung eines Objektes. Eine ∼ besteht in der Regel aus mehreren Sensoren wie einem 3-Achsen-Magnetfeld-Sensor, einem Beschleunigungssensor und einem Gyroskop. 48, 51, 87, 99, 197, 219, 334, 369

Kollisionserkennung Mit ∼ ist das Erkennen von dynamisch erscheinenden Hindernisse, die zu einer möglichen Kollision führen können, gemeint. 1–3, 6–11, 18, 20, 40, 42–45, 54, 58, 64, 67–69, 77, 78, 81, 82, 84, 86–92, 97–99, 107, 110–112, 117, 120, 122–124, 127, 129–134, 142–144, 146, 172, 174–176, 178, 179, 183, 185, 187, 190, 195–199, 201–203, 263, 264, 271, 291, 300–304, 309, 313, 315, 317, 318, 320–324, 326–329, 331–333, 337, 338, 340, 341, 344, 349, 353–358, 363, 364

Kollisionsverhütung Mit ∼ ist der Vorgang gemeint, der eine mögliche Kollision mit Hindernissen abwenden soll. 1, 2, 6, 9–11, 32, 35, 40, 44, 46, 87–93, 97, 123, 131, 148, 149, 195, 197–199, 201, 202, 221, 262, 272, 340, 342

Maven ∼ ist ein Build-Management-Tool, dass eine standardisierte Erstellung von Java-Programmen ermöglicht. Außerdem ist das Versionieren von Software möglich. 7, 45, 111, 195, 212, 309, 314, 317

neuronales Netz Ein ∼ bezeichnet eine miteinander verknüpfte Anzahl von Neuronen, die einen funktionalen Zusammenhang bilden. 95, 196, 313

Onboard-Software Mit ∼ ist der Softwareteil gemeint, der auf den Raspberry Pis des Wasserfahrzeugs ausgeführt wird. Dieser übernimmt sowohl die Kommunikation zur Onshore-Software als auch die Steuerung der Motoren. 3, 8, 9, 45, 48–50, 99, 112, 123, 128, 133, 140, 205, 207, 208, 210–212, 216, 259, 342, 371

Onshore-Software Die ∼ wird primär zur Missionsplanung und -überwachung eingesetzt. Diese verfügt über eine Benutzeroberfläche und wird auf einem Rechner außerhalb des Bootes ausgeführt. 18, 19, 25–28, 30, 32, 33, 45, 48–50, 68, 87, 97–99, 111–117, 127, 129, 131, 150, 151, 179, 202, 203, 205, 206, 208–212, 259, 312, 314, 320, 321, 337, 338, 341, 354, 359–361, 363–365, 379, 387

PID-Regler Ein ∼ ist eine Art der Prozesskontrolle. Hierbei wird eine Eingangsgröße mit einer mathematischen Funktion belegt um das Verhalten der Ausgangsgröße des Prozesses zu regeln. 55, 56, 95, 96, 151, 155, 157, 166, 281, 313, 392, 394

Regelung Bei der ∼ wird ein Prozess derart überwacht, dass die Ausgangsgröße des Prozesses mit der Eingangsgröße verglichen wird. Hiermit wird die Annäherung der Ausgangsgröße an einen Sollwert erreicht. 1–3, 6, 8, 10, 11, 17–20, 32, 33, 43, 44, 46, 47, 54–56, 86, 87, 92, 94, 96, 97, 111, 150–155, 157, 158, 161, 169, 171, 178, 180, 195–199, 201, 208, 259, 271, 272, 279, 281, 283, 284, 287, 300–302, 309, 310, 313, 317, 320, 323, 328, 330, 331, 333, 337, 339, 341, 344–346, 349, 353, 355, 356, 360

RGB-Farbraum ∼ kennzeichnet den Farbraum der durch das additive Mischen der drei Grundfarben Rot, Grün und Blau entsteht. 64, 65, 142

Scrum ∼ ist ein Vorgehensmodell für die agile Softwareentwicklung. 4, 5, 14, 241, 245–248, 250, 293, 294, 307, 345

