Höhere Technische Bundeslehranstalt

Klagenfurt, Mössingerstrasse

Abteilung Elektonik und Technische Informatik

DIPLOMARBEIT



**SmartRobot**

eingereicht von Jakob Ecker

Dominik Simon

Projektbetreuer Dipl.-Ing. Dr. Klampferer Wolfgang

Diese Diplomarbeit entspricht den Standards gemäß Leitfaden für Ingenieurprojekte des BM|BF.

Klagenfurt, am Sonntag, 15. Februar 2015

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Klagenfurt, am 15.09.2014 Verfasser:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jakob Ecker

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dominik Simon

# Kurzbeschreibung des Projektes

## Kurzfassung

## Abstract

Inhalt

[1 Kurzbeschreibung des Projektes 1](#_Toc412471470)

[1.1 Kurzfassung 1](#_Toc412471471)

[1.2 Abstract 1](#_Toc412471472)

[2 Einleitung 4](#_Toc412471473)

[3 Grundlagen und Methoden 5](#_Toc412471474)

[3.1 Stand der Technik und mögliche Lösungsansätze 5](#_Toc412471475)

[3.2 Gewählter Ansatz 6](#_Toc412471476)

[3.3 Realisierung 6](#_Toc412471477)

[4 Individuelle Zielsetzung der Teammitglieder 7](#_Toc412471478)

[4.1 Individuelle Zielsetzung des Teammitglieds „Jakob Ecker“ 7](#_Toc412471479)

[4.1.1 Aufgabenstellung 7](#_Toc412471480)

[4.1.2 Grundlagen und Methoden 8](#_Toc412471481)

[4.1.3 Realisierung 11](#_Toc412471482)

[4.1.4 Ergebnisse 13](#_Toc412471483)

[4.2 Individuelle Zielsetzung des Teammitglieds „Dominik Simon“ 14](#_Toc412471484)

[4.2.1 Aufgabenstellung 14](#_Toc412471485)

[4.2.2 Grundlagen und Methoden 14](#_Toc412471486)

[4.2.3 Realisierung 15](#_Toc412471487)

[4.2.4 Ergebnisse 15](#_Toc412471488)

[5 Zusammenfassung 16](#_Toc412471489)

[6 Abkürzungsverzeichnis: 16](#_Toc412471490)

[7 Literatur und/oder Quellenverzeichnis 16](#_Toc412471491)

[8 Anhang (Dokumente und Protokolle) 18](#_Toc412471492)

[8.1 Projektdokumentation 18](#_Toc412471493)

[8.1.1 Pflichtenheft 18](#_Toc412471494)

[8.1.2 Organisationsstruktur 19](#_Toc412471495)

[8.1.3 Projektstruktur 20](#_Toc412471496)

[8.1.4 Terminplan 23](#_Toc412471497)

[8.1.5 Kalkulation 24](#_Toc412471498)

[8.2 Besprechungsprotokolle 26](#_Toc412471499)

[8.3 Arbeitszeitnachweise 27](#_Toc412471500)

[8.4 Technische Dokumentation 29](#_Toc412471501)

[8.5 Arbeitspakete 29](#_Toc412471502)

[8.6 Businessplan (Optional) 30](#_Toc412471503)

[9 Abbildungsverzeichnis 31](#_Toc412471504)

# Einleitung

Das Leben wird immer mehr durch den Einsatz verschiedenster Roboter erleichtert. Beispielsweise sind stationäre Industrieroboter in Fertigungsanlagen schon lange der Standard und auch im alltäglichen Leben kommen Roboter verstärkt zum Einsatz. Haushaltshilfen wie automatische Staubsauger oder Rasenmäher verbreiten sich. Solche Roboter müssen in der Lage sein sich auf selbstständig berechneten Routen zu bewegen. Aus diesem Grund beschäftigt sich die vorliegende Arbeit mit der Weiterentwicklung von autonomer Routenplanung für Roboter aus allen möglichen Einsatzgebieten.

Auffällig ist, dass derzeit am Markt befindliche Roboter ihre Routen meist zufällig wählen und diese nur wenn sie auf ein Hindernis treffen ändern. Dieses Operationsschema ist sehr ineffizient und völlig ungeeignet wenn Hindernisse, zum Beispiel ein Abgrund, auftreten, die der Roboter nicht erkennen kann. Des Weiteren ist es mit dieser Art der Routenberechnung nur möglich Flächen abzudecken, einen einzelnen Punkt anzusteuern ist nahezu unmöglich.

Um die Bewältigung solcher Aufgaben und eine effizientere Routenplanung zu ermöglichen ist es essentiell das der Roboter seine eigene Position bestimmen kann. Wenn er dazu in der Lage ist können auch neue Einsatzgebiete wie selbstfahrende Transporter realisiert werden.

Ziel dieser Arbeit ist es daher, einen Roboter zu entwickeln der selbstständig die optimale Route planen und dieser folgen kann. Hierfür müssen, aufbauend auf am Markt befindlichen Geräten, folgende Herausforderungen bewältigt werden:

* Implementieren eines Systems zu Positionsbestimmung
* Erstellen eines Systems zur Ansteuerung der Motoren
* Entwickeln der Roboter-Hardware

Zur Lösung dieser Herausforderungen werden im Zuge dieser Arbeit bestehende Positionierungssysteme wie GPS und andere Ansätze für lokale Systeme analysiert und nach folgenden Gesichtspunkten bewertet:

* Genauigkeit

Für die Umsetzung einer leistungsfähigen, intelligenten Routenplanung darf die Abweichung nicht größer als wenige Zentimeter (>10cm) sein.

* Umsetzbarkeit:

Wie einfach ist es den Ansatz zu implementieren? Worauf muss man achten?

Des Weiteren wird getestet ob eine Kombination mehrerer Ansätze eine Verbesserung nach sich zieht. Geplant ist das die Erzielten Ergebnisse als Grundlage für künftige Entwicklungen in diesem Gebiet dienen.

# Grundlagen und Methoden

## Stand der Technik und mögliche Lösungsansätze

Verfügbare Technologien zur Positionsbestimmung können in verschiedene Kategorien eingeteilt werden. Es gibt globale Systeme wie GPS oder GLONASS, die auf ähnlichen Prinzipien aufbauen und weltweit genutzt werden können. Diese globalen Systeme haben allerdings den Nachteil, dass sie zum Beispiel nicht innerhalb von Gebäuden genutzt werden können. Deshalb existieren auch lokale Systeme, die hauptsächlich für die Verwendung in Innenräumen (zum Beispiel eine Navigation im Einkaufszentrum) entwickelt wurden. Da die erforderliche Genauigkeit von den globalen Systemen ohne teure spezial-Hardware nicht erreichbar ist (GPS: 5-20m, mit Korrektursignal: 1-3m [NATH 15]) werden diese im weiteren Verlauf der Arbeit nicht mehr Berücksichtigt.

