

SYSTEMSOFTWARE UND
VERTEILTE SYSTEME

Fakultät II - Department für Informatik

Masterthesis

Automatisierung einer
Berechnungsmaschine

Marvin Banse, B.Sc. Informatik

&

Florian Schmalriede, B.Sc. Informatik

Beginn SS 19 - Ende WS 19

1. Gutachter & Betreuer: Prof. Dr.-Ing. Oliver Theel
2. Gutachter: Prof. Dr. Andreas Winter
Betreuer: Dipl.-Ing. Michael Pokriefke



15. März 2020

Inhaltsverzeichnis

1	Einleitung	3
1.1	Aufgabenstellung	4
1.2	Zielsetzung	4
1.3	Initiale Lösungsidee	7
1.4	Kriterien zur Einschätzung der Lösung	8
1.5	Aufbau des Dokuments	9
2	Projektbeschreibung	10
2.1	Systemumgebung	10
2.2	Systembeschreibung	12
2.3	Vergleichbare Systeme	15
3	Anwendungsfälle	17
3.1	Fahrspuren kartieren	17
3.2	Berechnungssequenz verwalten	20
3.3	Berechnungsplan verwalten	21
3.4	Fläche berechnen	22
3.5	Berechnung überwachen	25
4	Architektur	28
4.1	Berechnungsmaschine	30
4.1.1	Hardware	30
4.1.2	Software	32
4.2	Kartiergerät	34
4.2.1	Hardware	34
4.2.2	Software	35
5	Realisierung	36
5.1	Grundlagen	36
5.1.1	Robot Operating System	36
5.1.2	Mathematisches Modell	36
5.1.3	Betriebsmodi	39
5.1.4	Fehlerverhalten	40
5.2	Spannungsversorgung	41
5.2.1	Batterieüberwachung	42
5.2.2	Schaltung der Versorgungsspannung	45

5.3	Steuerung	46
5.3.1	Hardware	46
5.3.2	Positionsregelung der Linearantriebe	53
5.3.3	Schaltung der Wasserversorgung	58
5.4	Lokalisierung	59
5.4.1	Odometrie	61
5.4.2	RTK	63
5.4.3	Sensorfusion	66
5.4.4	Lageerkennung	66
5.5	Kartierung	69
5.6	Berechnungspfad folgen	70
5.6.1	Bestimmung des aktuellen Berechnungssegment	75
5.6.2	Berechnung von Stellungparameter	75
5.6.3	Berechnung der Simulationszeit	76
5.6.4	Berechnung von Trajektorien	77
5.6.5	Bewertung von Trajektorien	78
5.7	Beregnung	79
6	Test & Optimierung	81
7	Zusammenfassung	84
8	Ausblick	85
A	Interview	87
B	Anforderungsliste	88
C	Konstruktionszeichnungen	92
C.1	Schaltplan	92
C.2	Platinenlayout	93
C.3	Kartiergerät	94
	Literatur	95
	Glossar	99

1 | Einleitung

In der Landwirtschaft ist die Beregnung von Nutzflächen von zentraler Bedeutung. Eine gute Beregnung steigert den Ertrag, verursacht aber auch Bewirtschaftungsaufwand und damit Kosten [1]. Es gibt unterschiedliche Lösungsansätze eine Beregnung umzusetzen. Die sogenannte Beregnungsmaschine ist ein Beregnungssystem, welches, wie der Name es suggeriert, häufig zur Beregnung von Flächen eingesetzt wird. Hierzu wickelt sie einen manuell verlegten, mit der Wasserversorgung verbundenen Schlauch auf, zerstäubt das ankommende Wasser und beregnet so gleichzeitig die abgefahrene Fläche entlang des Schlauchs. Gegenüber anderen Beregnungssystemen zeichnet sie sich durch ihren flexiblen Einsatz aus, da zusätzlich zum Wasseranschluss keine feste Installation benötigt wird.

Beregnungsmaschinen können grob in zwei Bauarten unterteilt werden [2]. Bei der ersten Bauart, wie in Abbildung 1.1a dargestellt, befindet sich eine Haspel auf einem Gestell, welches durch Stützen fixiert wird. Die Haspel ist durch einen Zugschlauch mit dem sogenannten Regenwagen verbunden. Dieser ist mobil und verfügt über eine Beregnungsvorrichtung. Er wird mit Hilfe einer Zugmaschine auf die gegenüberliegende Seite der jeweiligen Fläche gezogen. Wenn der Regenwagen positioniert worden ist, wird die Wasserzufuhr geöffnet und die Haspel über eine Turbine angetrieben. Die Turbine nutzt hierzu einen Teil des anliegenden Wasserdrucks. Die Haspel wickelt den Zugschlauch auf und zieht so den Regenwagen zurück zum Gestell. Währenddessen beregnet der Regenwagen über die Beregnungsvorrichtung die Fläche.

Die zweite Bauart, wie in Abbildung 1.1b dargestellt, verzichtet auf einen Regenwagen. Das Gestell steht auf Rädern und wird mit Hilfe einer Zugmaschine positioniert. Dabei wird der Schlauch während der Positionierung abgewickelt. Wenn die Beregnungsmaschine positioniert worden ist, wird die Wasserzufuhr geöffnet und die Haspel über eine Turbine angetrieben. Die Turbine nutzt hierzu einen Teil des anliegenden Wasserdrucks. Die Haspel wickelt den Schlauch auf und zieht sich selbst mit der Beregnungsmaschine in Richtung des Wasseranschlusses am Schlauch entlang. Die Beregnung erfolgt dabei durch eine auf der Maschine angebrachte Beregnungsvorrichtung.

Wenn der Regenwagen an die Beregnungsmaschine gezogen wurde (erste Bauart) oder die Beregnungsmaschine an den Wasseranschluss gezogen wurde (zweite Bauart), stoppt die Beregnungsmaschine. Um ggf. die Beregnung fortzusetzen und weitere Flächenabschnitte zu beregnen muss die Maschine manuell neu positioniert werden. Der daraus entstehende Arbeitsaufwand führt zur Aufgabenstellung dieser Abschlussarbeit (siehe Abschnitt 1.1).

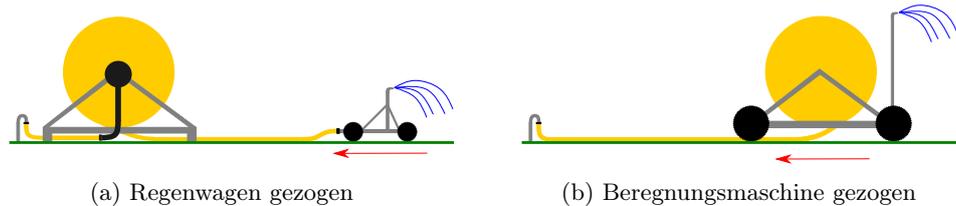


Abbildung 1.1: Bauarten von Beregnungsmaschinen

1.1 Aufgabenstellung

Durch die wiederholte Neupositionierung von Regenwagen oder Beregnungsmaschine entsteht mehrmals täglich ein stupider Arbeitsaufwand. Vor allem wenn der Weg zur Beregnungsmaschine weit ist, dann ist dieser Aufwand beträchtlich. Wenn die Beregnungsmaschine nachts betrieben wird und somit der Aufwand außerhalb der üblichen Arbeitszeiten bewerkstelligt werden muss, verursacht dieser eine erhebliche Belastung des Arbeiters. Mit dieser Problematik und der Frage nach einer möglichen Lösung ist der Geschäftsführer Dip.-Ing. Michael Prokriefke der seed2soil GmbH & Co. KG¹ an die Abteilung Systemsoftware und verteilte Systeme² der Universität Oldenburg herangetreten. Die Abteilung Systemsoftware und verteilte Systeme hat sich dieser Fragestellung angenommen und versucht die Problematik im Rahmen einer Abschlussarbeit zu lösen. Das konkrete Ziel ist die Automatisierung einer Beregnungsmaschine, welches im Abschnitt 1.2 ausgeführt wird.

1.2 Zielsetzung

Der wiederkehrende und stupide Arbeitsaufwand der Neupositionierung von Regenwagen oder Beregnungsmaschine soll durch die entsprechende Automatisierung einer Beregnungsmaschine eingespart werden. Außerdem ist es aufgrund der Automatisierung möglich, dass die Beregnungsmaschine die Beregnungen eigenständig protokolliert. Dieses unterstützt den Betreiber der Beregnungsmaschine bei der Planung und Überwachung von Beregnungen.

Aufgrund von baulich bedingten Gegebenheiten, eignet sich eine Beregnungsmaschine der ersten Bauart nicht so gut für die Automatisierung. Die zentralen Gründe dafür sind:

1. Der Regenwagen wird an einem Zugschlauch an die Haspel herangezogen. Somit sind keine Kurvenfahrten möglich. Dies führt dazu, dass eine manuelle Neupositionierung, um einen anderen Teil der Fläche zu beregnen, vorgenommen werden muss. Daher wäre eine automatisierte Beregnungsmaschine der ersten Bauart nur für Flächen geeignet, welche mit einer geraden Abfahrt komplett beregnet werden können.
2. Der Regenwagen verfügt über keinen Antrieb. Daher müsste der Regenwagen um einen Antrieb ergänzt werden, um automatisch positioniert zu werden. Dabei müsste der Antrieb des Regenwagens genügend Kraft und Traktion aufbringen, um den

¹<https://www.seed2soil.de>

²<https://uol.de/svs>

Zugschlauch auszuziehen und sich selbst zu positionieren. Dies ist eine große technische Herausforderung, da zwischen Schlauch und Berechnungsfläche eine immer größer werdende Reibungsfläche entsteht. Die aufzubringende Kraft wird mit wachsender Schlauchlänge so groß, dass diese nicht allein durch den Wasserdruck aufgebracht werden kann.

Daher wird die Automatisierung einer Berechnungsmaschine der zweiten Bauart als Ziel gesetzt.

Das Ziel bei der automatisierten Berechnung ist die Einhaltung eines zeitlich geplanten Ablaufes, welcher die Berechnung der jeweiligen Fläche zu gegebenen Zeitpunkten sicherstellt. Um dies zu formalisieren wurden die Definitionen 1 bis 5 erstellt und folgend beschrieben. Eine zeitliche Planung von Berechnungen wird als Berechnungsplan bezeichnet und ist nach Definition 5 eine zeitlich geordnete Liste von sogenannten Berechnungsaufgaben. Eine Berechnungsaufgabe fasst nach Definition 4 den Startzeitpunkt der Berechnung und eine sogenannte Berechnungssequenz zusammen. Eine Berechnungssequenz ist nach Definition 3 eine Liste von sogenannten Berechnungspfaden. Ein Berechnungspfad ist nach Definition 1 eine Liste von Koordinaten, die während einer Berechnung nacheinander angefahren werden sollen. In einer Berechnungssequenz gilt nach Definition 3, dass jeder Berechnungspfad bpf_i mit seinem nachfolgenden Berechnungspfad bpf_{i+1} und der letzte Berechnungspfad bpf_n mit dem ersten Berechnungspfad bpf_1 *zusammenhängt*. Nach Definition 2 werden zwei Berechnungspfade $bpf_a = \langle k_{a,1}, \dots, k_{a,m} \rangle$ und $bpf_b = \langle k_{b,1}, \dots, k_{b,l} \rangle$ als *zusammenhängend* bezeichnet, wenn die letzte Koordinate $k_{a,m}$ des ersten Berechnungspfades bpf_a mit der ersten Koordinate $k_{b,1}$ des folgenden Berechnungspfades bpf_b identisch ist. Somit kann innerhalb einer Berechnungssequenz ein Berechnungspfad nach dem anderen abgefahren werden.

Definition 1 *Berechnungspfad*

Ein **Berechnungspfad** bpf ist eine Liste von Koordinaten eines beliebigen, aber festen Koordinatensystems mit $bpf = \langle k_1, \dots, k_n \rangle$ und $n \in \mathbb{N}$.

Definition 2 *Zusammenhängende Berechnungspfade*

Ein Berechnungspfad bpf_a mit $bpf_a = \langle k_{a,1}, \dots, k_{a,n} \rangle$ und $n \in \mathbb{N}$ **hängt zusammen** mit einem anderen Berechnungspfad bpf_b mit $bpf_b = \langle k_{b,1}, \dots, k_{b,m} \rangle$ und $m \in \mathbb{N}$ genau dann, wenn gilt $k_{a,n} = k_{b,1}$.

Definition 3 *Berechnungssequenz*

Eine **Berechnungssequenz** $bseq$ ist eine Liste von Berechnungspfaden im Zeichen $bseq = \langle bpf_1, \dots, bpf_n \rangle$ und $n \in \mathbb{N}$, wobei bpf_n mit bpf_1 zusammenhängt und für alle in der Sequenz aufeinanderfolgenden Berechnungspfade bpf_i und bpf_{i+1} mit $i \in \{0, \dots, n-1\}$ gilt: bpf_i hängt mit bpf_{i+1} zusammen.