Torque-Vectoring Das ∼ ist eine Möglichkeit eine Richtungsänderung eines Fahrzeuges zu beschreiben, indem zwei Antriebseinheiten mit einer unterschiedlichen Drehmomentenvorgabe belegt werden. Hierdurch kann ein Richtungsvektor beschrieben werden. 94

valide Mission Eine Bootsmission ist dann valide, wenn die zu befahrene Route keine Gefahrenzonen schneidet. 27, 28

XBee ∼ ist ein auf Funk basierenden Datenübertragungsstandard. Die Daten werden seriell übertragen. 41, 49, 50, 52, 82, 87, 97–99, 111, 117, 127, 130, 131, 179, 185, 197, 204, 205, 210–212, 214, 216, 220, 320, 331, 362, 381

Literaturverzeichnis

[1] *Arduino MEGA - HMC 5883L Compass Display.* https://www.youtube.com/watch?v=gUS9Szvtzb8, besucht: 13. Dezember 2015.

[2] *Datasheet Nephelometric Turbidity.* http://www.ponsel-web.com/cbx/_ftp/datasheetdigisensntuuk.pdf, besucht: 8. November 2015.

[3] *FMCW Frequenzversatz.* http://www.nl.advsolned.com/images/fmcw_plot.jpg, besucht: 13. Dezember 2015.

[4] *IALA Maritime Buoyage System.* http://www.instructortoolkit.co.uk/shorebased-training/buoyage, besucht: 13. Dezember 2015.

[5] *Internationale Regeln von 1972 zur Verhütung von Zusammenstößen auf See.* http://www.gesetze-im-internet.de/seestro_1972/BJNR008160977.html, besucht: 4. Januar 2016.

[6] *NMEA-0183 Daten.* http://www.kowoma.de/gps/zusatzerklaerungen/NMEA.htm, besucht: 13. Dezember 2015.

[7] *OpenSeaMap. Freie Seekarten.* http://www.openseamap.org/index.php?id=openseamap&no_cache=1, besucht: 13. Dezember 2015.

[8] *OpenSeaMap. Freie Wetterkarten.* http://www.openseamap.org/index.php?id=wetter, besucht: 13. Dezember 2015.

[9] *Prinzipskizze Radartechnik.* http://www.seefunkschule.at/radar1.gif, besucht: 13. Dezember 2015.

[10] *RADARduino.* http://store.reactancelabs.com/product-p/radarduinov1.htm, besucht: 13. Dezember 2015.

[11] *Strömungsvorausberechnung Deutsche Bucht.* www.bsh.de, besucht: 8. November 2015.

[12] *Was ist Kanban?* http://www.it-agile.de/wissen/methoden/kanban, besucht: 13. Dezember 2015.

[13] *Agiles Projektmanagement*, 2015. http://agiles-projektmanagement.org, besucht: 13. Dezember 2015.

[14] *Duden*, 2015. http://www.duden.de/rechtschreibung/Projektmanagement, besucht: 13. Dezember 2015.

[15] *Jira-Software*, 2015. https://de.atlassian.com/software/jira, besucht: 13. Dezember 2015.

[16] *Jira-Software*, 2015. https://de.atlassian.com/software/jira, besucht: 13. Dezember 2015.

[17] *Projektmanagement*, 2015. https://www.blikk.it/blikk/angebote, besucht: 13. Dezember 2015.

[18] *Projektmanagement*, 2015. http://www.mediencommunity.de/system/files/wbts/projektmanagement, besucht: 13. Dezember 2015.

[19] *Projektmanagement: Definitionen, Einführungen und Vorlagen*, 2015. http://projektmanagement-definitionen.de/glossar/methode, besucht: 13. Dezember 2015.

[20] *Redmine*, 2015. http://www.redmine.org/projects/redmine, besucht: 13. Dezember 2015.

[21] *Redmine: Projektverwaltung und Ticketsystem*, 2015. http://www.programmieren-optimieren.de/tools/redmine-projektverwaltung-und-ticketsystem, besucht: 13. Dezember 2015.

[22] *Scrum*, 2015. https://www.3m5.de/scrum, besucht: 13. Dezember 2015.

[23] *Scrum Roles*, 2015. http://www.agile42.com/en/agile-info-center/scrum-roles, besucht: 13. Dezember 2015.

[24] *Spiralmodell*, 2015. http://members.it4education.at/fubbcontent/lektionen/letzte_lieferungen/Phasenmodelle/data/17.html, besucht: 13. Dezember 2015.