Lokale Systeme sind mit zwei Ansätzen, die jeweils wieder mit mehreren unterschiedlichen Technologien realisiert werden können, verfügbar. Einer dieser Ansätze ist es der Absolute, zu dem auch die globalen Systeme gehören, der die Position in einem bekannten Koordinatensystem berechnet. Der andere ist der relative Ansatz der die Änderung der Position bestimmt, die damit relativ zur Startposition ist.

Zu dem Ersteren gehören die Triangulation und die verbreitetere Trilateration während zum Zweiten die Odometrie und die inertiale Positionierung zählen. Genaueres zu diesen Ansätzen ist in Kapitel 4.2.2.1 nachzulesen.

Jede dieser Möglichkeiten hat große Nachteile, deshalb werden für viele Anwendungen mehrere dieser Ansätze kombiniert um ein geeignetes System zu erstellen. Meistens wird ein absoluter mit einem relativen Ansatz verknüpft, da dies ausgezeichnete Ergebnisse erzielt.

## Gewählter Ansatz

Nach eingehender Betrachtung der oben Beschriebenen Ansätze in Kapitel 4.2.2.1 wurde der Ansatz der Trilateration gewählt, da er vergleichsweise einfach umzusetzen ist und die Genauigkeit alleine von der Entfernungsmessung abhängt.

Die Entfernungsmessung ist mit Hilfe verschiedener Methoden umsetzbar. Diese Methoden beinhalten die Laufzeit- oder Pegelmessung von verschiedenen Medien. Zur Auswahl stehen etwa Infrarot, WLAN, Bluetooth oder Ultraschall. Während existierende Systeme oft auf schon installierter WLAN-Infrastruktur aufbauen wird in das in dieser Arbeit entwickelte System auf Ultraschall basieren, da sich das Medium Schall mit relativ geringer Geschwindigkeit ausbreitet und deshalb mit vergleichsweise geringem Aufwand eine Laufzeitmessung ausgeführt werden kann. Diese Messung bei einem Medium das sich mit Lichtgeschwindigkeit ausbreitet würde die technischen Möglichkeiten dieser Arbeit übertreffen.

## Realisierung

# Individuelle Zielsetzung der Teammitglieder

## Individuelle Zielsetzung des Teammitglieds „Jakob Ecker“

### Aufgabenstellung

Jakob Ecker arbeitet hauptsächlich an der erforderlichen Software für den Roboter und das Positionierungssystem. Dafür erforderlich sind:

* Entwickeln des Trilaterationsalgorithmus
* Erstellen des Algorithmus für inertiale Positionierung
* Erstellen eins Programms zum Aufzeichnen, Speichern und Verwalten von Fahrwegen (sogenannten „Tracks“)
* Lösung des Problems der Zeitmessung zwischen Ultraschall Sende- und Empfangseinheit

Beim Entwickeln des Trilaterationsalgorithmus muss beachtet werden das in einer realen Umgebung die Wahrscheinlichkeit eine Messung zu erhalten bei dem die drei zugrundeliegenden Kreisgleichungen einen gemeinsamen Schnittpunkt haben sehr gering ist. Deshalb muss eine Lösung gefunden werden bei dem dies nicht erforderlich ist. Anschließend sind die benötigten mathematischen Herleitungen auszuführen und in einem Java-Programm zu implementieren. Des Weiteren müssen Fälle bedacht werden in denen eine mathematische Lösung nicht möglich ist (zum Beispiel wenn eine Division durch null auftritt). Der fertige Algorithmus ist als mit javadoc dokumentierte Methode zur Verfügung zu stellen.

Beim Algorithmus für inertiale Positionsbestimmung sind die größten Herausforderungen das starke Rauschen von Beschleunigungssensoren und das entstandene Fehler sich mit der Zeit immer weiter aufsummieren, auch wenn der Roboter stillsteht. Deshalb sind neben der eigentlichen Algorithmik Methoden zur Minimierung dieser Fehler zu entwickeln.

Zur Aufzeichnung, Speicherung und Verwaltung der Fahrdaten ist ein geeignetes Format zu finden. Es soll möglich sein Fahrtwege zu speichern, gespeicherte Fahrtwege wieder aufzurufen und nachzufahren. Des Weiteren soll es möglich sein die Fahrtwege zu exportieren und importieren um die Routen zwischen Robotern austauschen zu können.

Um die Entfernung zu messen wird eine Laufzeitmessung benötigt. Da die Sende- und die Empfangseinheit unterschiedliche Uhrzeiten haben ist dies problematisch. Diese Aufgabenstellung umfasst die Findung und Implementierung einer geeigneten Methode zur akkuraten Zeitmessung. In diesem Fall wird gefordert das der Messfehler <1ms ist, da selbst 1ms schon eine Abweichung von ca. 34cm bedeutet.

### Grundlagen und Methoden

#### Positionsbestimmung

Die Positionsbestimmung kann grundsätzlich als relatives oder absolutes System realisiert werden, wobei die relativen Systeme die Veränderung zur Startposition berechnen/aufzeichnen und die absoluten Systeme eine Position relativ zu einem Bekannten Bezugspunkt berechnen. Da jede Methode Vor- und Nachteile hat werden heutzutage meist zwei Methoden aus beiden Systemen kombiniert um möglichst gute Ergebnisse zu erzielen.

In diesem Abschnitt werden die verbreitetsten Methoden beider Systeme kurz vorgestellt und analysiert. Grundlage der Erläuterungen ist das Skriptum „Positionsbestimmung mit Ultraschall“ von Friedemann Sebastian Winkler [FRIE 11].

##### Odometrie

Die Odometrie ist ein häufig verwendeter Ansatz zur relativen Positionsbestimmung, bei dem Bewegungsinformation (z.B. Umdrehungen) dazu genutzt wird die zurückgelegte Strecke zu messen. Die Vorteile dieser Methode sind die einfache Umsetzbarkeit und mögliche hohe Abtastraten. Zudem ist die Präzision auf kurze Distanzen sehr gut, allerdings ist einer der größten Nachteile der integrale Fehler, der mit zunehmender Zeit/Distanz immer größer wird (Drift).