Definition 4 *Berechnungsaufgabe*

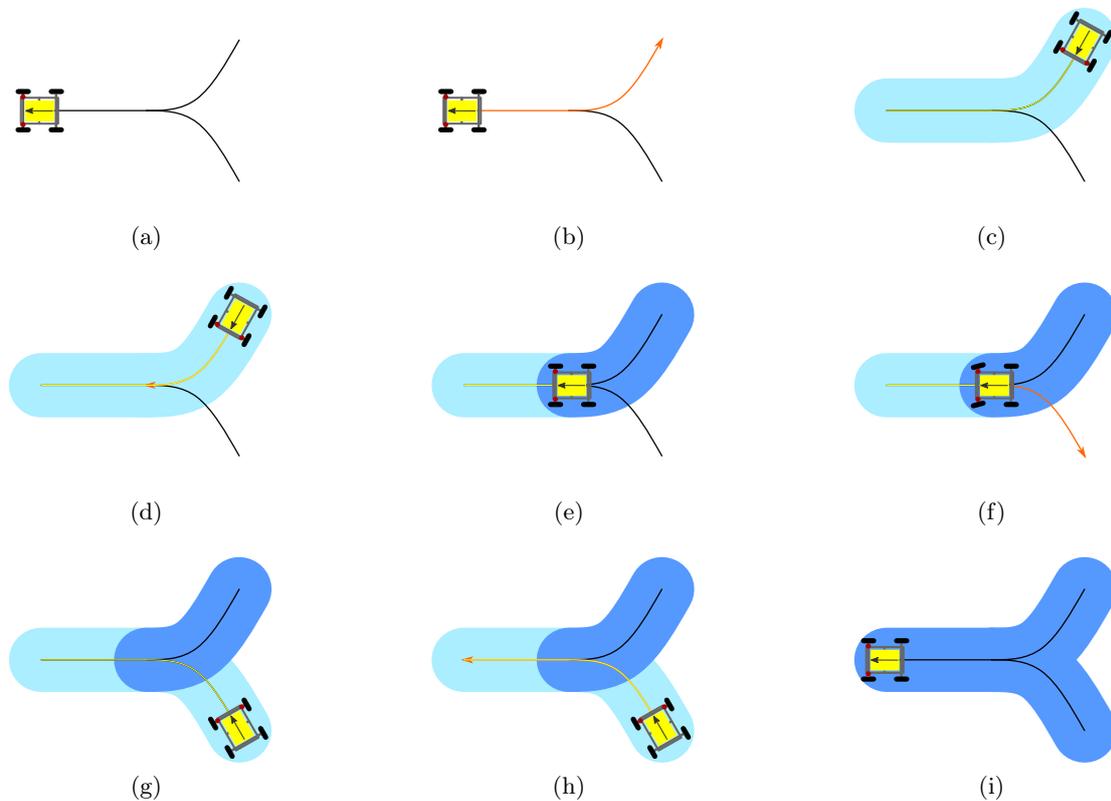
Eine **Berechnungsaufgabe** ba ist ein Zwei-Tupel mit $ba = (t, bseq)$, t ist ein Zeitpunkt und $bseq$ ist eine Berechnungssequenz.

Definition 5 *Berechnungsplan*

Ein **Berechnungsplan** bpl ist eine Liste von Berechnungsaufgaben mit $bpl = \langle ba_1, \dots, ba_n \rangle$

und $n \in \mathbb{N}$, wobei für alle in der Liste aufeinanderfolgenden Berechnungsaufgaben ba_i und ba_{i+1} mit $ba_i = (t_i, bseq_i)$, $ba_{i+1} = (t_{i+1}, bseq_{i+1})$ und $i \in \{1, \dots, n-1\}$ gilt: $t_i < t_{i+1}$.

In Abbildung 1.2 ist beispielhaft die Abarbeitung einer Berechnungssequenz, bestehend aus vier Berechnungspfaden, dargestellt. Jedes Einzelbild der Abbildung 1.2 stellt einen anderen Zeitpunkt während der Abarbeitung dar. Dabei entspricht das graue Rechteck mit einem Pfeil in der Mitte und den Rädern an der Seite der Berechnungsmaschine. Der Pfeil auf der Berechnungsmaschine zeigt in die Richtung der gelenkten Achse. Die Berechnungssequenz wird durch die schwarzen Linien dargestellt. Hellblau umrandete Teilstücke der Berechnungssequenz zeigen Flächen an, die einmal berechnet wurden. Dunkelblau umrandete Teilstücke der Berechnungssequenz zeigen Flächen an, die mindestens zweimal berechnet wurden. Der Berechnungspfad, welcher als nächstes abgefahren werden soll, ist durch einen orangenen Pfeil dargestellt.



Legende



Abbildung 1.2: Beispielhafte Abarbeitung einer Berechnungssequenz

Um einen Berechnungsplan starten zu können erfolgt die manuelle Platzierung der Berechnungsmaschine. Ist diese richtig positioniert, wird der erste Berechnungspfad aus der Berechnungssequenz gewählt (Abbildung 1.2b). Dieser Berechnungspfad wird entgegengesetzt

des Pfeiles, welcher die Ausrichtung der Berechnungsmaschine angibt, abgefahren. Daher fährt die Berechnungsmaschine diesen Berechnungspfad rückwärts ab und verlegt dabei den auf der Haspel aufgewickelten Schlauch. In Abbildung 1.2c wird der Zeitpunkt dargestellt, nachdem der erste Berechnungspfad aus der Berechnungssequenz abgefahren wurde. Anhand der farblichen Markierung in Abbildung 1.2c ist zu erkennen, dass die Fläche entlang des soeben abgefahrenen Berechnungspfades einmal beregnet wurde. Als Nächstes folgt in Abbildung 1.2d die Wahl des zweiten Berechnungspfades. Dieser bildet den Rückweg bis vor der Gabelung in der Berechnungssequenz ab. Der Pfeil, welcher die Ausrichtung der Berechnungsmaschine angibt, zeigt in dieselbe Richtung, in der der Berechnungspfad abgefahren werden soll. Daher fährt die Berechnungsmaschine diesen Berechnungspfad vorwärts ab und wickelt dabei den verlegten Schlauch auf die Haspel auf. In der nächsten Abbildung 1.2e wurde der zweite Berechnungspfad abgefahren und die anliegende Fläche ein zweites Mal beregnet. Die Abbildung 1.2f zeigt die Wahl des dritten Berechnungspfades. Nachdem der dritte Berechnungspfad abgefahren wurde, ist die an der Berechnungssequenz anliegende Fläche, wie in Abbildung 1.2g dargestellt, mindestens einmal beregnet worden. Die Wahl des vierten und somit letzten Berechnungspfades wird in Abbildung 1.2h dargestellt. Nach der Abfahrt des letzten Berechnungspfades ist die Berechnungssequenz abgearbeitet und die anliegende Fläche wurde, wie in Abbildung 1.2i dargestellt, mindestens zweimal beregnet (Teilflächen wurden hier gegebenenfalls viermal beregnet).

Um mit einer Berechnungsmaschine der ersten Bauart automatisch einen Berechnungsplan einzuhalten, muss diese umgebaut sowie ergänzt werden. Eine grobe Lösungsidee hierzu wird in Abschnitt 1.3 beschrieben.

1.3 Initiale Lösungsidee

Unter Verwendung von geeigneten Sensoren können für die Automatisierung benötigte Informationen erfasst und bereitgestellt werden. Aus diesen Informationen können Steuerungssignale ermittelt und an geeignete Aktoren übertragen werden. Die Aktoren erlauben die Umsetzung der Steuerbefehle durch eine Berechnungsmaschine. Somit kann die automatisierte Berechnung durch ein informationstechnisches System erreicht werden. Im Gegensatz zu mechanischen Lösungen, wie zum Beispiel dem Verlegen von Schienen, müssen keine oder nur wenige Änderungen an der Fläche, welche beregnet werden soll, vorgenommen werden. Dies begünstigt den flexiblen Einsatz von Berechnungsmaschinen.

Das System muss in der Lage sein, einen Berechnungsplan einzuhalten. Hierzu muss dieses System neben der zeitlich gestarteten Berechnung, vor allem die automatische Abfahrt von Berechnungssequenzen, beherrschen. Dafür muss das System die Position sowie die Lage der Berechnungsmaschine bestimmen und diese ins Verhältnis zum aktuellen Berechnungspfad setzen. Den errechneten Abweichungen entsprechend muss das System Steuerbefehle, wie zum Beispiel Lenken oder Halten, bestimmen und umsetzen.

Damit ein Berechnungsplan vom System eingehalten werden kann, muss dieser dem System bekannt sein. Hierzu muss das System eine Benutzungsschnittstelle bereitstellen, damit

ein Betreiber einer automatischen Berechnungsmaschine einen Berechnungsplan hinterlegen kann. Die Benutzungsschnittstelle muss den Benutzer in die Lage versetzen aus bestehenden Dateien, welche Fahrspuren einer Fläche beschreiben, eine Berechnungssequenz zu erstellen. Zum Beispiel könnten hierzu Shape-Dateien (ein verbreitetes Dateiformat für Geodaten), wie in Quelle [3] definiert, verwendet werden. Nachdem eine Berechnungssequenz erstellt wurde, muss die Benutzungsschnittstelle es erlauben, diese mit einer Zeitplanung zu kombinieren. Durch die Kombination ist dann ein Berechnungsplan beschrieben.

Die Erfassung von Fahrspuren kann durch ein zusätzliches Gerät, welches seine Position erfasst und aufzeichnet, erreicht werden. Die Positionen, die das Gerät entlang einer Fahrspur aufzeichnet, können zu einer Fahrspur zusammengefasst werden. Alle für die Berechnung der jeweiligen Fläche notwendigen Fahrspuren können so einzeln aufgezeichnet und in einem geeigneten Format als Datei bereitgestellt werden.

Da die automatische Berechnungsmaschine bereits ihre Position bestimmt und diese auch aufzeichnen könnte, wäre es möglich diese auch zur Erfassung von Fahrspuren zu verwenden. Doch ein zusätzliches Gerät könnte wesentlich kleiner und leichter als eine Berechnungsmaschine ausfallen. Es kann getragen oder auf einem Fahrzeug befestigt werden, welches deutlich schneller und mobiler als eine Berechnungsmaschine ist. Daher eignet sich ein dediziertes Kartierungsgerät besser zur Erfassung von Fahrspuren.

1.4 Kriterien zur Einschätzung der Lösung

Zum Abschluss der Masterarbeit soll die entwickelte Lösung gegenüber den in diesem Abschnitt definierten Kriterien evaluiert werden. Gleichzeitig schränken die Kriterien den Lösungsraum ein und geben somit Richtlinien für eine Lösung vor.

Eine automatisierte Berechnungsmaschine sollte weniger Arbeitsaufwand verursachen als eine nicht automatisierte Berechnungsmaschine. Dies ist nur gegeben, wenn der Betrieb der automatisierten Berechnungsmaschine insgesamt weniger Interaktion mit dem Betreiber erfordert, als der Betrieb einer nicht-automatisierten Berechnungsmaschine. Das Optimum wäre erreicht, wenn die automatisierte Berechnungsmaschine nach einer initialen Positionierung auf der jeweiligen Fläche keine manuellen Eingriffe erfordert.

Während der Berechnung sollte eine automatisierte Berechnungsmaschine einer definierten Berechnungssequenz folgen. Große Abweichungen zwischen einem Berechnungspfad und der Berechnungsmaschine würden zum Verlassen der beschriebenen Fahrspuren führen. Dies wiederum könnte zu Beschädigungen an der Berechnungsmaschine selbst oder an schützenswerten Pflanzen führen. Somit ist es wichtig, dass während einer Berechnung der Abstand zwischen Berechnungspfad und Berechnungsmaschine möglichst gering gehalten wird. Das Optimum wäre erreicht, wenn keine Abweichungen zwischen Berechnungspfad und Berechnungsmaschine auftreten.

Die Automatisierung der Berechnungsmaschine soll ihre Einsatzmöglichkeiten nicht ein-

schränken. Eine Berechnungsmaschine ist sehr mobil. Dieser Vorteil gegenüber anderen Berechnungsverfahren soll nicht durch einen hohen Instrumentalisierungsaufwand einer zu berechnenden Fläche negiert werden. Die Automatisierung der Berechnungsmaschine soll sich außerdem nicht negativ auf ihre Wetterresistenz auswirken. Das Optimum wäre erreicht, wenn eine automatisierte Berechnungsmaschine die selben Einsatzmöglichkeiten wie eine herkömmliche Berechnungsmaschine bietet.

1.5 Aufbau des Dokuments

Im Kapitel 2 wird die Beschreibung des Projekts konkretisiert und allgemeine Anforderungen an die Lösung erhoben. Es folgt die Ermittlung von zentralen Anwendungsfällen des zu entwickelnden Systems in Kapitel 3. Anhand der Anwendungsfälle werden spezifische Anforderungen an die Lösung erhoben und strukturiert. Alle erhobenen Anforderungen in diesem Dokument besitzen eine eindeutige Identifikationsnummer. Die bis dahin vorliegenden Anforderungen werden genutzt, um eine Architektur des zu entwickelnden Systems in Kapitel 4 aufzustellen. In Kapitel 5 werden Designentscheidungen mit der Architektur und den Anforderungen begründet und die Umsetzung des Systems beschrieben. Die Tests der eingesetzten Komponenten und Funktionen werden in Kapitel 6 dokumentiert. In Kapitel 7 werden die Resultate der Arbeit zusammengefasst. Verbesserungspotenziale, Anregungen für zusätzliche Funktionen und die für diese geschaffenen Schnittstellen werden in Kapitel 8 beschrieben.