[25] *Waterfall-model*, 2015. http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm, besucht: 13. Dezember 2015.

[26] *Waterfall-Model*, 2015. http://www.isicore.de/Waterfall-Model, besucht: 13. Dezember 2015.

[27] *Waterfall/V-Model*, 2015. http://www.kalpa.it/index.php/en/methodologies/waterfallv-model-eng, besucht: 13. Dezember 2015.

[28] *What is V-model- advantages, disadvantages and when to use it?*, 2015. `http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it`, besucht: 13. Dezember 2015.

[29] *What is Waterfall model-advantages, disadvantages and when to use it?*, 2015. `http://istqbexamcertification.com/?s=waterfall`, besucht: 13. Dezember 2015.

[30] ADAC Marinaführer: *Oldenburger Yacht-Club e.V.* `http://www.marinafuehrer.adac.de/haefen/oldenburger-yacht-club-e-v-2/`, besucht: 4. Januar 2016.

[31] Alotaibi, Faisal, Tim Baalmann, Matthias Esser, Conrad Fifelski, Thomas Hellkamp, Christelle Kamga, Paul Kröger, Simon Ortmann, Christof Schlaak, Weert Stamm und Carole Tchuenkam: *Abschlussbericht der Projektgruppe Marine Observation Platform for Surfaces III.* Universität Oldenburg, 2015.

[32] Alotaibi, Faisal, Tim Baalmann, Matthias Esser, Conrad Fifelski, Thomas Hellkamp, Christelle Kamga, Paul Kröger, Simon Ortmann, Christof Schlaak, Weert Stamm und Carole Tchuenkam: *Marine Observation Plattform for Surfaces. Final Presentation of PG MOPS III.* Abschlussvortrag, Universität Oldenburg, 2015.

[33] Amazon. `https://www.amazon.de/Logitech-C920-Webcam-1080p-schwarz/dp/B006A2Q81M/ref=sr_1_1?ie=UTF8&qid=1474448817&sr=8-1&keywords=Logitech+HD+Pro+C920`, besucht: 21. September 2016.

[34] Aström, Karl Johan: *PID Regler.* 2002.

[35] Bechtloff, Jürgen: *Regelungstechnik.* Vogel Studienmodule, 1. Auflage, 2012.

[36] Borgmann, Björn, Justus Sebastian Bücker, Daniel Fay, Jan Friedrichs, Konstantin Franzuski, Kamran Ghanaat, Julian Janke, Niels Klissing, Matthias Larisch, Jan Manemann und Philip Weber: *Abschlussbericht der Projektgruppe Marine Observation Platform for Surfaces I.* Universität Oldenburg, 2013.

[37] Brox, Detlef: *Kardinal- und Lateralsystem*, 2003. `http://www.hoch-am-wind.de/05-Sonstiges/092-sbf/pdf/002-sbf-02.pdf`, besucht: 13. Dezember 2015.

[38] Budde, Verena: *Kanban – extrem lean und hoch effektiv!*, 2015. `http://de.dice.com/nachrichten/kanban-extrem-lean-und-hoch-effektiv`, besucht: 13. Dezember 2015.

[39] Bundes, Wasser und Schiffsverwaltung des: *Strömungsvorausberechnung*. `http://www.wsa-bremerhaven.de`, besucht: 8. November 2015, Gewässerkunde und Strömungen.

[40] Bundesamt für Seeschifffahrt und Hydrographie: *Gezeitenvorausberechnung*. `http://www.bsh.de/cgi-bin/gezeiten/was_tab.pl?ort=DE__748P&zone=Gesetzliche_20Zeit_niveau=KN`, besucht: 4. Januar 2016.

[41] Bundesministerium der Jutiz und für Verbraucherschutz: *Internationale Regeln von 1972 zur Verhütung von Zusammenstößen auf See*. `http://www.gesetze-im-internet.de/seestro_1972`, besucht: 15. November 2015.

[42] Bundesministerium der Jutiz und für Verbraucherschutz: *Seeschifffahrtsstraßen-Ordnung*. `http://www.gesetze-im-internet.de/seeschstro_1971`, besucht: 15. November 2015.