##### Inertiale Positionierung

Dieser ebenfalls relative Ansatz nutzt gemessene Beschleunigungen die zwei Mal integriert um so die zurückgelegte Distanz zu berechnen. Im Integrieren liegt auch der größte Nachteil dieses Systems, da jeder Noch so kleine Fehler immer weiter Anwächst, sogar während der Roboter stillsteht.

##### Triangulation

Die Triangulation ist eine Methode der Positionsbestimmung die eine absolute Position berechnet. Dafür sind (für die Navigation auf einer Ebene) mindestens drei Sender auf bekannten Positionen im Raum verteilt, zu denen ein rotierender Sensor an Bord des Roboters die zur Längsachse relativen Winkel misst. Daraus können die Koordinaten x, y und die Orientierung φ berechnet werden. Da diese Berechnungen allerdings nicht trivial sind und jeder Berechnungsansatz gravierende Nachteile hat wird diese Methode eher selten verwendet.

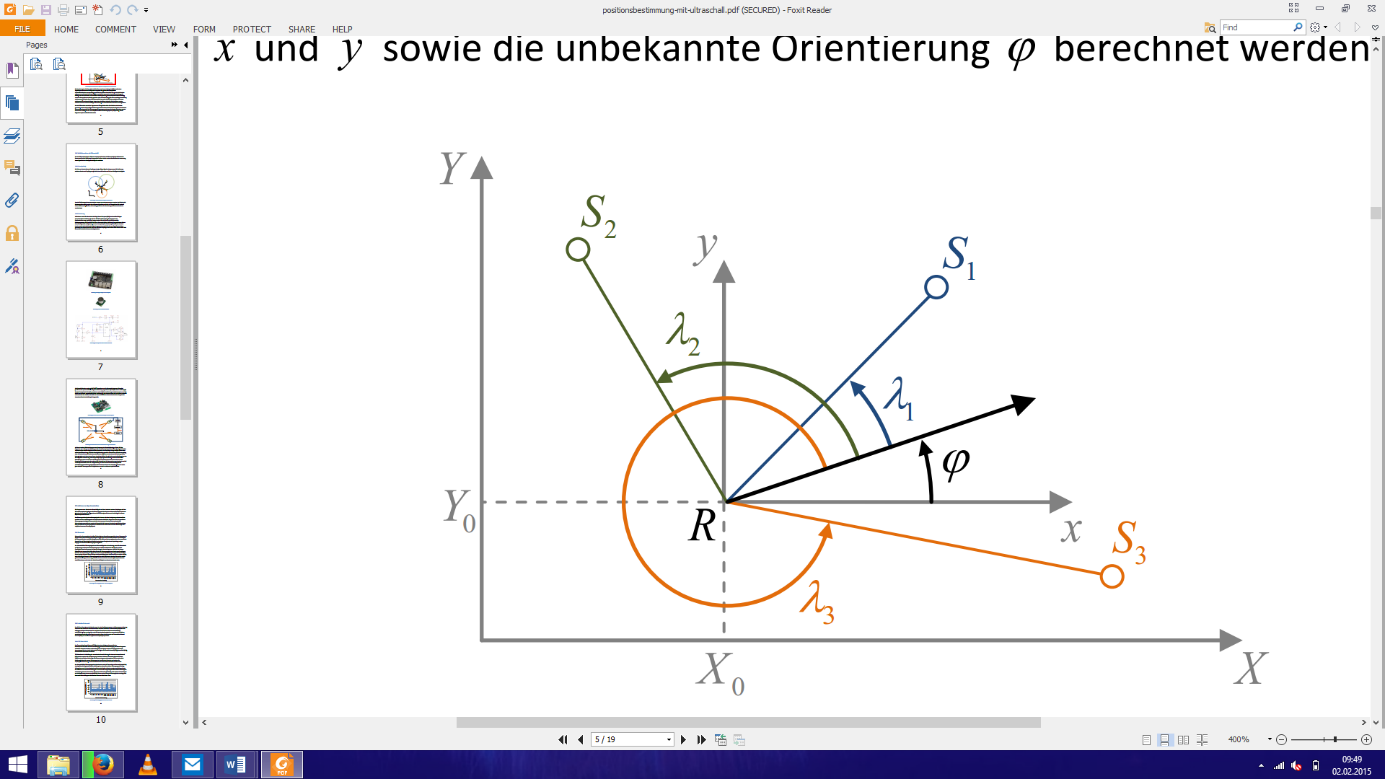


Abbildung ‑: Graphische Darstellung der Positionsberechnung mittels Triangulation [FRIE 11]

##### Trilateration

Die Trilateration ist die verbreitetste Methode zur absoluten Positionsberechnung. Sie wird beispielsweise von GPS genutzt und basiert ähnlich wie die Triangulation mit drei Bekannten Punkten (Sendern) berechnet wird. Allerdings misst man hier die Distanzen zu den Sendern, und nicht den Winkel. Die Berechnung ist auch hierbei recht anspruchsvoll, aber für das einwandfreie Funktionieren des Algorithmus müssen nicht so viele Bedingungen zutreffen. Die Größte Schwierigkeit ist, dass die Sender zur bekannten Zeiten senden müssen. Ein weiterer Nachteil ist das die Die Orientierung des Roboters nicht bestimmt werden kann.