2 | Projektbeschreibung

In diesem Projekt geht es um die prototypische Realisierung einer automatisierten Beregnungsmaschine. Das Projekt wird von Marvin Banse, B.Sc. Informatik, und Florian Schmalriede, B.Sc. Informatik, im Rahmen ihrer Master-Abschlussarbeit durchgeführt. Die Abschlussarbeit wird von Herrn Prof. Dr.-Ing. Oliver Theel, Leiter der Abteilung Systemsoftware & verteilte Systeme der Carl von Ossietzky Universität Oldenburg, und Herrn Dipl.-Ing. Michael Pokriefke, Geschäftsführer der seed2soil GmbH & Co. KG, betreut.

Die seed2soil GmbH & Co. KG nimmt im Rahmen des Projekts die Stellung eines Partners ein. Das Unternehmen unterstützt die Abschlussarbeit mit Wissen über landwirtschaftliche Gerätschaften, der Umsetzung von mechanischen Umbauten und der Bereitstellung sowie Beschaffung von Geräten und Materialien.

Im Folgenden werden das Projekt und dessen Rahmenbedingungen genauer beschrieben. Dabei werden die allgemeinen Anforderungen an das zu entwickelnde System genannt. Jede Anforderung besitzt eine dokumentenweite eindeutige Nummer, welche durch einen Präfix den Adressaten nach Tabelle 2.1 kodiert.

Präfix	Adressat
PR	seed2soil GmbH & Co. KG.
CR	Marvin Banse und Florian Schmalriede
PCR	Kombination aus PR und CR
UC	Bediener

Tabelle 2.1: Anforderungsnummernpräfixe und deren Adressate

2.1 Systemumgebung

Beregnungsmaschinen werden unter anderem in der Landwirtschaft, in Baumschulen und in der Grünanlagenpflege eingesetzt. Sie beregnen Sonderkulturen, Pflanzenbestände oder auch Grünflächen. Aufgrund dessen gibt es verschiedenste Varianten von Beregnungsmaschinen, die individuell für das jeweilige Einsatzszenario gewählt werden. Die Varianten unterscheiden sich in Größe, Bauart und weiteren Attributen. Jedoch ist die zentrale Aufgabe aller Beregnungsmaschinen die Beregnung von sogenannten Beregnungsflächen. Eine Beregnungsfläche ist eine Fläche, welche beregnet werden soll. Beregnungsflächen liegen zumeist im Außenbereich. Wenn eine Beregnungsfläche im Innenbereich liegt, dann wurde bereits ein großer Installationsaufwand betrieben. Dies deutet daraufhin, dass die jeweilige Beregnungsfläche länger genutzt wird und ein Bewässerungssystem fest installiert ist. Um

den Regelfall abzudecken, muss eine automatisierte Beregnungsmaschine in der Lage sein, eine Beregnungsfläche im Außenbereich automatisch zu beregnen. Dabei muss die automatisierte Beregnungsmaschine resistent gegenüber den vorherrschenden Wetterbedingungen sein.

PCR1 Eine automatisierte Beregnungsmaschine muss Beregnungsflächen, welche im Außenbereich liegen, beregnen können.

PCR1.1 Eine automatisierte Beregnungsmaschine muss unter den selben Wetterbedingungen wie eine nicht automatisierte Beregnungsmaschine arbeiten können.

Der Beregnungsradius, der auf der Beregnungsmaschine montierten Beregnungsvorrichtung, ist begrenzt. Deshalb muss sich eine Beregnungsmaschine im Regelfall auf einer Beregnungsfläche bewegen, damit diese entsprechend beregnet wird. Die Beregnung einer Beregnungsfläche erfolgt jedoch, um schützenswerte Pflanzen mit Wasser zu versorgen. Daher ist bei der Bewegung einer Beregnungsmaschine darauf zu achten, dass schützenswerte Pflanzen möglichst nicht beschädigt werden. Hierzu werden ggfs. bereits vorhandene Fahrspuren auf der jeweiligen Beregnungsfläche erfasst. In diesen wachsen nur wenige bis keine schützenswerte Pflanzen. Die Fahrspuren werden auch für andere mobile Geräte wie zum Beispiel Traktoren benötigt und sind somit meist gegeben. Jedoch müssen sich die Fahrspuren, welche zur Beregnung durch eine Beregnungsmaschine genutzt werden sollen, für die jeweilige Beregnungsmaschine eignen. Neben der Breite der Fahrspur ist auch die Bodenfestigkeit ein wichtiges Kriterium hierzu. Wenn die Fahrspur zu unbefestigt ist, könnte die jeweilige Beregnungsmaschine steckenbleiben oder umkippen. Da schützenswerte Pflanzen und unbestimmte Bedingungen bezüglich Bodenfestigkeit außerhalb der Fahrspuren vorliegen, darf eine automatisierte Beregnungsmaschine nur zuvor definierte Fahrspuren zur Beregnung nutzen.

UR2 Auf einer Beregnungsfläche müssen für die Beregnung durch eine Beregnungsmaschine erforderliche Fahrspuren definiert sein.

CR3 Eine automatisierte Beregnungsmaschine darf sich nur auf zuvor definierten Fahrspuren selbstständig bewegen.

Für den Betrieb einer Beregnungsmaschine ist ein Wasseranschluss notwendig. Der Wasseranschluss muss ein für die jeweilige Beregnungsmaschine ausreichenden Wasserdruck aufweisen und innerhalb des Aktionsradius der Beregnungsmaschine liegen. Gegenüber anderen Bewässerungssystemen ist der zu erbringende Aufwand, bezüglich der Instrumentalisierung einer Beregnungsfläche, bei einer Beregnungsmaschine gering. Daher kann eine Beregnungsmaschine mit relativ geringem Aufwand auf unterschiedlichen Beregnungsflächen eingesetzt werden. Diese Flexibilität muss bei der Automatisierung einer Beregnungsmaschine erhalten bleiben.

UR4 Für den Betrieb einer automatisierten Beregnungsmaschine muss ein Wasseranschluss im Aktionsradius der Beregnungsmaschine vorhanden sein.

PCR5 Der Instrumentalisierungsaufwand einer Beregnungsfläche darf bei einer automatisierten Beregnungsmaschine nicht höher sein als bei einer nicht automatisierten Beregnungsmaschine.

Die Fahrspuren könnten durch Hindernisse wie Lebewesen oder Objekte blockiert sein. Jedoch wird die Berechnungsmaschine nicht mit einer Hindernisdetektion ausgestattet. Der Aufwand hierfür wird als hoch eingeschätzt, da die Berechnungsmaschine aufgrund ihrer sehr geringen Geschwindigkeit für Lebewesen nur eine sehr geringe Gefahr darstellt. Sie soll aber mit Notausschaltern ausgestattet werden, sodass sie bei Bedarf schnell gestoppt werden kann. Damit der Beginn einer Berechnung von einem Lebewesen erkannt werden kann, muss eine automatisierte Berechnungsmaschine die Berechnung mit einem Signal ankündigen. Aufgrund der fehlenden Hindernisdetektion ist vom Betreiber sicherzustellen, dass die Fahrspuren während einer Berechnung frei von Objekten ist.

PCR6 Eine automatisierte Berechnungsmaschine muss mit Notausschaltern ausgestattet sein.

PCR7 Eine automatisierte Berechnungsmaschine muss den Beginn einer Berechnung signalisieren.

UR8 Fahrspuren dürfen während einer Berechnung nicht durch Lebewesen oder Objekte blockiert sein.

2.2 Systembeschreibung

Die seed2soil GmbH & Co. KG gibt eine Berechnungsmaschine des Herstellers Fasterholt¹ vom Modell Minimobil 40 vor und stellt diese bereit. Sie soll als Basis für einen Prototypen einer automatisierten Berechnungsmaschine dienen.

PR9 Es muss eine Berechnungsmaschine vom Hersteller Fasterholt vom Modell Minimobil 40 gestellt werden.

PCR10 Als Grundlage der automatisierten Berechnungsmaschine soll eine Berechnungsmaschine vom Hersteller Fasterholt mit der Produktbezeichnung Minimobil 40 verwendet werden.

Das in Abbildung 2.1 dargestellte Minimobil 40 ist eine Berechnungsmaschine der zweiten Bauart. Somit eignet sie sich für die in Abschnitt 1.2 gezeigte Automatisierung.

Um einen Berechnungspfad mit dem Minimobil 40 zu berechnen, muss dieses zuerst an einen Wasseranschluss mit entsprechendem Wasserdruck angeschlossen werden. Hierzu wird das Schlauchende an der Vorderseite der Berechnungsmaschine mit dem Wasseranschluss verbunden. Nachdem der Schlauch mit dem Wasseranschluss verbunden wurde, muss der Riemen spanner gelöst werden, um die mechanische Verbindung zwischen Rotor und Haspel zu trennen. Dieser befindet sich an der Hinterseite des Minimobil 40. Mit gelöstem Riemen kann die Berechnungsmaschine per Hand oder unter Hilfenahme einer Zugmaschine bewegt werden. Auf dem Weg zur Zielposition wird der Schlauch von der Haspel abgewickelt. An der Zielposition angekommen, muss der Riemen wieder gespannt und der Wasserhahn, welcher sich ebenfalls an der Hinterseite befindet, geöffnet werden. Hierbei wird eine am

¹<https://www.fasterholt.dk>



Abbildung 2.1: FASTERHOLT - Minimobil 40 [4]

Wasserhahn angebrachte Feder gespannt und der Wasserhahn durch einen Metallbolzen in offener Position gesichert. Die Stoppposition der Beregnungsmaschine wird durch eine am Schlauch anzubringende Stoppvorrichtung, welche zusätzlich im Boden verankert wird, bestimmt. Anschließend kann der Wasseranschluss geöffnet werden. [5]

Das Wasser aus dem Wasseranschluss läuft durch den Schlauch der Beregnungsmaschine über die Wasserverteilung zur Beregnungsvorrichtung und dem Rotor. Der Rotor wird durch den Rückstoß des ausströmenden Wassers in Rotation versetzt. Die Rotation wiederum wird an die Antriebswelle übertragen und über den gespannten Riemen an die Haspel weitergegeben. Aufgrund der Rotation der Haspel wird der Schlauch auf dieser aufgewickelt. Dadurch zieht die Haspel sich zusammen mit der Beregnungsmaschine an den Wasseranschluss heran. Eine auf dem Schlauch laufende Deichsel ist mit der Lenkung an der Vorderseite des Minimobil 40 verbunden und sorgt so dafür, dass die Beregnungsmaschine dem Schlauch folgt. Außerdem ist vorne an der Deichsel eine Kippvorrichtung angebracht. Diese löst aus, sobald die Stoppvorrichtung am Schlauch erreicht wird. Die Auslösung hat wiederum zur Folge, dass der Metallbolzen, welcher den Wasserhahn der Beregnungsmaschine in offener Position hält, über einen Draht gezogen wird. Somit schließt sich der Wasserhahn an der Beregnungsmaschine und unterbricht damit die Wasserzufuhr zum Rotor und zur Beregnungsvorrichtung. In Folge stoppt das Minimobil 40 und beendet die Beregnung.

Die Beregnung eines Beregnungspfades erfolgt durch das Minimobil 40 halb automatisch. Der Bediener muss die Beregnungsmaschine positionieren und anschließend die Beregnung durch das Öffnen der Wasserzufuhr aktivieren. Die Beregnung entlang des Schlauches erfolgt dann automatisch. Um so eine Beregnungsfläche zu bedienen, muss das Minimobil 40 wiederholt neu positioniert und die Beregnung aktiviert werden. Wie bereits in Abschnitt 1.3 dargestellt, kann dies durch ein informationstechnisches System automatisiert erfolgen.

CR11 Das Minimobil 40 muss durch ein informationstechnisches System ergänzt werden.

Das Minimobil 40 kann sich eigenständig nur Vorwärts durch den Zug an einem Schlauch

bewegen. Für die automatisierte Einhaltung einer Beregnungssequenz muss sich eine Beregnungsmaschine vorwärts sowie rückwärts, wie in Abschnitt 1.2 begründet, bewegen können. Somit muss das Minimobil 40 um eine Komponente ergänzt werden, welche die Vorwärts- sowie die Rückwärtsfahrt erlaubt. Die Komponente muss außerdem eine Schnittstelle, geeignet für informationstechnische Systeme, zur Bestimmung der Fahrtrichtung bieten. Dies erlaubt die Steuerung der Fahrtrichtung durch informationstechnische Systeme.

PR12 Das Minimobil 40 muss durch eine Komponente, welche die Vorwärts- sowie die Rückwärtsfahrt erlaubt, ergänzt werden.

PR12.1 Die Komponente zur Fahrtrichtungsbestimmung muss eine Schnittstelle, geeignet für informationstechnische Systeme, zur Steuerung der Fahrtrichtung bieten.