[43] Burger, W. und M.J. Burge: *Digitale Bildverarbeitung: Eine algorithmische Einführung mit Java*. Springer Berlin Heidelberg, 2006, ISBN 9783540309413.

[44] Canny, J: *A Computational Approach to Edge Detection*. IEEE Trans. Pattern Anal. Mach. Intell., 1986.

[45] Caspary, Wilhelm: *Vom statischen zu kinematischen Messmethoden*. Fachbeitrag, Mai 2002.

[46] Charvat, Gregory L.: *Try Radar for Your Next Project*. `http://hackaday.com/2014/02/24/guest-post-try-radar-for-your-next-project`, besucht: 13. Dezember 2015.

[47] Diemke, Johannes: *Pfadplanung mit harmonischen Potentialfeldern*. Bachelorarbeit, Universität Oldenburg, 2012.

[48] Dorf, Richard C. und Robert H. Bishof: *Moderne Regelungssysteme*. Pearson Studium, 2007, ISBN 978-3-8632-6623-3.

[49] Dörner, Wieland: *Vorfahrtsregeln, Ausweichregeln, Verhalten auf dem Wasser*. `http://www.xn-yachtclub-moenchengladbach-voc.de/wetterkunde/wegerecht-verhalten-auf-see`, besucht: 15. November 2015.

[50] Ehmen, Günter: *Praktikum Fahrzeuginformatik*. Vorlesung, Universität Oldenburg, Wintersemester 2015/216.

[51] Eisenhauer, Norbert: *Schiffsstabilität*. Technischer Bericht, Hochschule Karlsruhe Technik und Wirtschaft.

[52] Electronics UK Limited tyco: *RS485 & Modbus Protocol Guide*. Technischer Bericht, Energy Division.

[53] Evers, Hauke, Nils Giza, Uwe Wilko Grünefeld, Dennis Koers, Markus Alexander Lehmann, Philipp Mählmeyer, David Reiher, Philipp Rudolph, Liliane Anick Tchoua Tsafack, Bastian Veltin, Daniel Weinberg und Sven Wiegand: *Abschlussdokumentation im Rahmen des Projekts Marine Observation Platform for Surfaces 2*. Universität Oldenburg, 2014.

[54] Flaßkamp, Maik: *Wie funktioniert GPS? Einfach erklärt.* `http://praxistipps.chip.de/wie-funktioniert-gps-einfach-erklaert_41414`, besucht: 13. Dezember 2015.

[55] Forschungs- und Entwicklungszentrum Fachhochschule Kiel GmbH: *FINO - Forschungsplattformen in Nord- und Ostsee*. `http://www.fino-offshore.de/de/`, besucht: 7. Dezember 2015.

[56] Forschungs- und Entwicklungszentrum Fachhochschule Kiel GmbH: *Projekt*. `http://www.fino1.de/fino1/projekt`, besucht: 7. Dezember 2015.

[57] Greguhn, Dennis: *RS-485 Modbus*. `http://www.greguhn.de/rs485-modbus-kommunikation/`, besucht: 11. November 2015.

[58] Group, SDI 12 Support: *SDI-12 A Serial-Digital Interface Standard for Microprocessor-Based Sensors*, January 2016. Version 1.3.

[59] Haake, Hauke: *Entwicklung eines automatischen Sensormoduls auf Grundlage des MOPS2-Systems*. Bachelorarbeit, Jade Hochschule, 2015.

[60] Hasnain, Kazim: *Antrieb ohne Öl: Reeder planen das Hochsee-Schiff der Zukunft. Brennstoffschiff: Am Ende bleibt nur Wasserdampf*, 2008. `http://www.spiegel.de/wirtschaft/antrieb-ohne-oel-reeder-planen-das-hochsee-schiff-der-zukunft-a-569312-3.html`, besucht: 7. Dezember 2015.

[61] Hasnain, Kazim: *Antrieb ohne Öl: Reeder planen das Hochsee-Schiff der Zukunft. Wellenkraft: Energie aus dem Meer*, 2008. `http://www.spiegel.de/wirtschaft/antrieb-ohne-oel-reeder-planen-das-hochsee-schiff-der-zukunfta-569312-11.html`, besucht: 7. Dezember 2015.

[62] Hauser, Endress: *Operating Instructions Turbimax CUS52D*. `https://portal.endress.com`, besucht: 8. November 2015, BA01275CEN0113.pdf.