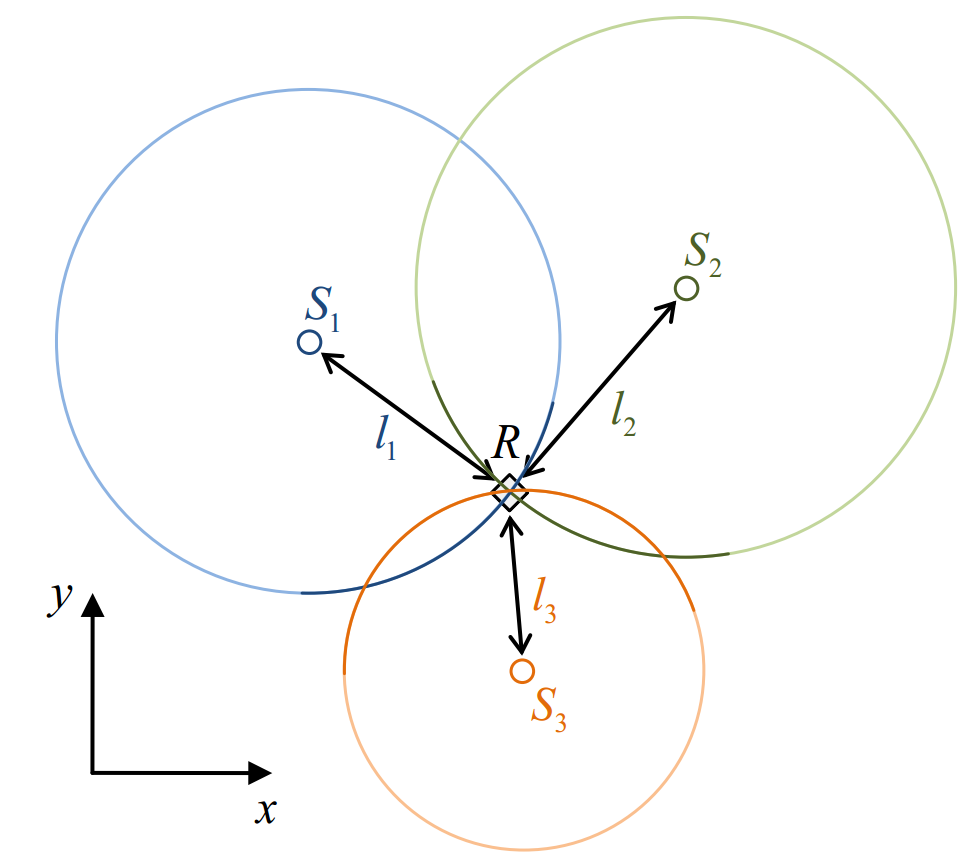


Abbildung 4.1‑2: Graphische Darstellung der Positionsberechnung mittels Trilateration [FRIE 11]

#### Entfernungsmessung

Um die Position bestimmen zu können müssen die Distanzen zu min. 3 Fixpunkten bekannt sein. Diese werden mittels Laufzeitmessung ermittelt. In diesem Projekt wird dazu das Medium Ultraschall verwendet.

Nach der Formel müssen dazu die Laufzeit des Ultraschallsignals und die Ausbreitungsgeschwindigkeit bekannt sein. Da die Schallgeschwindigkeit in der Luft bekannt (c=343m/s) ist hängt die Messung alleine von der Laufzeitmessung ab. Diese ist trotz der im Vergleich zu anderen Medien (z.B. Infrarot, WLAN) geringen Ausbreitungsgeschwindigkeit sehr Anspruchsvoll, da die Sende und Empfangseinheit notwendigerweise keine gemeinsame Zeitbasis haben (getrennte Geräte) und eine möglichst hohe Genauigkeit benötigt wird (1ms Fehler entspricht ca. 34cm Abweichung). Um eine präzise Zeitmessung zu ermöglichen gibt es mehrere Möglichkeiten, die im Folgenden erläutert werden:

##### Uhren-Synchronisation

Dabei werden die internen Uhren der Sende- und Empfangseinheit auf eine Abweichung von weniger als 1ms synchronisiert. Obwohl bereits Netzwerkprotokolle für diese Aufgabe existieren erweisen diese sich oft als zu ungenau oder sind nur mit großem Aufwand zu implementieren. Beispielsweise bietet das verbreitete *Network Time Protocol* (NTP) eine theoretische Genauigkeit von unter 200µs Abweichung, allerdings sind dafür ein ideales Netzwerk sowie stabile lokale Taktgeber notwendig. In schnellen LANs sind ca. 0,1ms Abweichung erreichbar [MILL 15]. Da bei SmartRobot aber WLAN eingesetzt werden soll um die Kabellose Verbindung zwischen Roboter und Sendestationen zu realisieren sind diese niedrigen Abweichungen nicht erreichbar (für WLAN ca. 2-3ms) [FRIE 11]. Zusätzlich zu der langsamen WLAN-Verbindung ist die interne *Real Time Clock* (RTC) in einem modernen Smartphone so unpräzise, dass selbst bei kurzen Zeitspannen (wenige Sekunden) bereits eine relativ große Abweichung entsteht. Aus diesen Gründen ist die Methode der Synchronisation nicht umsetzbar.

##### Synchronisation des Sendezeitpunkts

Hierbei sendet die Sendeeinheit ein UDP-Packet (*User Datagram Protocol*) an die Empfangseinheit um den Sendezeitpunkt bekannt zu geben. Die Ungenauigkeit wird durch die variable WLAN-Laufzeit verursacht. In einem unbelasteten Netzwerk liegt die durchschnittliche Laufzeit bei ca. 3ms. Der Fehler kann aber durch systematische Korrektur der Zeitmessung mit dem Durchschnittswert der WLAN-Laufzeit minimiert werden. Um die Genauigkeit weiter zu verbessern wird die durchschnittliche WLAN-Laufzeit periodisch oder sogar bei jedem Startsignal gemessen. Diese Methode ist wesentlich einfacher als die Synchronisation und mit den Beschriebenen Korrekturen kann der Fehler unter den Zielwert von <10cm gedrückt werden.

### Realisierung

#### Trilateration

Die Trilateration ist das Berechnen einer Position im Koordinatensystem anhand von min. 3 Punkten (im 2 dimensionalen Bereich, für 3D wird min. ein 4ter Punkt benötigt) und den Distanzen zu diesen Punkten. Als Ansatz für die Positionsbestimmung dient hierbei die allgemeine Kreisgleichung: . In diese kann man alle bekannten Größen einsetzen und erhält somit ein Gleichungssystem aus 3 Gleichungen mit 2 unbekannten. Die dritte Gleichung kann genutzt werden um eines der zwei Ergebnisse der quadratischen Gleichungen auszuwählen.

##### Anpassung an eine reale Umgebung

Da es sehr unwahrscheinlich ist das bei Messungen ein Ergebnis erzielt wird bei dem tatsächlich 3 Kreise einen gemeinsamen Schnittpunkt haben wird der Ansatz leicht abgewandelt. Es werden die Schnittpunkte jeder möglichen Kreiskombination berechnet (mit den Kreisgleichungen) und dann ein Dreieck aus den sich am nächsten liegenden konstruiert. Als Position wird nun der Schwerpunkt (Schnittpunk der Seitenhalbierenden) behandelt.