Die Bewegung des Minimobil 40 erfolgt durch den am Rotor anliegenden Wasserdruck. Wenn kein Wasserdruck am Rotor anliegt, dann steht die Beregnungsmaschine still. Somit kann das Minimobil 40 sich nur während der Beregnung bewegen. Die Einhaltung eines Beregnungsplans erfordert, dass die Beregnung und die Bewegung zu entsprechenden Zeitpunkten aktiviert beziehungsweise deaktiviert werden kann. Am Minimobil 40 kann dies durch den Wasserhahn an der Rückseite erreicht werden. Der Wasserhahn öffnet oder sperrt die Wasserzufuhr zum Rotor und zur Beregnungsvorrichtung. Daher soll der Wasserhahn um eine für informationstechnische Systeme geeignete Schnittstelle ergänzt werden.

PR13 Der Wasserhahn des Minimobil 40 soll durch eine für informationstechnische Systeme geeignete Schnittstelle ergänzt werden.

Innerhalb eines Beregnungspfades können Knicke definiert sein, welche Lenkbewegungen während der Abfahrt erfordern. Das Minimobil 40 verfügt über eine Achsschenkelenkung. Diese wird bei der Vorwärtsfahrt durch eine auf dem Schlauch aufliegende Deichsel gesteuert. Damit kann die Beregnungsmaschine dem Schlauch auch in Kurven folgen. Bei einer Rückwärtsfahrt wäre, ohne zusätzlichen Instrumentalisierungsaufwand der zu beregnenden Fläche, keine Führung für die Deichsel vorhanden. Daher muss die Lenkung zumindest während der Rückwärtsfahrt durch ein informationstechnisches System gesteuert werden. Hierzu muss die Lenkung um eine Schnittstelle, geeignet für ein informationstechnisches System, ergänzt werden. Da zumindest während der Rückwärtsfahrt die Lenkung über ein informationstechnisches System erfolgen muss, kann dies auch bei der Vorwärtsfahrt übernommen werden. Ansonsten müsste das Minimobil 40 zusätzlich um eine Komponente zum An- und Entkoppeln der Deichsel ergänzt werden. Die An- und Entkoppelung der Deichsel wäre notwendig, da ansonsten der bereits verlegte Schlauch entsprechend der Lenkbewegungen auf der Rückwärtsfahrt verschoben werden würde. Der Schlauch ist zu diesem Zeitpunkt gespannt und mit Wasser gefüllt. Diesen zu verschieben würde einen hohen Kraftaufwand erfordern, den die Lenkung nicht aufbringen kann. Außerdem läge der verschobene Schlauch nicht mehr in der Spur für die später folgende Vorwärtsfahrt.

PR14 Die Lenkung des Minimobil 40 soll durch eine für informationstechnische Systeme geeignete Schnittstelle ergänzt werden.

UR15 Die Deichsel des Minimobil 40 darf während der Beregnung nicht auf dem Schlauch aufliegen.

Für den Betrieb eines informationstechnischen Systems ist eine Stromquelle notwendig. Diese muss ausreichend für die Aktoren, Sensoren und Steuereinheiten dimensioniert sein. Das Minimobil 40 besitzt keine Stromquelle und muss daher um eine ergänzt werden.

PR16 Das Minimobil 40 muss um eine entsprechend dimensionierte Stromquelle ergänzt werden.

Der Rotor, welcher das Minimobil 40 antreibt, wird durch ausströmendes Wasser in Rotation versetzt. Mit dem ausströmenden Wasser wird die Beregnungsmaschine selbst beregnet. Daher müssen alle zusätzlich angebrachten Komponenten und Umbauten gegenüber Wasser resistent sein. Um die elektronischen Komponenten zu schützen, muss das Minimobil 40 um einen wasserfreien Bereich ergänzt werden. Dies kann zum Beispiel durch eine geschlossene Box erreicht werden.

PCR17 Alle zusätzlich angebrachten Komponenten und Umbauten müssen gegenüber Wasser resistent sein.

PR18 Das Minimobil 40 muss um einen trockenen Bereich ergänzt werden.

Zusammenfassend soll das wie beschrieben ergänzte und umgebaute Minimobil 40 mit entsprechender Software in der Lage sein einen Beregnungsplan automatisch einzuhalten. Abgesehen von der initialen Positionierung und der Verbindung mit einem Wasseranschluss auf der Beregnungsfläche, sind im Normalfall keine weiteren Eingriffe durch den Bediener notwendig.

2.3 Vergleichbare Systeme

Die automatisierte Beregnungsmaschine soll eine geplante und spezifizierte Aufgabe erledigen. Sie muss der Aufgabe draußen, auf einer zur bearbeitenden Fläche im zweidimensionalen Raum nachgehen und darf dabei keine Hilfe eines Bedieners benötigen. Auf der zur bearbeitenden Fläche muss sie präzise einem Pfad folgen.

LIO Das Projekt Lawn In Order (LIO) stand nach [6] vor ähnlichen Problemen. LIO ist ein automatisches Rasenmähsystem (siehe Abbildung 2.2, welches nach einem vorgegebenen Zeitplan eine zuvor kartierte Rasenfläche mäht. Es unterteilt den Rasen in Teilstücke und definiert über diese einen langen, zusammenhängenden Pfad, der die gesamte Mähfläche abdeckt. Um den Rasen zu mähen, folgt es präzise diesem Pfad mit eingeschaltetem Mähwerk.

Dabei kennt es die globalen Koordinaten des Pfades und bestimmt die eigenen. LIO ist dauerhaft versucht den Abstand zum Pfad möglichst gering zu halten. Die Genauigkeit der bestimmten Koordinaten ist entscheidend dafür, wie genau LIO seinem Pfad folgen kann. Daher setzt es zur Lokalisierung ein C94-M8P-Modul nach [7] ein, welches eine Lokalisierungsgenauigkeit im Zentimeterbereich ermöglicht. Das Modul liefert die eigene Position aber nur mit einer Rate von 1 Hz. Um die eigene Position jederzeit abrufen zu können, setzt es zusätzlich auf eine Odometrie, welche die eigenen Radbewegungen misst und die

Änderungen auf die letzte, bekannte Position addiert.

Zusätzlich zur eigenen Position ist die Ausrichtung erforderlich, um einen Pfad zu folgen. Ohne die Ausrichtung kann das System nicht einschätzen, welche Fahrmanöver es dem Pfad näher bringen. LIO setzt dafür auf eine () in Kombination mit einem Magnetometer. Diese orientiert sich unter anderen an dem Erdmagnetfeld und der Erdbeschleunigung, um die eigene Ausrichtung relativ zum magnetischen Nordpol zu ermitteln.

Für die Automatisierung der Berechnungsmaschine sind die Mechanismen der Lokalisierung und der Pfadverfolgung interessant. Allerdings ist das System LIO deutlich wendiger als die Berechnungsmaschine und kann aufgrund von separat angetriebenen Rädern präzise gesteuert werden.

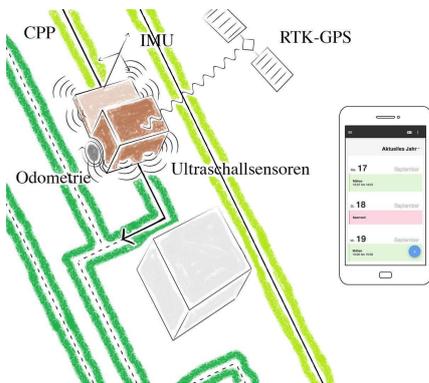


Abbildung 2.2: Übersicht über das Projekt LIO

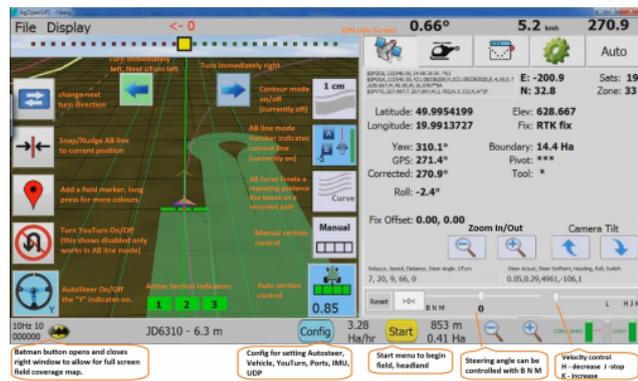


Abbildung 2.3: Bedienpanel der AgOpenGPS-Software

AgOpenGPS Das Lenkassistentensystem AgOpenGPS ist nach [8] für Parallelfahrten von Traktoren gedacht (siehe Abbildung 2.3). Wie auch LIO nutzt es ein ()-Modul zur Bestimmung der eigenen Position sowie einen Ausrichtungssensor um einer vorgegebenen Linie zu folgen. Ziel ist eine möglichst überdeckungsfreie Bearbeitung einer Fläche durch parallele Abfahrten. AgOpenGPS übernimmt ausschließlich die Lenkung. Es misst den Lenkwinkel mit einem Potentiometer am Lenkgestänge und nimmt mit einem am Lenkrad befestigten Motor Einfluss auf diesen. Die Lenkung eines Traktors ist jener der Berechnungsmaschine ähnlich. Das AgOpenGPS-Projekt ist experimentell und eignet sich noch nicht für den Einsatz auf einem realen Traktor. Aber die in der Software verwendeten Prinzipien können ggfs. auf Teilprobleme der automatisierten Berechnungsmaschine angewendet werden .

Selbstlenkende Traktoren werden in der selben Umgebung wie die automatisierte Berechnungsmaschine eingesetzt. Technologie, die sich für selbstlenkende Traktoren bewährt hat, ist auch auf die automatisierte Berechnungsmaschine interessant.

3 | Anwendungsfälle

Aus dem Interview mit der seed2soil GmbH & Co. KG sind fünf zentrale Anwendungsfälle (Abbildung 3.1) hervorgegangen. Die daraufhin formulierten Anforderungen motivieren sich anhand dieser Fälle.

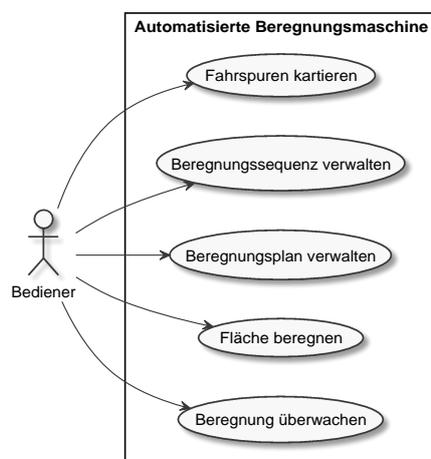


Abbildung 3.1: Anwendungsfalldiagramm der zentralen Anwendungsfälle

Damit eine automatisierte Beregnungsmaschine sich nur auf Fahrspuren eigenständig bewegt (Anforderung CR3), müssen diese zuvor vom Bediener definiert bzw. kartiert werden (Anforderung UR2). Aufbauend auf den kartierten Fahrspuren können Beregnungssequenzen erstellt werden. Der zeitliche Ablauf von Beregnungen wird im Beregnungsplan der Beregnungsmaschine festgelegt. Eine einzelne Beregnungsaufgabe gibt an, wann die Beregnungsmaschine eine Fläche beregnet. Während der Beregnung ist es für den Bediener hilfreich, sich über den Zustand der Beregnungsmaschine und den Fortschritt der Beregnung informieren zu können. So kann dieser Störungen erkennen und entsprechend eingreifen. Auch gibt es dem Bediener die Möglichkeit spätere Beregnungen besser planen zu können.

3.1 Fahrspuren kartieren

Die Fahrspuren auf einer Beregnungsfläche erlauben es, dass Fahrzeuge auf dieser agieren können. Auf einer landwirtschaftlichen Fläche werden diese zum Beispiel von Traktoren befahren. Daher sind Fahrspuren im Allgemeinen so beschaffen, dass diese eine ausreichende Festigkeit aufweisen und möglichst wenig schützenswerte Pflanzen bei der Überfahrt beschädigt werden. Somit eignen sich diese Fahrspuren auch für eine automatisierte Beregnungsmaschine.