[63] ICBM - Carl von Ossietzky Universität Oldenburg: *Messtation Spiekeroog*, 2014. http://www.icbm.de/messstation/, besucht: 7. Dezember 2015.

[64] Inc., PBworks: *RS-485 Signale*. http://wilbo666.pbworks.com/w/page/50747257/RS422-RS485, besucht: 13. November 2015.

[65] INNOC: *ASV Roboat*. http://www.roboat.at/de/technologie/technologie/, besucht: 7. Dezember 2015.

[66] INNOC: *Roboat*. http://www.roboat.at/de/home/, besucht: 7. Dezember 2015.

[67] Intland GmbH: *CodeBeamer*, 2015. http://www.aktiv-verzeichnis.de/anlagen/631/codebeamer.pdf, besucht: 13. Dezember 2015.

[68] Intland GmbH: *Wiki Plugins*, 2015. http://www.intland.com/cb/wiki/200361, besucht: 13. Dezember 2015.

[69] Kohlhoff, Christopher M.: *Boost Asio*, 2015. http://www.boost.org/doc/libs/1_61_0/doc/html/boost_asio.html, besucht: 2016-09-05.

[70] Kongsberg Maritime AS: *HUGIN Autonome Unterwasserfahrzeuge*. http://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/6A524BCD3B1DFEE4C1257911002C6809?OpenDocument, besucht: 7. Dezember 2015.

[71] Kongsberg Maritime AS: *REMUS 100 - Autonomes Unterwasserfahrzeug*. http://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/BB2E902E56DBD340C12579110030CA7B?OpenDocument, besucht: 7. Dezember 2015.

[72] Koord, Y. und V. Krauter: *Überblick Vorgehensmodelle im Projektmanagement*, 2015. http://winfwiki.wi-fom.de/index.php/%C3%9Cberblick_Vorgehensmodelle_im_Projektmanagement, besucht: 13. Dezember 2015.

[73] Kunst, Michael, 2013. http://segelreporter.com/panorama/alternativer-antrieb-e-ship-1-mit-flettner-rotoren-2014-wieder-unterwegs/, besucht: 7. Dezember 2015.

[74] Latombe, Jean Claude: *Robot Motion Planning*. Band 124. Springer, 1991, ISBN 978-0-7923-9206-4.

[75] Lewis, F. L. und Shuzhi Sam Ge: *Neural Networks in Feedback Control Systems*. http://www.pdx.edu/sites/www.pdx.edu.sysc/files/media_assets/SySc576_FrankLewisNNsControl.pdf, besucht: 2005.

[76] Likhachev, Maxim, Geoff Gordon und Sebastian Thrun: *ARA*: Anytime A* with Provable Bounds on Sub-Optimality*. Band 16. MIT Press, 2003.

[77] MRPT: *MRPT - Empowering C++ development in robotics*, 2016. http://www.mrpt.org/, besucht: 2016-09-08.

[78] Müller, Ernst: *Eisenschiffbau*. Salzwasser Verlag, 2011, ISBN 978-3864441776.

[79] Neumeier, Franz: *Wie Stabilisatoren auf Schiffen den Seegang überlisten*, 2015. http://www.cruisetricks.de/kreuzfahrtschiff-stabilisatoren, besucht: 13. Dezember 2015.

[80] Odegard, Vidar: *Nonlinear Identification of Ship Autopilot Models*. Dissertation, Norwegian University of Science and Technologie, 2009.

[81] OpenCV: *Sobel Derivatives*, 2014. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html, besucht: 2016-05-14.

[82] OpenCV: *Camera calibration With OpenCV*, 2015. http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html, besucht: 2016-05-14.

[83] OpenCV: *Extract horizontal and vertical lines by using morphological operations*, 2015. http://docs.opencv.org/3.1.0/d1/dee/tutorial_moprh_lines_detection.html, besucht: 2016-05-14.

[84] OpenCV: *Image Inpainting*, 2015. http://docs.opencv.org/3.1.0/df/d3d/tutorial_py_inpainting.html, besucht: 2016-05-14.

[85] OpenCV: *Miscellaneous Image Transformations*, 2015. http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html, besucht: 2016-05-14.