##### Mathematische Herleitung

###### Der Schnittpunkt zweier Kreise

Um die Herleitung zu vereinfachen, die Fehleranfälligkeit zu minimieren und die Rechenzeit zu optimieren wurde der Algorithmus noch weiter vereinfacht indem bei der Schnittpunktberechnung zweier Kreise der Koordinatenursprung in den Mittelpunkt eines Kreises verschoben wird. Dann ergibt sich aus den Mittelpunkten und die Werte und.

Nun erhält man zur Berechnung der Schnittpunkte folgendes Gleichungssystem:

Nach Anwendung des Subtraktionsverfahren:

Zur weiteren Vereinfachung werden die Variablen a und b eingeführt:

Zum Lösen nach y werden das Einsetzungsverfahren (in) und anschließend die pq-Lösungsformel für quadratische Gleichungen verwendet:

*Die pq-Lösungsformel:*

Bei gilt:

Zum Abschluss müssen nur noch die zugehörigen x-Werte mit obiger Formel berechnet werden.

###### Die Dreiecks-Konstruktion

Das Dreieck wird programmtechnisch mithilfe der Distanzen (s) zwischen den einzelnen Schnittpunkten ermittelt. Dabei wird darauf geachtet, dass nur jeweils ein Schnittpunkt eines Kreispaares ausgewählt wird. Die Distanzen werden mithilfe des Satz des Pythagoras berechnet:

###### Die Schwerpunktberechnung

Der Schwerpunkt eines Dreiecks wird entspricht dem Schnittpunkt der Seitenhalbierenden. Um die lineare Funktionen der Seitenhalbierenden aufzustellen, werden die Punkte, die die Seiten des Dreiecks halbieren, berechnet:

Nach dem der teilende Punkt für jede Seite des Dreiecks berechnet wurde kann gemeinsam mit dem Gegenüberliegenden Eckpunkt die lineare Funktion der Seitenhalbierenden () bestimmt werden:

Um den Schnittpunkt zweier Seitenhalbierenden und damit den Schwerpunkt zu berechnen müssen ihre Funktionen gleichgesetzt werden:

Der y-Wert des Schnitt- und Schwerpunktes kann ermittelt werden indem der errechnete x-Wert in eine der Funktionen eingesetzt wird.

.

### Ergebnisse

## Individuelle Zielsetzung des Teammitglieds „Dominik Simon“

### Aufgabenstellung

Die Aufgabe ist es hauptsächlich die Hardwarekomponenten der Diplomarbeit zu erstellen. Diese Umfassen als erstes die Auswahl der Roboterplattform und des Weiteren die Realisierung der Motoransteuerung und Entwicklung der Benötigten Hardware zur Ultraschallpositionierung (Sende- und Empfangseinheit). Im Zuge der Motoransteuerung wird auch die Kommunikation zwischen der Rechenplattform (Android) und der Steuereinheit (Arduino) erstellt. Zusätzlich werden die benötigten Filter softwareseitig implementiert und durch Fusion mehrerer Ansätze die Genauigkeit erhöht.

### Grundlagen und Methoden

#### Roboterplattform

Zur Veranschaulichung und zum Testen der, hier im Laufe der Diplomarbeit erarbeiteten, Plattformlösung wird ein fahrbarer Roboter entwickelt. Dieser soll einfach zu bedienen und sich stabil und kontrolliert bewegen können. Dazu werden mehrere Komponente benötig, zu einem der mechanische Roboter an sich, ein Antrieb und eine dazu passende Steuerung.

(Aus Zeitgründen werden die mechanischen Teile fertig dazugekauft, welche als Antrieb Gleichstrommotoren, die mit einem Getriebe versehen sind, benutzt.)

##### Motorsteuerung:

In diesem Kapitel wird die Ansteuerung von Gleichstrommotoren behandelt. Grundsätzlich wird Drehzahl und Drehrichtung gesteuert.

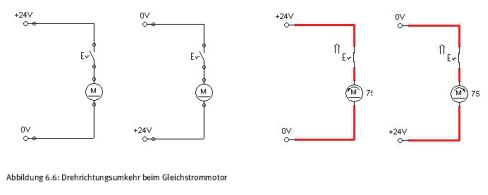


Abbildung ‑: Drehrichtungsumkehr beim Gleichstrommotor [LAND 14]

Um die Drehrichtung zu steuern muss nur die Polarität der Anschlüsse des Motors ausgetauscht werden (s. Abb. 4.2-1). Um das mechanische austauschen der Anschlüsse zu vermeiden, wird dazu der Motor an einer sogenannten H-Brücke angeschlossen (s. Abb. 4.2-2).

Wie in der Abbildung ersichtlich ist, kann die Polarität des Motors über die Transistoren geschaltet werden. Dabei muss beachtet, dass nur einer der diagonal liegende Transistor-Paare (Q1 / Q4 und Q2 / Q3) durchgeschaltet wird, sonst entsteht ein Kurzschluss. Wird dann das andere Transistor-Paar durchgeschaltet wird der Motor seine Drehrichtung ändern.

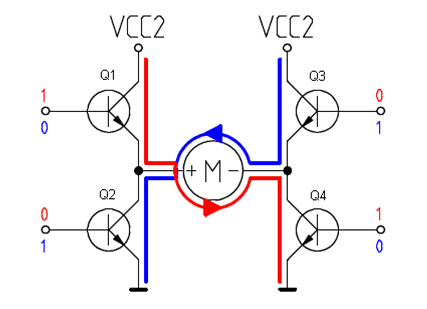


Abbildung ‑: H-Brücke

##### Kommunikation zwischen Rechenplattform und Steuereinheit

Um eine Kommunikation zwischen Rechenplattform und Steuereinheit zu realisieren wurden verschiedene Ansätze

#### Entfernungsmessung

Wie schon im Kapitel 4.1.2.1 Positionsbestimmung beschreiben wurde, wird eine Distanzmessung benötigt. In diesem Projekt wird dies auf Ultraschall-basis realisiert und eine dazu entsprechende Ultraschallhardware entwickelt. Diese Hardware besteht aus einen bzw. mehrere Sender und Empfänger, welche mit akustischen Signalen arbeiten, die über den Menschlichen Frequenzhörgang liegen.

##### Sender:

Ein Sender entspricht im Grunde einem einfachen Lautsprecher, welcher an die hohe Frequenz angepasst ist. Als Signalquelle dient ein Oszillator mit einer konstanten Frequenz von 20kHz.

##### Empfänger:

Der Empfänger ist grundsätzlich ein Mikrofon, das für hohe Frequenzen ausgelegt ist.