Aufgrund der fortschreitenden Automatisierung von landwirtschaftlichen Geräten (siehe Abschnitt 2.3) liegen Fahrspuren gegebenenfalls bereits in digitaler Form vor. Ein hierfür häufig verwendetes Format, laut Anhang A, ist die Shape-Datei, definiert durch Quelle [3]. Mit Shape-Dateien lassen sich geometrische Objekte wie zum Beispiel Punkte, Polygonzüge und Polygone beschreiben. Zusätzlich können diese Objekte jeweils mit benutzerdefinierten Attributen versehen werden. Eine Shape-Datei besteht im Kern aus drei Dateien und umfasst eine Hauptdatei, eine Indexdatei und eine optionale Attributdatei. In der Hauptdatei, mit der Namensendung „.shp“, werden die geometrischen Objekte beschrieben. Innerhalb der Indexdatei, mit der Namensendung „.shx“, werden die Positionen der Datensätze, zu den geometrischen Objekten innerhalb der Hauptdatei, angegeben. Zusätzlich können die geometrischen Objekte mit der Attributdatei, in Form einer dBASE-Datei definiert nach Quelle [9], um Attribute ergänzt werden. Laut Definition sollten alle geometrischen Objekte in einer Hauptdatei der selben Art sein. Besonders Polygonzüge eignen sich zur Beschreibung von Fahrspuren, da sie eine geordnete Liste von Punkten beschreiben und somit der Definition 1 von Berechnungspfaden entsprechen. Die seed2soil GmbH & Co. KG konnte eine beispielhafte Shape-Datei (Abbildung 3.2), welche die Fahrspuren auf einem Feld in der Nähe des Wasserturms in Uelzen beschreibt, zur Verfügung stellen. Die Datei wurde von einem Landwirt mit Hilfe seines entsprechend ausgerüsteten Traktors erstellt. Zur Visualisierung der Shape-Datei wurde die Datei in ein () geladen und über eine Open Street Map-Ebene gelegt. Die einzelnen Fahrspuren werden in der Shape-Datei jeweils durch einen

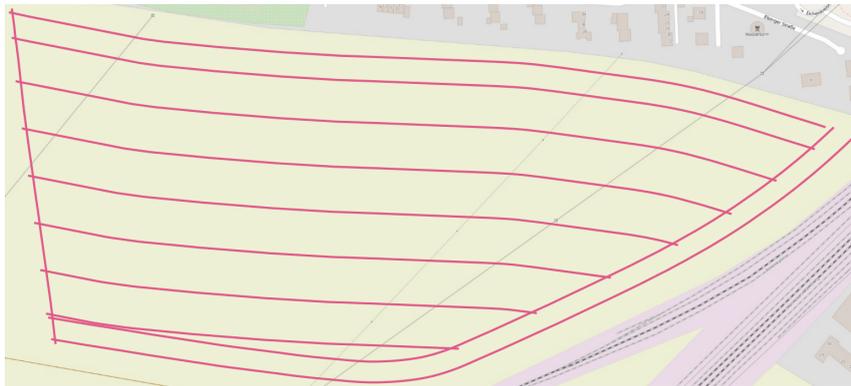


Abbildung 3.2: Durch Shape-Datei beschriebene Fahrspuren auf einem Feld in der Nähe des Wasserturms in Uelzen; die elf Fahrspuren sind violett dargestellt

Polygonzug beschrieben. Die X- und Y-Werte der Punkte im Polygonzug entsprechen Koordinaten aus dem () und beschreiben somit eine eindeutige Position auf der Erde. Das WGS84 ist ein weit verbreitetes geodätisches Referenzsystem und wird von vielen GIS unterstützt [10, 11, 12]. So können Koordinaten aus anderen Referenzsystemen leicht unter Verwendung eines entsprechenden GIS in das WGS84 übertragen werden. Viele Empfänger des () liefern außerdem ihre Koordinaten entsprechend dem WGS84 [13]. Aufgrund der weiten Unterstützung der beschriebenen Technologien, soll das zu entwickelnde System diese für die Beschreibung von Fahrspuren akzeptieren.

UR2.1 Fahrspuren sollen als Polygonzüge in einer Shape-Datei mit Koordinaten aus dem WGS84 beschrieben werden.

CR2.2 Fahrspuren, die durch Polygonzüge in einer Shape-Datei mit Koordinaten aus

dem WGS84 beschrieben sind, sollen vom System verarbeitet werden können.

Für den Fall, dass die Fahrspuren nicht in digitaler Form vorliegen oder die Beschreibung inkompatibel zum System ist, muss dem Bediener eine Möglichkeit zur Kartierung von Fahrspuren geboten werden. Eine manuelle Eingabe von WGS84-Koordinaten ist ohne Unterstützung eines dafür geeigneten Informationssystems sehr aufwendig und würde den eingesparten Arbeitsaufwand einer automatisierten Berechnungsmaschine bei weitem übertreffen. Die Verwendung eines Kartendienstes könnte den Aufwand der Kartierung erheblich senken. Der Bediener müsste lediglich entsprechende Punkte auf der Karte markieren und zu Polygonzüge zusammenfassen. Doch das Kartenmaterial könnte veraltet sein und Fahrspuren somit nicht korrekt dargestellt werden. Fahrspuren könnten außerdem zu klein sein oder durch andere Objekte verdeckt sein, sodass diese im Kartenmaterial gar nicht sichtbar sind. Für die Evaluierung der Genauigkeit, die das System bei der Berechnung einhält, ist es entscheidend die Abweichungen zwischen Berechnungsmaschine und Berechnungspfad zu messen. Bei einer auf einem Kartendienst basierten Lösung können die Abweichungen zwischen geplantem und tatsächlichem Berechnungspfad nur sehr schwer bestimmt werden. Somit sind auch die Abweichungen der Berechnungsmaschine zum Berechnungspfad nur sehr schwer zu bestimmen. Dies würde die Evaluation erheblich verkomplizieren. Daher sollen die Fahrspuren mit einem mobilen Gerät aufgezeichnet werden.

Selbstfahrende Traktoren sind bereits mit den notwendigen Lokalisierungscomponenten ausgestattet. Der Bediener fährt mit dem Traktor über die Fahrspuren und zeichnet dabei die Lokalisierungsdaten auf. Vor der Abfahrt einer Fahrspur aktiviert der Bediener die Aufzeichnung. Am Ende einer Fahrspur wird die Aufzeichnung wiederum beendet. So kann das System die einzelnen Fahrspuren unterscheiden und Polygonzüge erstellen. Wie die bereits vorgestellte Shape-Datei (Abbildung 3.2) zeigt, können diese anschließend exportiert werden. Die Kartierung von Fahrspuren soll nach diesem Muster erfolgen.

Auch die automatisierte Berechnungsmaschine muss mit entsprechenden Lokalisierungscomponenten versehen werden. Sonst könnte diese nicht ihre Position in Relation zum Berechnungspfad setzen und diesem somit nicht folgen. Allerdings lassen sich Berechnungsmaschinen nicht leicht rangieren. Die Größe sowie ein beschränkter Lenkwinkel erlauben nur eine eingeschränkte Bewegungsfreiheit. Somit eignet sich eine Berechnungsmaschine nicht so gut wie mobiles, dediziertes Kartierungsgerät für diese Aufgabe.

Ein dediziertes Kartierungsgerät kann die benötigte Flexibilität, Mobilität und Genauigkeit bieten. Gleichzeitig kann ein solches System kostengünstiger als ein Traktor mit entsprechenden Lokalisierungscomponenten hergestellt werden. Da nur die Lokalisierungscomponenten, sowie ein System für die Sammlung der Daten und der Bereitstellung einer Bedienoberfläche notwendig sind, kann das System sehr klein ausfallen. Dies erlaubt es dem Bediener mit dem Kartierungssystem die jeweiligen Fahrspuren abzulaufen oder mit einem Fahrzeug, an dem das System angebracht ist, abzufahren. Die Bedienoberfläche unterstützt dabei den Bediener und ermöglicht die Markierung von Anfang und Ende einer Fahrspur. Ähnlich zum selbstfahrenden Traktoren kann das System aus diesen Informationen entsprechende Polygonzüge erstellen. Auch der Export der Daten als Shape-Datei ist mit

einem solchen System möglich. Daher muss im Rahmen dieses Projektes ein eigens für die Kartierung von Fahrspuren vorgesehenes System mit dem beschriebenen Funktionsumfang entwickelt werden.

CR20 Es muss ein Kartierungssystem zur Kartierung von Fahrspuren entwickelt werden.

CR20.1 Das Kartierungssystem muss so klein und leicht sein, dass es von Hand getragen werden kann.

CR20.2 Das Kartierungssystem muss so flexibel sein, dass es von Hand getragen werden kann sowie an einem Fahrzeug befestigt werden kann.

CR20.3 Das Kartierungssystem muss den Export dieser gemäß Anforderung UR2.1 ermöglichen.

CR20.4 Das Kartierungssystem muss über eine Bedienoberfläche verfügen.

CR20.4.1 Die Bedienoberfläche des Kartierungssystems muss es dem Bediener ermöglichen, den Anfang sowie das Ende einer Fahrspur zu markieren.

CR20.4.2 Die Bedienoberfläche des Kartierungssystems muss es dem Bediener ermöglichen, die nach Anforderung UR2.1 exportierten Daten abzurufen.

PR20.5 Die benötigten Komponenten für die Entwicklung und den Bau des Kartierungssystems müssen gestellt werden.

Ein solches System zur Kartierung von Fahrspuren wäre nicht auf den Einsatz zur Bestimmung von Berechnungspfaden beschränkt. Auch andere automatisch agierende Systeme könnten die Informationen nutzen.

3.2 Berechnungssequenz verwalten

Die nach Abschnitt 3.1 kartierten Fahrspuren entsprechen der geforderten Form in Definition 1 und werden somit als Berechnungspfade betrachtet. Um eine Berechnungssequenz zu bilden müssen die Berechnungspfade außerdem, wie in Definition 3 formuliert, zusammenhängend nach Definition 2 sein. Die Berechnungspfade aus einer Shape-Datei sind nicht zwingend zusammenhängend, was daher noch erreicht werden muss. Der Einsatz einer Software, welche den Bediener hierbei unterstützt, kann Abhilfe schaffen. Diese ist in der Lage die exportierten Fahrspurdaten einzulesen und darzustellen. Anschließend können die Fahrspuren, algorithmisch oder manuell durch den Bediener, gegebenenfalls erweitert werden, sodass sich Schnittpunkte zwischen diesen ergeben. Die Schnittpunkte der Fahrspuren können als Knoten und die Fahrspuren selbst als Kanten eines Graphen betrachtet werden. Unter Berücksichtigung der möglichen Startpunkte kann aus diesem Graphen heraus wiederum eine Berechnungssequenz extrahiert werden. Auch dies kann algorithmisch oder manuell durch den Bediener erfolgen.

Da die Entwicklung einer Software zur Erstellung von Berechnungssequenzen viel Aufwand verursacht und die Arbeit zeitlich beschränkt ist, wird zugunsten des zentralen Ziels, die

Automatisierung einer Berechnungsmaschine, auf diese verzichtet. Um aber die Kompatibilität mit dieser sicherzustellen, soll das System eine Schnittstelle bieten, welche Berechnungssequenzen annimmt.

CR21 Das System muss eine Schnittstelle anbieten, über die Berechnungssequenzen hinterlegt werden können.

Um trotzdem die Eigenschaften einer Berechnungssequenz zu erreichen, wird eine Vorgehensweise bei der Kartierung der Fahrspuren vorgeschrieben. Der Bediener muss die Fahrspuren so kartieren, wie die Berechnungsmaschine diese nacheinander abfahren soll. Es wird dabei vorausgesetzt, dass die Berechnung mit einer Rückwärtsfahrt beginnt und sich die Fahrtrichtung immer zwischen zwei Berechnungspfaden ändert. Im Regelfall sollte somit die Berechnung mit einer Vorwärtsfahrt enden. Damit die aufgezeichneten Berechnungspfade zusammenhängend nach Definition 2 sind, muss die Endposition eines Berechnungspfades mit der Startposition des darauffolgenden Berechnungspfades übereinstimmen.

Diese Vorgehensweise setzt bisher nicht voraus, dass eine Berechnungssequenz an der selben Position endet, wo diese begonnen hat. Allerdings wird die Berechnungsmaschine typischerweise auf einer Berechnungsfläche platziert, wo sie nach ihrem Berechnungsplan eine einzige Berechnungssequenz immer wieder ausführt. Daher muss nach Definition 3 die Startposition des ersten Berechnungspfades mit der Endposition des letzten Berechnungspfades übereinstimmen. So wird garantiert, dass die Berechnungsmaschine eine Berechnungssequenz ausführen kann, nachdem sie dieselbe zuvor abgeschlossen hat, ohne dass ein Eingriff des Bedieners erforderlich ist. Bei Einhaltung dieser Vorschriften können die kartierten Fahrspuren, abgebildet als einzelne Polygonzüge in richtiger Reihenfolge, als Shape-Datei exportiert werden. Somit beschreibt diese eine Berechnungssequenz entsprechend der Definition 3.

UR2.3 Der Bediener muss Fahrspuren so kartieren, dass bei einem Export als Shape-Datei diese einer Berechnungssequenz nach Definition 3 entsprechen.

CR2.4 Das System muss eine Schnittstelle anbieten, über die Shape-Dateien, welche Berechnungssequenzen nach Definition 3 entsprechen, angenommen werden.

Durch die Erfüllung von Anforderung CR2.4, wäre auch die Anforderung CR21 erfüllt. Eine später entwickelte Software könnte ihre Ausgabe entsprechend der Anforderung CR2.4 formatieren und übergeben. Durch diese Lösung können in der Prototypenphase ohne zusätzlicher Software Berechnungssequenzen definiert und getestet werden. Gleichzeitig können die geschaffenen Schnittstellen später durch systemergänzende Komponenten genutzt werden.

3.3 Berechnungsplan verwalten

Einzelne Berechnungen von Berechnungssequenzen werden durch Berechnungsaufgaben terminiert. Diese wiederum werden in einem Berechnungsplan pro Berechnungsmaschine zusammengefasst. Damit hat das System die Informationen, wann und welche Berechnungssequenz

berechnet werden soll. Im optimalen Fall reicht somit die initiale Positionierung als einzige physische Interaktion aus, um mehrere Berechnungssequenzen, gegebenenfalls mehrfach, zu berechnen.