[86] OpenCV: *Basic Thresholding Operations*, 2016. http://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html, besucht: 2016-05-14.

[87] OpenCV: *Canny Edge Detector*, 2016. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html, besucht: 2016-09-05.

[88] OpenCV: *Discrete Fourier Transform*, 2016. http://docs.opencv.org/2.4/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html, besucht: 2016-09-02.

[89] OpenCV: *Eroding and Dilating*, 2016. http://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html, besucht: 2016-05-14.

[90] OpenCV: *Histogram Calculation*, 2016. http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_calculation/histogram_calculation.html, besucht: 2016-05-14.

[91] OpenCV: *Hough Line Transform*, 2016. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html, besucht: 2016-05-14.

[92] OpenCV: *Structural Analysis and Shape Descriptors*, 2016. http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html, besucht: 2016-05-14.

[93] Ponsel: *Datasheet C4E*. www.ponsel-web.com, besucht: 8. November 2015.

[94] Radulescu und Uerpmann GbR: *Positionslichter*. http://www.segeln-wissen.de/segelwissen/seezeichen/positionslichter.html, besucht: 4. Januar 2016.

[95] Reuter, Rainer: *Meeresdaten rund um die Uhr: Die Station im Watt*. http://www.presse.uni-oldenburg.de/einblicke/41/2-reuter.pdf, besucht: 7. Dezember 2015.

[96] Robots, Let's Make: *Occupancy Grid Programming Help*. http://letsmakerobots.com/node/26083, besucht: 2016-08-25.

[97] ROS.org: *LMS1xx*, 2015. http://wiki.ros.org/LMS1xx, besucht: 2016-09-05.

[98] Schifffahrt, Generaldirektion Wasserstraßen und: *Ausbaugrundlagen*. http://www.ast-mitte.gdws.wsv.de/ausbau/ausbaugrundlagen/index.html, besucht: 4. Januar 2016.

[99] Schnabel, Gisela: *Die Farbmodelle HSV und HLS - Widersprüche in Theorie und Praxis*. Multimedia-Anwendungen. Humboldt-Universität zu Berlin, 1999.

[100] Schwan, Ben: *Mannschaft von Bord*, 2014. http://www.heise.de/tr/artikel/Mannschaft-von-Bord-2138501.html, besucht: 7. Dezember 2015.

[101] Seifert, Georg: *Solar-Tipps*. http://www.shipshop.de/produkte/energie-an-bord/solar/solar-tipps.html, besucht: 7. Dezember 2015.

[102] Seifert, Georg: *Windgeneratoren.* http://www.shipshop.de/produkte/energie-an-bord/windgeneratoren.html, besucht: 7. Dezember 2015.

[103] Sheller, Alain, 2013. http://www.allboatsavenue.com/suntory-mermaid-ii-de-kenichi-horie-le-premier-bateau-a-propulsion-a-vagues/, besucht: 7. Dezember 2015.

[104] SICK, Sensor Intelligence. www.sick.com/de/de/p/p1418401, besucht: 21. September 2016.

[105] Stewart, Robert: *Introduction to Physical Oceanography.* http://oceanworld.tamu.edu/resources/ocng_textbook/PDF_files/book_pdf_files.html, besucht: 13. Dezember 2015.

[106] Struckhof, Detlef: *Minentaucher erproben neuartige Tauchdrohne*, 2008. http://www.presseportal.de/pm/67428/1318666, besucht: 7. Dezember 2015.

[107] Students Sail Autonomously: *Das Projekt.* http://www.ssa.ethz.ch/?q=node/13, besucht: 7. Dezember 2015.

[108] Süße, Herbert und Erik Rodner: *Bildverarbeitung und Objekterkennung: Computer Vision in Industrie und Medizin.* Springer Fachmedien Wiesbaden, 2014, ISBN 9783834826060.

[109] Tanaka, Kazuo und Hua O. Wang: *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach.* Wiley, 2001, ISBN 978-0-471-32324-1.

[110] Thol, Andreas: *Slippen: die Dos and Don'ts des Zuwasserlassens.* http://www.bootstechnik.de/2012/05/14/slippen/, besucht: 4. Januar 2016.