### Realisierung

### Ergebnisse

# Zusammenfassung

# Abkürzungsverzeichnis:

GPS Global Positioning System

GLONASS Globalnaja nawigazionnaja sputnikowaja sistema

Rus. Für Globales Satellitennavigationssystem

WLAN Wireless Local Area Network

bzw. beziehungsweise

s. Abb. siehe Abbildung

ca. circa

cm Zentimeter

z.B. zum Beispiel

# Literatur und/oder Quellenverzeichnis

* [MATH 19.06.2014] Mathematik Trilateration: Matroids Matheplanet (Forum, Zugriff 19.06.2014): <http://matheplanet.com/default3.html?call=viewtopic.php?topic=123940&ref=http%3A%2F%2Fwww.google.at%2Furl%3Fsa%3Dt%26rct%3Dj%26q%3D%26esrc%3Ds%26source%3Dweb%26cd%3D2%26ved%3D0CCcQFjAB>
* [MILL 15] David Mills, Network Time Synchronization Research Project; Zugriff: 10.02.2015: http://www.eecis.udel.edu/~mills/ntp.html
* [FRIE 11] Friedemann Sebastian Winkler, 29.11.2011, „Positionsbestimmung mit Ultraschall“; Zugriff am 19.02.2015: <http://me-lrt.de/pa/positionsbestimmung-mit-ultraschall.pdf>
* [NATH 15] Martin Nathansen, GPS-Genauigkeit und Einschränkungen; Zugriff 15.02.2015: <http://gpso.de/technik/gpsgenau.html>
* [LAND 14] Landesakademie für Fortbildung und Personalentwicklung an Schulen, Ansteuerung von Gleichstrommotoren; Zugriff: 06.09.2014: <http://lehrerfortbildung-bw.de/faecher/nwt/fb/atechnik/grundlagen/de/kapitel/abb_66.jpg>
* [CHPD] Fachhochschule Wiesbaden - Fachbereich Informatik: H-Brücke; Zugriff: 06.09.2014: <http://www.cs.hs-rm.de/~linn/vpdv05/aufzug/hbruecke.html>

# Anhang (Dokumente und Protokolle)

## Projektdokumentation

### Pflichtenheft

### Organisationsstruktur

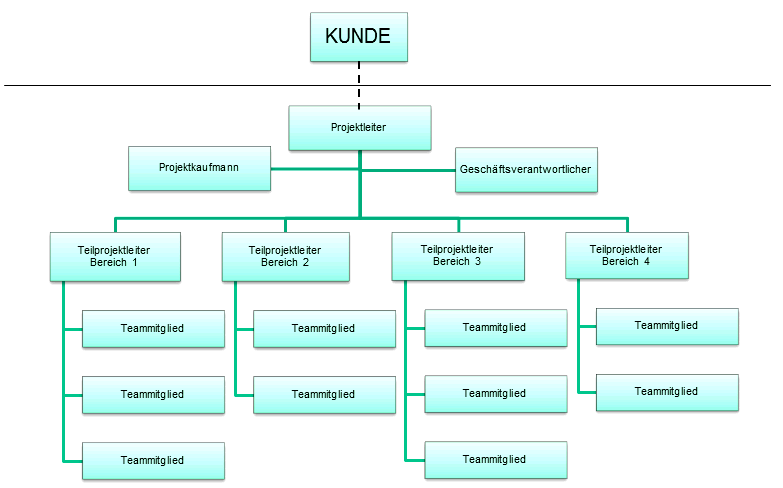
Die Organisationsstruktur beschreibt das Zusammenwirken aller Beteiligten im Projekt, deren Funktionen und Verantwortlichkeiten und die Fachgebiete, die von den Teammitgliedern

Abbildung ‑: Organisationsstruktur

eigenständig bearbeitet werden sowie die Beziehung zum Auftraggeber bzw. dem Kunden.

### Projektstruktur



Abbildung 8.1‑2: Projektstrukturstruktur

### Terminplan

Der Terminplan beschreibt den zeitlichen Ablauf der Projektumsetzung. Stellen Sie auf der Zeitachse die Dauer der geplanten Projektphasen und Arbeitspakete, wie unter Punkt 7.1.4 beschrieben, dar.

Verwenden Sie dazu das Planungstool MS Project und fügen Sie entsprechende Screenshots ein. Diese Screenshots, wie in Abbildung 7-5 zu sehen, müssen lesbar sein, eventuell teilen Sie die Gesamtübersicht auf mehrere Grafiken auf. Es ist vor allem darauf zu achten, dass Zeitachsen und Vorgangsachsen auf jeder Grafik sichtbar sind!

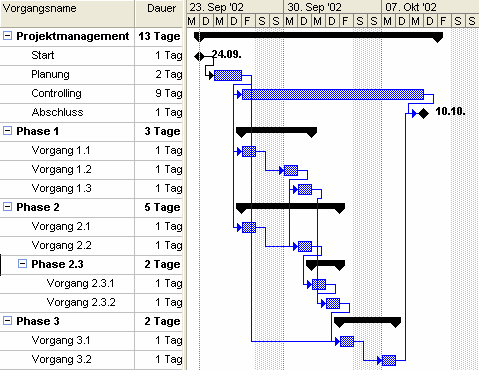


Abbildung ‑: Projektbalkenplan

### Kalkulation

Für die Kalkulation im Gesamtprojekt sind folgende Kosten zu erfassen:

Personalaufwand (kalkulatorische Richtsätze laut WIRE-Unterricht)

Kosten für Material (Hard- und Software)

externe Kosten (z.B.: Zukauf von Sensoren, Funkmodule, spezielle Entwicklungsumgebungen, etc.)