Ein Berechnungsplan kann mit einer gegebenen Aufgabenplanungssoftware realisiert werden. Eine solche erlaubt die zeit- oder ereignisgesteuerte Ausführung von Hintergrundaufgaben auf einem Computer. Da alle Informationen für eine automatisierte Berechnung im Planungsmoment feststehen, ist keine Interaktion mit dem Bediener während der Berechnung notwendig. Daher kann die automatisierte Berechnung als eine Hintergrundaufgabe durch eine gegebene Aufgabenplanungssoftware ausgeführt werden. Die Aufgabenplanungssoftware muss dem System lediglich die Berechnungssequenz übergeben. Damit dies erfolgen kann, muss das System eine Schnittstelle anbieten, über die eine Berechnung mit einer Berechnungssequenz gestartet werden kann.

CR22 Das System muss eine Schnittstelle anbieten, über die eine Berechnung mit einer Berechnungssequenz gestartet werden kann.

Wenn es zu einem Wetterumschwung kommt und somit eine gegebenenfalls begonnene Berechnung nicht mehr notwendig ist, sollte es möglich sein, diese zu pausieren oder abbrechen. Bei einer pausierten Berechnung sollte es außerdem möglich sein diese fortzusetzen.

CR23 Das System soll eine Schnittstelle anbieten, über die eine Berechnung einer Berechnungssequenz abgebrochen werden kann.

CR24 Das System soll eine Schnittstelle anbieten, über die eine Berechnung einer Berechnungssequenz pausiert werden kann.

CR25 Das System soll eine Schnittstelle anbieten, über die eine pausierte Berechnung einer Berechnungssequenz fortgesetzt werden kann.

Da die meisten Betriebssysteme bereits mit einer Aufgabenplanungssoftware ausgeliefert werden [14, 15, 16] oder entsprechende Software installiert werden kann, wird auf die Entwicklung einer Aufgabenplanungssoftware im Rahmen dieser Arbeit verzichtet. Eine später entwickelte, auf die Berechnung ausgerichtete Aufgabenplanungssoftware kann auf die Schnittstellen aus den Anforderungen CR22, CR23, CR24 und CR25 aufsetzen.

3.4 Fläche berechnen

Die Berechnung einer Fläche erfolgt durch die Berechnung entlang einer Berechnungssequenz. Aufgrund des begrenzten Wirkungsbereichs einer Berechnungsvorrichtung reicht eine Berechnung entlang eines Berechnungspfades im Allgemeinen nicht aus. Daher muss eine automatisierte Berechnungsmaschine eine Berechnungssequenz abfahren können.

CR26 Die automatisierte Berechnungsmaschine muss eine Berechnungssequenz abfahren können.

Aufgrund der Anforderung UR2.3 liegen dem System Berechnungssequenzen vor. Sie werden durch eine entsprechend der Abfahrreihenfolge geordnete Liste von Berechnungspfaden beschrieben. Dies erlaubt dem System die jeweilige Abfahrreihenfolge zu bestimmen. Die Berechnungspfade sind wiederum durch eine entsprechend der Abfahrreihenfolge geordnete Liste von Punkten, mit Koordinaten nach WGS84, global eindeutig repräsentiert. Somit kann das System seine eigene Pose in Relation zu den Koordinaten aus den Berechnungspfaden setzen und diese nacheinander abfahren. Um die eigene Pose und somit die Position sowie Ausrichtung zu bestimmen, muss sich die automatisierte Berechnungsmaschine lokalisieren. Die Lokalisierung muss zudem global erfolgen, damit die Position mit den vorgegebenen Koordinaten in Relation gesetzt werden kann.

CR27 Die automatisierte Berechnungsmaschine muss ihre Pose global bestimmen können.

Nach Anforderung UR2 sind die Fahrspuren, auf denen die Berechnungssequenzen liegen, entsprechend bodenfest und breit genug, sodass die jeweilige Berechnungsmaschine diese befahren kann. Da die Fahrspuren oft auch von größeren Geräten genutzt werden, ist die Breite im Regelfall mit 30 cm bis 40 cm je Spur, nach Anhang A, größer als von der Berechnungsmaschine benötigt wird. Aufgrund der undefinierten Gegebenheiten außerhalb der Fahrspuren darf sich die automatisierte Berechnungsmaschine, nach Anforderung CR3, nur auf diesen eigenständig bewegen. Um den Regelfall abzudecken, ergibt sich zusammen mit Anhang A, dass die Lokalisierungsgenauigkeit mindesten 15 cm betragen muss.

CR27.1 Die automatisierte Berechnungsmaschine muss sich auf mindestens 15 cm genau lokalisieren können.

Die Genauigkeit, mit der eine Berechnungssequenz abgefahren wird, bestimmt sich aus der Lokalisierungsgenauigkeit und der Abweichung zwischen Berechnungsmaschine und Berechnungssequenz. Die Abweichung lässt sich durch den Abstand zwischen der ermittelten Position und den vorgegebenen Koordinaten bestimmen. Da die ermittelte Position durch den Lokalisierungsfehler beeinflusst ist, muss dieser zusätzlich addiert werden. Um auf den Fahrspuren zu bleiben, muss die Ungleichung 3.1 erfüllt sein. Auch die Koordinaten aus der Berechnungssequenz könnten fehlerbehaftet sein. Allerdings wird an dieser Stelle davon ausgegangen, dass die Fehler während der Kartierung minimiert werden und somit nicht betrachtet werden müssen.

$$\text{Abweichung} + \text{Lokalisierungsfehler} \leq \frac{\text{Fahrspurbreite}}{2} \quad (3.1)$$

Wenn zu einem Zeitpunkt während der Abfahrt einer Berechnungssequenz die Ungleichung 3.1 nicht eingehalten werden kann, dann muss die automatisierte Berechnungsmaschine präventiv stoppen. Die Berechnungsmaschine befindet sich dann potentiell außerhalb einer Fahrspur und eine Weiterfahrt könnte dazu führen, dass die Berechnungsmaschine Schaden nimmt oder Objekte, wie zum Beispiel schützenswerte Pflanzen, beschädigt werden. Es ergibt sich ein Konflikt zwischen hoher Genauigkeit und einer schnellstmöglichen Berechnung. Hier entscheidet der Bediener, welche Genauigkeit bei der Abfahrt eines Berechnungspfades eingehalten werden muss. Mit einer Möglichkeit, notwendige Genauigkeiten zu definieren, können breite Fahrspuren mit einer höheren Toleranz als schmale Fahrspuren versehen

werden. Somit muss das System gegebenenfalls seltener stoppen und trotzdem werden sicherheitsrelevante Belange berücksichtigt.

CR28 Das System muss eine Möglichkeit bieten, eine einzuhaltende Genauigkeit für einen Berechnungspfad zu definieren.

UR28.1 Der Bediener muss eine einzuhaltende Genauigkeit für einen Berechnungspfad definieren.

CR29 Die automatisierte Berechnungsmaschine muss, sobald die vorgegebene Genauigkeit nicht mehr eingehalten werden kann, stoppen.

Trotz der definierten Gegebenheiten auf Fahrspuren kann es dazu kommen, dass die Fahrspuren keine hohe Bodenfestigkeit aufweisen. Eine Beeinflussung der Bodenfestigkeit, zum Beispiel durch einen Fuchsbau, ist nicht auszuschließen. Daher kann es zum Umkippen einer Berechnungsmaschine kommen. Um Folgeschäden durch das ausströmende Wasser an der Berechnungsmaschine, der Berechnungsfläche und schützenswerten Pflanzen zu vermeiden, muss das System ein Umkippen der Berechnungsmaschine erkennen und die Berechnung stoppen.

CR30 Das System muss ein Umkippen der Berechnungsmaschine detektieren.

CR31 Die Berechnungsmaschine muss stoppen, wenn sie umkippt.

Das System soll die Berechnungsmaschine entlang einer Berechnungssequenz steuern und hierzu die Schnittstellen, gefordert in den Anforderungen PR12.1, PR13 und PR14, nutzen. Über die Schnittstelle zum Wasserhahn kann nach Anforderung PR13 die Wasserzufuhr geöffnet oder geschlossen werden. Da sich das Minimobil 40 nur durch Wasserkraft bewegt, kann so die Fahrbewegung gesteuert werden. Zusammen mit der Schnittstelle zur Fahrtrichtungssteuerung kann die Berechnungsmaschine, nach Anforderung PR12.1, vorwärts oder rückwärts bewegt werden. Um auftretende Abweichungen zwischen Berechnungsmaschine und Berechnungssequenz über die Zeit zu reduzieren, kann die Schnittstelle, nach Anforderung PR14, zur Lenkung genutzt werden. Das System kann so die Lenkung stellen, um die Abweichung zu reduzieren.

CR32 Das System soll die Schnittstelle, nach Anforderung PR12.1, zur Steuerung der Fahrtrichtung verwenden.

CR33 Das System soll die Schnittstelle, nach Anforderung PR13, zur Steuerung der Wasserzufuhr verwenden.

CR34 Das System soll die Schnittstelle, nach Anforderung PR14, zur Steuerung der Lenkung verwenden.

Die seed2soil GmbH & Co. KG plant bei späteren Modellen des Prototyps den Antrieb um eine Geschwindigkeitsregelung zu ergänzen. Derzeitig kann das Minimobil 40 nur mit unregelmäßiger Geschwindigkeit fahren. Sobald die Wasserzufuhr geöffnet ist, wird die durch den Rotor erzeugte Kraft an die Räder übertragen. Somit nimmt das Minimobil 40, entsprechend der Bodenbeschaffenheit, der Steigung und des anliegenden Wasserdrucks, eine

relativ konstante Geschwindigkeit an. Von der seed2soil GmbH & Co. KG wurde während einer Testfahrt auf einer geraden und gepflasterten Fläche eine Geschwindigkeit von ca. $6 \frac{\text{cm}}{\text{min}}$ gemessen. Die Wassermenge, die bei der Beregnung verteilt wird, kann derzeit nur über wechselbare Düsen eingestellt werden. Mit einer Geschwindigkeitsregelung kann dies ergänzend, ohne Wechsel der Düsen, erfolgen. Wenn weniger Wasser auf dem jeweiligen Teilbereich aufgebracht werden muss, fährt die Beregnungsmaschine schneller und verbringt so weniger Zeit auf der Fläche. Wenn mehr Wasser aufgebracht werden muss, dann fährt die Beregnungsmaschine langsamer. Somit kann durch eine Geschwindigkeitsregelung eine gezielte Beregnung erreicht werden. Um eine solche Ergänzung zu unterstützen, soll das System die Vorgabe einer Geschwindigkeit auf einem Beregnungspfad erlauben und diese, wenn möglich, einhalten.

CR35 Das System soll eine Möglichkeit bieten, eine einzuhaltende Geschwindigkeit für einen Beregnungspfad zu definieren.

UR35.1 Der Bediener soll eine einzuhaltende Geschwindigkeit für einen Beregnungspfad definieren.

CR36 Die automatisierte Beregnungsmaschine soll, wenn möglich, vorgegebene Geschwindigkeiten einhalten.

Das Minimobil 40 wird durch den anliegenden Wasserdruck angetrieben, weshalb es sich nur während der Beregnung bewegen kann. Ein von der Wasserzufuhr unabhängiger Antrieb könnte die Beregnungsmaschine bewegen, ohne dass die Umgebung beregnet wird. Das System wäre in der Lage die Wasserzufuhr getrennt vom Antrieb zu steuern und somit nur dann zu beregnen, wenn dies auch erforderlich ist. Da auch eine alternative Antriebstechnologie für einen späteren Prototypen in Frage kommt, soll das System so entwickelt werden, dass dieses durch die Ansteuerung einer solchen erweitert werden kann.

CR37 Das System soll so entwickelt werden, dass die Ergänzung um eine Ansteuerung einer alternativen Antriebstechnologie möglich ist.

Durch die Erfüllung der Anforderungen CR26 und CR22 ist es möglich, dass das System automatisch eine Fläche beregnet. Eine Interaktion mit dem Bediener ist nur erforderlich, wenn während der Beregnung eine Störung auftritt.

3.5 Beregnung überwachen

Störungen müssen zum sofortigem Stopp der Beregnungsmaschine führen. Ein sicherer Betrieb kann während einer Störung nicht mehr sichergestellt werden und je nach Störung kann eine Interaktion erforderlich sein. Um genauere Informationen über eine Störung zu erhalten, muss das System eine Schnittstelle anbieten, über die Störungsinformationen angegeben werden.

CR38 Das System muss Störungen erkennen und identifizieren.

CR38.1 Das System muss Informationen zu Störungen über eine Schnittstelle bereitstellen.