[111] Töpel, Daniel: *Das kleine Lexikon der Regelungstechnik.* http://dt77.gmxhome.de/Daten/Regelungstechnik/files/einst/index.htm, besucht: 11. November 2015.

[112] Ulrich, Iwan und Nourbakhsh Illah: *Appearance-Based Obstacle Detection with Monocular Color Vision.* 2000.

[113] Unbehauen, Heinz: *Regelungstechnik 1.* VIEWEG + TEUBNER, 2008.

[114] Viola, Paul: *Rapid Object Detection using a Boosted Cascade of Simple Features*, 2001. https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf, besucht: 24. September 2016.

[115] Walther, Hansjoachim und Günter Nägler: *Graphen-Algorithmen—Programme.* 1987.

[116] Wasser- und Schifffahrtsverwaltung des Bundes: *Elektronischer Wasserstraßen-Informationsservice*. https://www.elwis.de/Schifffahrtsrecht/Seeschifffahrtsrecht/KVR/Teil-B/Abschnitt-I/Regel10, besucht: 13. Dezember 2015.

[117] Wasser- und Schifffahrtsverwaltung des Bundes: *Schifffahrtsrecht, Schiffszulassung, Patente*. www.elwis.de/Schifffahrtsrecht, besucht: 15. November 2015.

[118] Wikipedia: *Ausgleichsrechnung*. https://de.wikipedia.org/wiki/Ausgleichungsrechnung, besucht: 28. Oktober 2015.

[119] Wikipedia: *Automatisches Identifikationssystem*. https://upload.wikimedia.org/wikipedia/commons/6/63/Ais_dcu_bridge.jpg, besucht: 13. Dezember 2015.

[120] Wikipedia: *Doppler-Effekt*. https://de.wikipedia.org/wiki/Doppler-Effekt, besucht: 13. Dezember 2015.

[121] Wikipedia: *Hunte*. https://de.wikipedia.org/wiki/Hunte, besucht: 4. Januar 2016.

[122] Wikipedia: *Kompass*. https://de.wikipedia.org/wiki/Kompass#/media/File:Compass_in_a_wooden_frame.png, besucht: 13. Dezember 2015.

[123] Wikipedia: *Küstenkanal*. https://de.wikipedia.org/wiki/K%C3%BCstenkanal, besucht: 4. Januar 2016.

[124] Wikipedia: *Küstenmotorschiff*. https://de.wikipedia.org/wiki/K%C3%BCstenmotorschiff, besucht: 4. Januar 2016.

[125] Wikipedia: *Radar-Bildschirm*. https://upload.wikimedia.org/wikipedia/commons/0/0d/Radar_screen.jpg, besucht: 13. Dezember 2015.

[126] Wikipedia: *Schiffsstabilisator*. https://de.wikipedia.org/wiki/Schiffsstabilisator, besucht: 4. Januar 2016.

[127] Wikipedia: *Schlingerkiel*. https://de.wikipedia.org/wiki/Schlingerkiel, besucht: 4. Januar 2016.

[128] Wikipedia: *Schlingertank*. https://de.wikipedia.org/wiki/Schlingertank, besucht: 4. Januar 2016.

[129] Wikipedia: *Seeschifffahrtsstraßen-Ordnung*. https://de.wikipedia.org/wiki/Seeschifffahrtsstrassen-Ordnung, besucht: 15. November 2015.

[130] Wikipedia: *Verkehrstrennungsgebiet*. `https://de.wikipedia.org/wiki/Verkehrstrennungsgebiet`, besucht: 15. November 2015.

[131] Wikipedia: *Kennzeichnung von Gefahrenstellen mit kardinalen Seezeichen*, 2009. `https://de.wikipedia.org/wiki/Kardinalsystem#/media/File:Cardinal_mark_diagram.svg`, besucht: 13. Dezember 2015.

[132] Wikipedia: *V-Modell*, 2015. `https://de.wikipedia.org/wiki/V-Modell`, besucht: 13. Dezember 2015.

Selbstständigkeitserklärung

Hiermit versichern wir, dass wir diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Außerdem versichern wir, dass wir die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt haben.

Oldenburg, den *28. September 2016*

Michael Beering	Konstantin Gebel
Eike Hagena	Maximilian Hipp
Björn Koopmann	Henning Lawatsch
Zahra Paya	Dennis Pilny

Peter Tank