Reisekosten (Fahrtkosten und Spesenersatz)

Eine Zusammenstellung der Gesamtkosten ist zu sehen in Tabelle 7-1:

|  |  |  |  |
| --- | --- | --- | --- |
| 2014\_12/01 |  | PROJEKTKOSTEN | 11.06.2014 |
| Maturaprojekt Anlage 01 |  |  | KOSTEN EUR |
| Material (eigen) | 1a |  | 0,0 |
| Material (fremd) | 1b |  | 0,0 |
|  | 1c |  | 0,0 |
| Kabel | 1d |  | 0,0 |
|  | 1e |  | 0,0 |
| Material, Produkte gesamt | KA1 |  | 0,0 |
| Montage (eigen) | 2a | 0 h | 0,0 |
| Montage (fremd) | 2b | 0 h | 0,0 |
| Inbetriebnahme (Kalttest) | 2c | 0 h | 0,0 |
|  | 2d | 0 h | 0,0 |
| Verlegen (Kabel) | 2e | 0 h | 0,0 |
| Anschluss (Kabel) | 2f | 0 h | 0,0 |
|  | 2g | 0 h | 0,0 |
| Montage gesamt | KA2 | 0 h | 0,0 |
| Ing.-Leistung Vertrieb | 3a | 0 h | 0,0 |
| Programmierung / FAT | 3b | 0 h | 0,0 |
| Erstellung Software | 3c | 0 h | 0,0 |
| Inbetriebnahme Hardware | 3d | 0 h | 0,0 |
| Erstellung CAD-Pläne | 3e | 0 h | 0,0 |
| Engineering | 3f | 0 h | 0,0 |
| Projektleitung/Bauleitung | 3g | 0 h | 0,0 |
|  | 3h | 0 h | 0,0 |
| Dienstreisen | 3i | 0 h | 0,0 |
| Nebenkosten | 3j | 0 h | 0,0 |
| Ing. Sonst. Leistungen | KA3 | 0 h | 0,0 |
| Gesamt ------------------------> | KA1+2+3 |  | 0,0 |

Tabelle 7.1‑1: Kostenübersicht

## Besprechungsprotokolle

Fügen Sie an dieser Stelle alle Dokumente, die im Zuge der Projektabwicklung erstellt wurden, an:

Projektantrag

Protokoll des Projektstartgespräches

Protokoll der Projektstatussitzungen

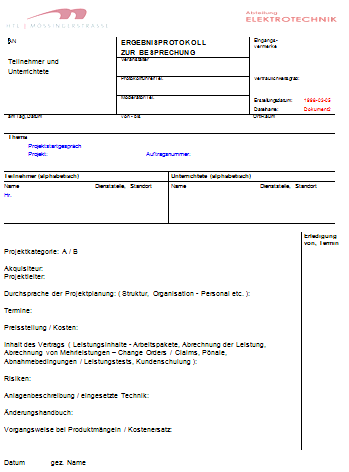
z.B.

Abbildung ‑: Besprechungsprotokoll vom 22.11.2014

## Arbeitszeitnachweise

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ecker Jakob** | | | | | |
| **Datum** | **Von** | **Bis** | **Pause** | **Arbeitszeit** | **Arbeiten** |
| 09.06.2014 | 14:30 | 16:30 |  | 02:00 | Recherche, Definition, Brainstorming |
| 11.06.2014 | 15:00 | 17:00 |  | 02:00 | Recherche, Definition, Brainstorming |
| 12.06.2014 | 15:00 | 17:00 |  | 02:00 | Auswahl Mobiles OS |
| 16.06.2014 | 15:00 | 16:00 |  | 01:00 | Einlesen Android |
| 18.06.2014 | 14:00 | 17:00 |  | 03:00 | Einlesen Android |
| 19.06.2014 | 15:00 | 17:00 |  | 02:00 | Trilaterateratrion - Math. Hintergrund |
| 24.06.2014 | 13:30 | 19:00 | 01:30 | 04:00 | Trilaterateratrion - Math. Hintergrund |
| 26.06.2014 | 15:00 | 17:00 |  | 02:00 | java: Trilateration implementierung |
| 27.06.2014 | 14:00 | 15:00 |  | 01:00 | java: Trilateration implementierung |
| 30.06.2014 | 16:00 | 17:00 |  | 01:00 | java: Trilateration implementierung |
| 02.07.2014 | 14:30 | 17:00 |  | 02:30 | java: Trilateration implementierung |
| 04.07.2014 | 15:00 | 18:00 |  | 03:00 | java: Trilateration implementierung |
| 19.07.2014 | 15:00 | 18:00 |  | 03:00 | java: Trilateration v2, Fehlerentfernung, debuggen |
| 26.07.2014 | 15:00 | 22:00 | 00:30 | 06:30 | java: Trilateratin v2 fertigstellen, Portierung auf Android |
| 27.07.2014 | 14:00 | 19:00 | 02:00 | 03:00 | Doku Trilateration, Test/Demo GUI |
| 11.08.2014 | 16:00 | 18:15 |  | 02:15 | java: ECP Kontrolle |
| 27.08.2014 | 14:45 | 21:00 |  | 06:15 | Arduinoprogramm Kontrolle, testen |
| 07.09.2014 | 14:05 | 19:20 |  | 05:15 | USB-Accessory, Logo Idee |
| 09.09.2014 | 14:15 | 19:05 |  | 04:50 | Recherche Synchronisation, Logo entwerfen |
| 17.09.2014 | 15:05 | 15:55 |  | 00:50 | Accessory Kontrolle |
| 20.09.2014 | 14:00 | 16:10 |  | 02:10 | Doku schreiben, Raspberry Recherche |
| 20.09.2014 | 19:30 | 21:05 |  | 01:35 | Android Sensorzugriff testen |
| 24.09.2014 | 13:40 | 15:50 |  | 02:10 | Distanzberechnung |
| 11.10.2014 | 12:45 | 18:00 | 02:00 | 03:15 | Doku schreiben |
|  |  |  |  | **66:35** | **Gesamt** |