Bei Störungen ist eine Interaktion mit der automatisierten Berechnungsmaschine notwendig. Da der Bediener in der Regel während einer Berechnung nicht in der Nähe der Berechnungsmaschine ist, muss dieser wiederholt anreisen, um die Berechnungsmaschine auf Störungen zu überprüfen. Durch die Anreise wird ein Aufwand erzeugt, welcher eigentlich durch den Einsatz einer automatisierten Berechnungsmaschine gesenkt werden soll. Daher ist es sinnvoll, dass der Bediener die Berechnungsmaschine aus der Ferne überwachen kann. Eine geeignete Bedienoberfläche kann dem Bediener über den Zustand des Systems informieren. Für diesen wird erkennbar, ob eine Berechnung ungewollt gestoppt werden musste und ein Eingriff notwendig ist. Die aktuelle Position der Berechnungsmaschine hilft dem Bediener dabei, diese auf großen Flächen zu lokalisieren.

Eine Bedienoberfläche kann den Fortschritt einer Berechnung anzeigen und somit das voraussichtliche Ende dieser angeben. So kann der Bediener eine anschließende Berechnung planen oder die Informationen sammeln, um den Berechnungsplan zu optimieren. Dies würde zu weniger Stillstand der Berechnungsmaschine führen und die Auslastung steigern.

In der Prototypenphase ist eine solche Bedienoberfläche allerdings nicht notwendig und wird daher nicht im Rahmen dieser Arbeit umgesetzt. Jedoch sind die genannten Informationen auch während der Entwicklungs- und Testphase hilfreich und müssen erhoben werden. Deshalb müssen entsprechende Schnittstellen zur Abfrage dieser Informationen in das System integriert werden.

Um die Genauigkeit bei der Einhaltung von Berechnungssequenzen beurteilen zu können, muss die aktuelle Pose, die Koordinaten des aktuellen Berechnungspfades, der Fortschritt innerhalb der aktuellen Berechnungssequenz, die berechnete Abweichung und der ermittelte Lokalisierungsfehler ausgewertet werden. Daher müssen auch diese Informationen vom System über eine Schnittstelle abrufbar sein.

CR39 Das System muss die aktuell ermittelte Pose über eine Schnittstelle bereitstellen.

CR40 Das System muss die Koordinaten des aktuellen Berechnungspfades über eine Schnittstelle bereitstellen.

CR41 Das System muss den Fortschritt innerhalb der aktuellen Berechnungssequenz über eine Schnittstelle bereitstellen.

CR42 Das System muss die ermittelte Abweichung über eine Schnittstelle bereitstellen.

CR43 Das System muss den ermittelten Lokalisierungsfehler über eine Schnittstelle bereitstellen.

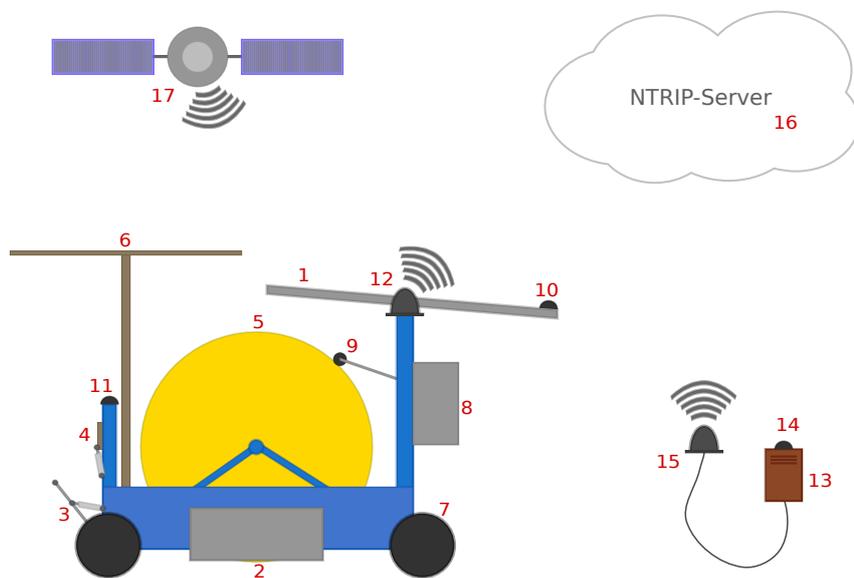
Für spätere Modelle des Prototypen hat die seed2soil GmbH & Co. KG den Wunsch geäußert, dass auch Wasserverbrauchswerte erfasst und protokolliert werden. Da Wasser oft kontingentiert ist, soll der Bediener so die Möglichkeit haben, den Verbrauch zu erfassen. Dies kann den Bediener bei der Planung von Berechnungen unterstützen und eine nachhaltige Berechnung fördern. Aufgrund der Kompatibilität zu einer späteren Lösung soll daher die Ergänzung des Systems um weitere Sensoren möglich sein.

CR44 Das System soll so entwickelt werden, dass die Ergänzung um weitere Sensoren möglich ist.

Eine spätere Bedienoberfläche kann die Schnittstellen verwenden und die daraus resultierenden Informationen aufbereitet anzeigen.

4 | Architektur

Abbildung 4.1 skizziert die prototypisch automatisierte Berechnungsmaschine und ein Kartiergerät, welche nach den Anforderungen entworfen wurden. Beide nutzen GNSSs und ein (-)System. Die Kommunikation mit dem NTRIP-System erfolgt über ein Mobilfunknetz.



- | | | |
|-----------------|-----------------------|---------------------|
| 1 Solarpanel | 7 Achsschenkellenkung | 13 Kartiergerät |
| 2 Batteriebox | 8 Gehäuse | 14 GNSS-Antenne |
| 3 Gangwahlhebel | 9 Drehwinkelgeber | 15 Mobilfunkantenne |
| 4 Wasserhahn | 10 GNSS-Antenne | 16 NTRIP-Server |
| 5 Haspel | 11 GNSS-Antenne | 17 GNSS-Satellit |
| 6 Rotor | 12 Mobilfunkantenne | |

Abbildung 4.1: Systemskizze

Die Berechnungsmaschine Das Minimobil 40 ist von der seed2soil GmbH & Co. KG. entsprechend der Anforderung PR16 um eine Spannungsquelle ergänzt worden. Diese besteht aus einem Solarpanel, einer Batterie und einem Laderegler. Das Solarpanel ist oberhalb einer Metallkonstruktion, welche von der seed2soil GmbH & Co. KG. an der Vorderseite der Berechnungsmaschine angebracht wurde, befestigt. Das ebenfalls an der Metallkonstruktion befestigte Gehäuse schützt den darin verbauten Laderegler vor Wasser und bietet Platz für weitere Komponenten. Mit dem Gehäuse wird die Anforderung PR18 erfüllt. Die Batterie ist in einer Batteriebox seitlich am Minimobil 40 untergebracht.

Die seed2soil GmbH & Co. KG. hat die Beregnungsmaschine mit einem Gangwahlhebel ausgestattet, mit dem sich die Fahrtrichtung, wie in PR12 gefordert, vorgeben lässt. Der Gangwahlhebel, die Wasserzufuhr und die Lenkung sind mit Linearantrieben versehen. Diese sind durch ein informationstechnisches System steuerbar, wodurch die Anforderungen PR12.1, PR13 und PR14 erfüllt sind.

Die elektrischen Komponenten des Informationssystems sind im Gehäuse verbaut und über Steckverbindungen mit den Aktoren und Sensoren verbunden. Somit werden die Linearantriebe vom System, gemäß den Anforderungen CR32, CR33 und CR34, bedient und erlauben diesem die Steuerung der Beregnungsmaschine.

Um die Anforderung CR27 zu erfüllen, muss sich die Beregnungsmaschine global lokalisieren können. Dies wird durch die Kombination einer absoluten Lokalisierung und einer relativen Lokalisierung erreicht. Die absolute Lokalisierung akkumuliert keinen Fehler, während sie die Position nur mit einer Frequenz von 1 Hz bestimmt. Dagegen bestimmt eine relative Lokalisierung die Position mit einer hohen Frequenz, akkumuliert dabei aber einen Fehler. Mit der Kombination von absoluter und relativer Lokalisierung werden die jeweiligen Schwächen ausgeglichen. Zwischen zwei Messungen der absoluten Lokalisierung wird auf die relative Lokalisierung zurückgegriffen, welche ihre Messung auf der letzten Messung der absoluten Lokalisierung aufbaut.

Die relative Lokalisierung erfolgt durch die Erfassung der von der Beregnungsmaschine zurückgelegten Strecke. Hierzu wird die Drehgeschwindigkeit der Haspel durch einen Drehwinkelgeber gemessen und die Stellung der Lenkung erfasst. Die Drehgeschwindigkeit der Haspel wird in die Fahrgeschwindigkeit der Beregnungsmaschine umgerechnet und mit der Lenkstellung kombiniert. Über die Zeit betrachtet kann so die zurückgelegte Strecke der Beregnungsmaschine berechnet werden. Doch die Messungen sowie das mathematische Modell zur Beschreibung der Fahrbewegung weisen Fehler auf. Diese werden durch die Berechnung über die Zeit akkumuliert.

Die absolute Lokalisierung erfolgt über das auf GNSS basierende ()-Verfahren. Die GNSS-Signale werden von zwei RTK-kompatiblen Empfängern mit jeweils einer GNSS-Antenne empfangen. Zusammen erlauben die Empfänger neben der globalen Position auch die globale Ausrichtung und somit die Pose der Beregnungsmaschine zu bestimmen.

Alle mechanischen Ergänzungen am Minimobil 40 sind wasserfest. Offenliegende Aktoren und Sensoren sind mindestens nach Schutzklasse IP65 zertifiziert und damit unempfindlich gegenüber Wasser und Staub. Durch das Gehäuse und die Batteriebox sind die übrigen Komponenten vor Wasser geschützt. Somit ist die Anforderung PCR17 eingehalten.

Das Kartiergerät Mit dem Kartiergerät können Fahrspuren erfasst werden, womit es die Anforderung CR20 erfüllt. Die Komponenten des Kartiergeräts sind in einem Holzgehäuse untergebracht. Mit den Abmessungen 15,3 cm × 8,9 cm × 6,9 cm und einem Gewicht von ca. 0,8 kg kann das Gerät von Hand getragen werden und entspricht somit der An-

forderung CR20.1. Im Gehäuse sind Anschlussbuchsen für die Spannungsversorgung, die GNSS-Antenne und für eine Mobilfunkantenne eingelassen. Eine 5 V-Spannungsquelle ist erforderlich und wird über ein ()-Kabel mit dem Kartiergerät verbunden. Eine externe GNSS-Antenne verbessert den Empfang und erlaubt so eine präzisere Lokalisierung. Eine zusätzliche Mobilfunkantenne ist optional und muss nur angeschlossen werden, wenn die integrierte Mobilfunkantenne keine ausreichende Empfangsqualität bietet. Durch das geringe Gewicht und die externen Antennen kann das Kartiergerät an einem Fahrzeug angebracht werden und entspricht daher der Anforderung CR20.2.

Zur Kartierung von Fahrspuren muss sich das Kartiergerät lokalisieren. Hierzu wird, wie bei der Berechnungsmaschine, das RTK-Verfahren eingesetzt. Entsprechend ist im Gehäuse des Kartiergeräts ein RTK-geeigneter Empfänger untergebracht.

Die GNSS & das NTRIP-System Die Berechnungsmaschine sowie das Kartiergerät lokalisieren sich über das RTK-Verfahren. Da dieses auf dem ()-Verfahren basiert, wird ein Korrektursignal benötigt. Der NTRIP-Standard ermöglicht den Empfang solcher als ()-Nachrichten über ein () basiertes Netzwerk. Hierzu sind die Berechnungsmaschine und das Kartiergerät mit einem Mobilfunknetz verbunden. Über dieses melden sich beide, entsprechend dem NTRIP-Standard, bei einem NTRIP-Server an und empfangen über diesen die Korrektursignale in Form von RTCM-Nachrichten.

4.1 Berechnungsmaschine

Die automatisierte Berechnungsmaschine ist ein komplexes System, welches aus vielen für sich stehenden Komponenten besteht, die über Schnittstellen miteinander kommunizieren. Dadurch wird eine Modularität erreicht, welche erlaubt, die automatisierte Berechnungsmaschine um Funktionen zu erweitern oder Komponenten auszutauschen, die sich nicht bewähren.

4.1.1 Hardware

Das in Abbildung 4.2 dargestellte Komponentendiagramm zeigt die Hardware, welche zur Automatisierung der Berechnungsmaschine eingesetzt wird. *PC* und *Controller* sind zentrale Komponenten, die das System orchestrieren und daher die Schnittstellen der anderen Komponenten bedienen.

Controller Der *Controller* bindet andere Komponenten über Schnittstellen, wie einzelne Pins oder Bussysteme, an. Seine Aufgabe liegt darin, die Hardware der Berechnungsmaschine vor dem *PC* zu maskieren. Er kommuniziert mit dem *PC* über eine ()-Schnittstelle, nimmt Befehle von diesem entgegen und sendet von Sensoren gemessene Werte.

Die Befehle sind Steuerbefehle, nach denen der *Controller* die Wasserversorgung, die Lenkung oder das Getriebe stellt. Auf diese kann er durch das Setzen von dedizierten, digitalen Pins Einfluss nehmen. Die Signale auf diesen Pins werden von Motor-Treibern interpretiert, welche dann eine den Signalen entsprechende Versorgungsspannung an den jeweiligen

Motor anlegen.