Tabelle 7.3‑1 Arbeitszeitnachweis Herr Ecker

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Simon Dominik** | | | | | | | | |
| **Datum** | | **Von** | | **Bis** | | **Pause** | **Arbeitszeit** | **Arbeiten** |
| 09.06.2014 | | 14:30 | | 16:30 | |  | 02:00 | Recherche, Definition, Brainstorming |
| 11.06.2014 | | 15:00 | | 17:00 | |  | 02:00 | Recherche, Definition, Brainstorming |
| 12.06.2014 | | 15:00 | | 17:00 | |  | 02:00 | Auswahl Hardware, Roboterplattform |
| 16.06.2014 | | 15:00 | | 16:00 | |  | 01:00 | Einlesen UDOO |
| 18.06.2014 | | 14:00 | | 17:00 | |  | 03:00 | Einlesen UDOO |
| 19.06.2014 | | 15:00 | | 17:00 | |  | 02:00 | Recherche: Filter |
| 24.06.2014 | | 13:30 | | 16:00 | |  | 02:30 | Aufbau Steckplatine: Motorsteuerung |
| 26.06.2014 | | 15:00 | | 17:00 | |  | 02:00 | Trilaterateratrion: Kontrolle |
| 27.06.2014 | | 14:00 | | 15:00 | |  | 01:00 | Platinendesign Motoransteuerung |
| 30.06.2014 | | 16:00 | | 17:00 | |  | 01:00 | Kommunikation Arduino-Android, Protokolldefinition (ECP) |
| 02.07.2014 | | 14:30 | | 17:00 | |  | 02:30 | Kommunikation Arduino-Android, Protokolldefinition (ECP) |
| 04.07.2014 | | 15:00 | | 18:00 | |  | 03:00 | Java: Trilateration Kontrolle |
| 19.07.2014 | | 15:00 | | 18:00 | |  | 03:00 | Trilaterationv2, Fehlerentfernung, debuggen |
| 26.07.2014 | | 15:00 | | 22:00 | | 00:30 | 06:30 | Recherche Kalmanfilter, Programmierung ECP |
| 27.07.2014 | | 17:30 | | 19:00 | |  | 01:30 | Doku kontrolllesen, Test-App erstellen |
| 24.08.2014 | | 18:00 | | 21:00 | |  | 03:00 | Hardwareaufbau, Adurino IDE einrichten |
| 25.08.2014 | | 12:00 | | 14:00 | |  | 02:00 | Arduino: Hardwareansteuerung, USB |
| 27.08.2014 | | 14:45 | | 21:00 | |  | 06:15 | USB-Ardurino, Aufsetzen PC und Entwicklungsumgebung |
| 05.08.2014 | | 18:00 | | 20:00 | |  | 02:00 | PC konfiguration |
| 07.09.2014 | | 14:05 | | 19:20 | |  | 05:15 | USB-Accessory einlesen, Logo Idee |
| 09.09.2014 | | 14:15 | | 19:05 | |  | 04:50 | Accessory programmieren, Fehlersuche: Ardurino, Logo entwerfen |
| 10.09.2014 | | 18:00 | | 20:00 | |  | 02:00 | Logo erstellen |
| 11.09.2014 | | 16:00 | | 20:00 | |  | 04:00 | Accessory testen, Fehlersuche |
| 17.09.2014 | | 15:05 | | 15:55 | |  | 00:50 | Usb Handler implementieren |
| 20.09.114 | | 14:00 | | 16:10 | |  | 02:10 | Doku schreiben, Raspberry einrichten |
| 24.09.2014 | | 13:40 | | 15:50 | |  | 02:10 | ECPv2.1 updaten, |
| 11.10.2014 | | 12:45 | | 18:00 | | 02:00 | 03:15 | Doku schreiben |
|  |  | |  | |  | | **72:45** | **Gesamt** |

Tabelle 7.3‑2 Arbeitszeitnachweis Herr Simon

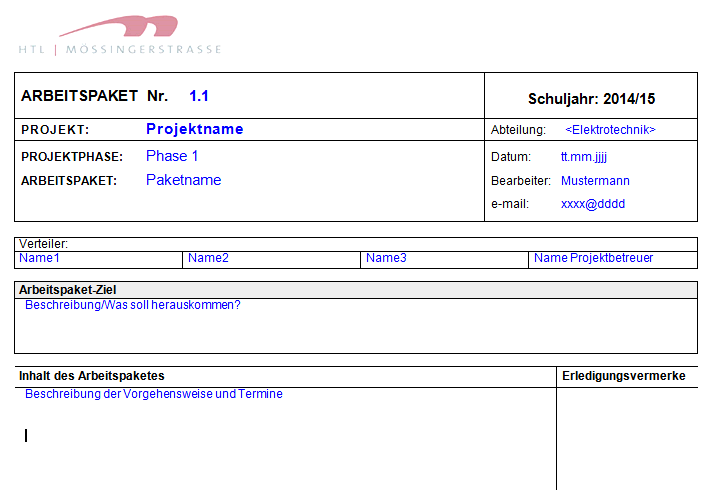
Wichtiger Hinweis:

Die Arbeitszeit pro Schüler, die außerhalb der Unterrichtszeit für die Bearbeitung der Diplomarbeit zu erbringen ist, ist in einem Ausmaß von 150 Stunden bis 180 Stunden nachzuweisen!

## Technische Dokumentation

Fügen Sie hier technische Beschreibungen, Berechnungen, Konstruktionszeichnungen, und Versuchsberichte, etc. ein.

## Arbeitspakete

Fügen Sie hier die Beschreibungen der Arbeitspakete ein.

## Businessplan (Optional)

Halten Sie sich, was die Struktur des Businessplanes angeht, an die Angaben, die Sie im WIRE-Unterricht bekommen.

# Abbildungsverzeichnis

[Abbildung 4.1‑1: Graphische Darstellung der Positionsberechnung mittels Triangulation [FRIE 11] 9](file:///C:\Users\Dominik\Desktop\Gesamtdokumentation_v2.docx#_Toc411850306)

[Abbildung 4.1‑2: Graphische Darstellung der Positionsberechnung mittels Trilateration [FRIE 11] 9](file:///C:\Users\Dominik\Desktop\Gesamtdokumentation_v2.docx#_Toc411850307)

[Abbildung 4.2‑1: Drehrichtungsumkehr beim Gleichstrommotor [LAND 14] 14](file:///C:\Users\Dominik\Desktop\Gesamtdokumentation_v2.docx#_Toc411850308)

[Abbildung 4.2‑2: H-Brücke 15](file:///C:\Users\Dominik\Desktop\Gesamtdokumentation_v2.docx#_Toc411850309)

[Abbildung 8.1‑1: Organisationsstruktur 19](file:///C:\Users\Dominik\Desktop\Gesamtdokumentation_v2.docx#_Toc411850310)

[Abbildung 8.1‑2: Systemstruktur 22](file:///C:\Users\Dominik\Desktop\Gesamtdokumentation_v2.docx#_Toc411850311)

[Abbildung 8.1‑3: Projektbalkenplan 23](file:///C:\Users\Dominik\Desktop\Gesamtdokumentation_v2.docx#_Toc411850312)

[Abbildung 8.2‑1: Besprechungsprotokoll vom 22.11.2014 26](file:///C:\Users\Dominik\Desktop\Gesamtdokumentation_v2.docx#_Toc411850313)

[Tabelle 7.1‑1: Kostenübersicht 22](#_Toc401048890)

[Tabelle 7.3‑1 Arbeitszeitnachweis Herr Ecker 24](#_Toc401048891)

[Tabelle 7.3‑2 Arbeitszeitnachweis Herr Simon 25](#_Toc401048892)