Neben den Steuerbefehlen nimmt der *Controller* Signalbefehle entgegen. Diesen entsprechend zeigt er einen Systemstatus mittels optischen oder akustischen Signalen an. Die Signalgeber steuert er über das Setzen von dedizierten, digitalen Pins.

Der *Controller* liest Daten über den Zustand der Berechnungsmaschine aus Sensoren aus und bereitet diese auf. Er wertet die Daten er in erster Instanz aus und ergreift gegebenenfalls Maßnahmen zum Schutz der Berechnungsmaschine. Alle gemessenen Daten teilt er dem *PC* über eine *UART*-Schnittstelle mit.

Die Sensoren können nach dem Push- oder dem Pull-Prinzip arbeiten. Ein *IMU* arbeitet nach dem Pull-Prinzip, ist über eine *I2C*-Schnittstelle angebunden und liefert Werte über die Rotationsgeschwindigkeit und die Neigung der Berechnungsmaschine. Der *Controller* muss diese Werte zyklisch abfragen, was auch für die Sensoren in den Antrieben der Lenkung und des Getriebes gilt. Diese sind über jeweils einen Pin angeschlossen, über den die Position des Aktors ausgelesen wird.

Ein Drehgeber ist an der Haspel befestigt, arbeitet nach dem Push-Prinzip und misst, mit welcher Geschwindigkeit der Schlauch ab- oder aufgewickelt wird. Der *Controller* wird mittels eines *Interrupts* über eine Änderung des Sensorwertes benachrichtigt, woraufhin er den Sensorwert ausliest. Ein Bedienerpanel arbeitet nach dem selben Prinzip. Jede Eingabe eines Bedieners führt zum *Interrupt*, der *Controller* reagiert entsprechend der Eingabe und teilt diese dem *PC* mit. Mögliche Eingaben sind ein Notaus und ein Ein- bzw. Ausschaltsignal.

PC Der *PC* verfügt über deutlich mehr Rechenleistung als der *Controller*. Er berechnet die Steuerbefehle für den *Controller* abhängig von der eigenen Position und dem zu folgenden Pfad. Um seine Position zu bestimmen, greift er primär auf zwei *RTK Controller* zurück. Diese sind an jeweils eine *USB*-Schnittstelle angebunden und mit jeweils einer Antenne verbunden, um die zur Lokalisierung notwendigen Signale zu empfangen.

Neben den Signalen benötigen die *RTK Controller* Korrekturdaten, welche aus dem Internet abgerufen werden. Den Zugriff auf das Internet ermöglicht ein über eine *USB*-Schnittstelle angeschlossenes () *Modem*, in welchem sich eine SIM-Karte befindet, mit der es auf das Mobilfunknetz zugreifen darf. Da eine Berechnungsmaschine auch in entlegenen Gegenden mit schlechtem Mobilfunkempfang eingesetzt wird, ist das *LTE Modem* mit einer Antenne verbunden, welche die Empfangsleistung verstärkt.

Die Berechnungsmaschine wird in der Prototypenphase durch mindestens einen direkt verbundenen Computer überwacht. Eine solche Verbindung wird durch einen *WLAN Access Point* ermöglicht, welcher über eine interne *UART* Verbindung an den *PC* angebunden ist. So können Einstellungen der Berechnungsmaschine komfortabel verändert und erprobt werden.

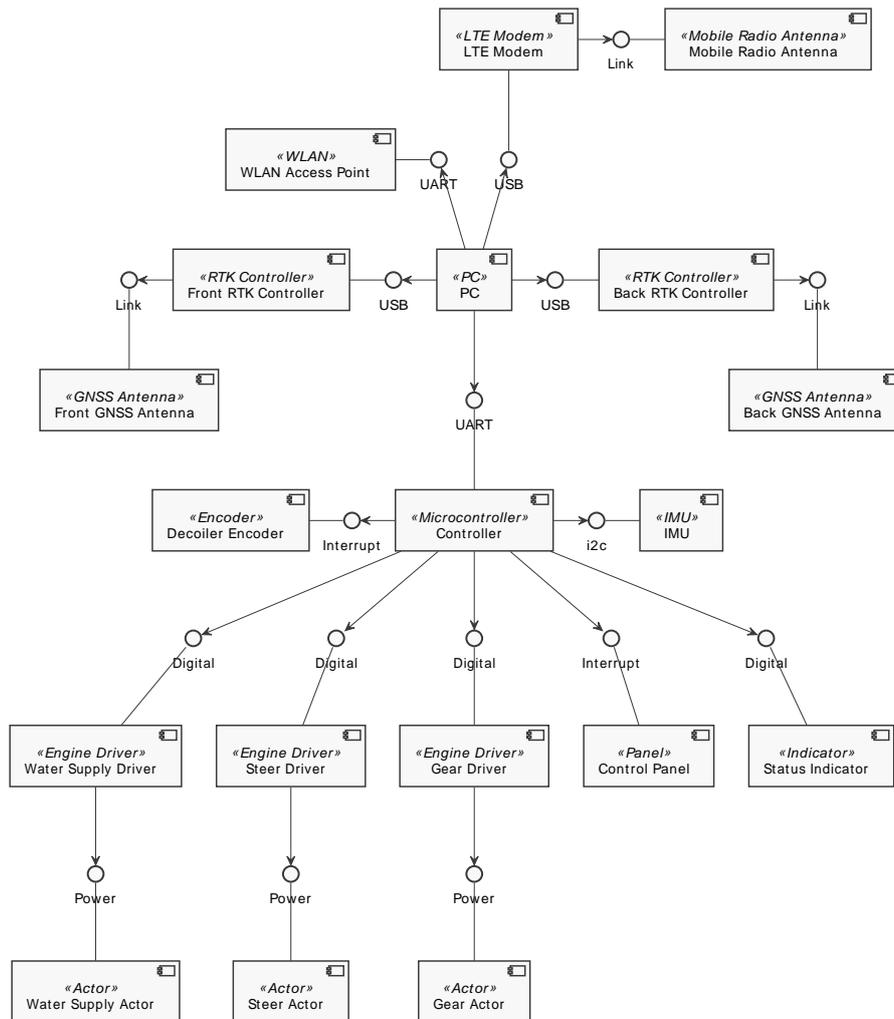


Abbildung 4.2: Komponentendiagramm über die Hardware zur Automatisierung der Beregnungsmaschine

4.1.2 Software

Die Software auf der Beregnungsmaschine ist in einzelne Komponenten unterteilt, welche jeweils dedizierte Aufgaben übernehmen und in Abbildung 4.3 dargestellt werden. Die Beregnung wird von der *Sprinkling*-Komponente gesteuert. Sie startet die Beregnung, lädt dafür die Beregnungssequenz aus dem *Sequence Repository*, extrahiert den aktuellen Beregnungspfad aus der Beregnungssequenz und sendet diesen an die *Path Following*-Komponente.

Die *Path Following*-Komponente fährt den zugewiesenen Pfad ab. Dafür ermittelt sie die Pose der Beregnungsmaschine mithilfe der *Localization*-Komponente, berechnet die Abweichung gegenüber dem Beregnungspfad, bestimmt ein geeignetes Manöver und teilt dieses dem *Hardware Interface* als *Drive Command* mit.

Die *Localization*-Komponente verwendet Sensordaten aus verschiedenen Quellen, um eine Pose zu errechnen. Dafür greift sie auf zwei Geo-Datensätze zu, die von der *Hardware Interface*-Komponente bereitgestellt werden und kombiniert diese. Zusätzlich sammelt die *Hardware Interface*-Komponente Odometriedaten, welche ebenfalls zur Lokalisierung ver-

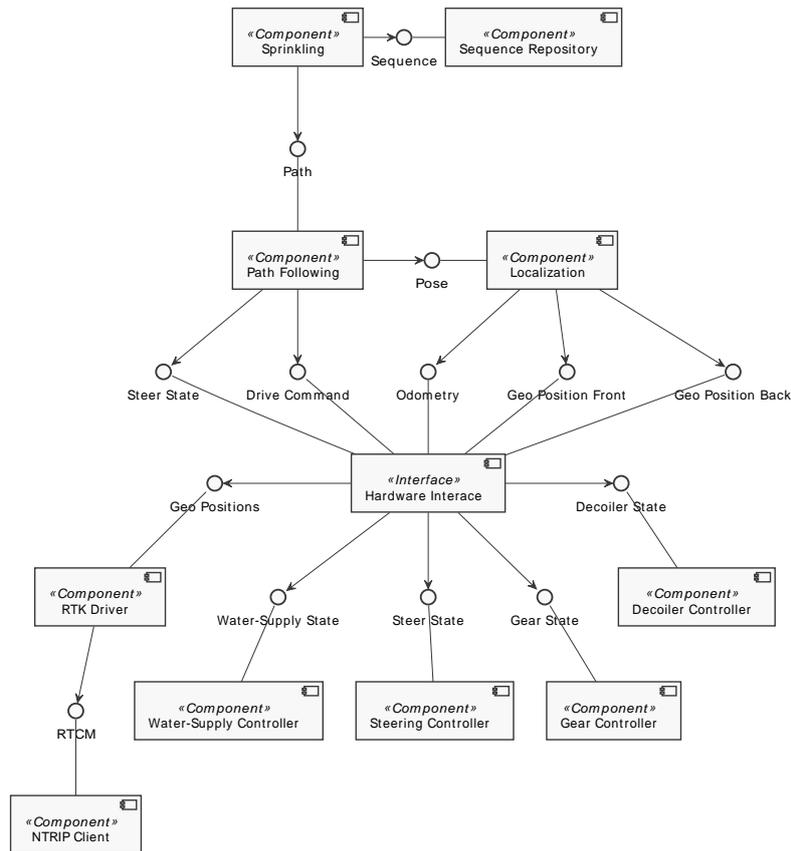


Abbildung 4.3: Komponentendiagramm über die Software zur Automatisierung der Beregnungsmaschine

wendet werden.

Die *Hardware Interface*-Komponente maskiert die Hardware. Der Zugriff auf Sensoren oder Aktoren geschieht über diese. Sie wandelt von der *Path Following*-Komponente gesendete Manöver in Anweisungen für die am Manöver beteiligten Komponenten um und stellt von Sensoren abgefragte Daten bereit.

Der *RTK-Driver* ermittelt die Positionen an der Front und am Heck der Beregnungsmaschine. Er steuert die Sensoren an, fragt Daten ab und verbessert deren Messungen, indem er Korrekturdaten in die Sensoren speist. Dafür verwendet er *RTCM*-Korrekturdaten, die von einem *NTRIP-Client* bereitgestellt werden.

Die *Controller*-Komponenten übernehmen die korrekte Ansteuerung und Abfrage je einer Hardwarekomponente. Der *Steering Controller* setzt und liest die Lenkungsstellung, wie der *Gear Controller* die Getriebestellung. Der *Water Supply Controller* öffnet und schließt die Wasserversorgung. Dabei hält er diese im Fehlerfall geschlossen, um die Beregnungsmaschine zu schützen. Der *Decoiler Controller* liest den Sensor aus, welcher die Haspelbewegungen misst und berechnet aus diesen Daten eine Geschwindigkeit.

4.2 Kartiergerät

Das Kartierungsgerät besteht aus verschiedenen Hard- und Softwarekomponenten, die kombiniert ermöglichen, Fahrspuren zu kartieren.

4.2.1 Hardware

Das in Abbildung 4.4 dargestellte Komponentendiagramm zeigt die Hardwarekomponenten des Kartierungsgeräts und deren Verbindungen untereinander. Kern des Kartierungsgeräts bildet der *PC*, welcher Schnittstellen zur Anbindung weiterer Komponenten bietet. Aufgabe des Kartierungsgeräts ist die Kartierung von Fahrspuren, während es über diese bewegt wird. Dafür muss es in der Lage sein, die eigene Position zu bestimmen. Für diesen Zweck verfügt es über einen *RTK Controller*, welcher über eine *USB*-Schnittstelle angebunden ist. Der *RTK Controller* ist mit einer Antenne verbunden, über die die zur Lokalisierung notwendigen Signale empfangen werden.

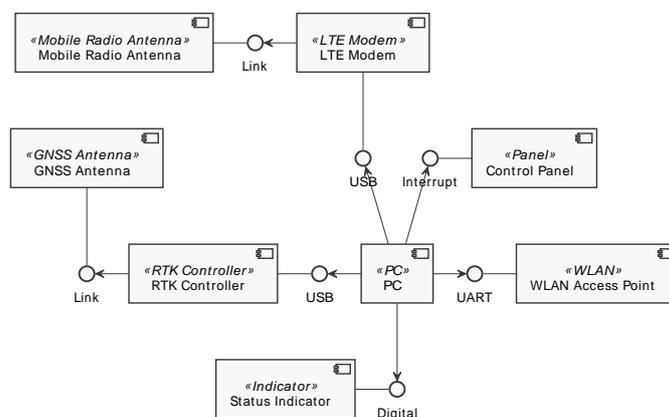


Abbildung 4.4: Komponentendiagramm über die Hardware des Kartierungsgeräts

Neben den Signalen benötigt der *RTK Controller* Korrekturdaten, um sich präzise zu lokalisieren. Diese werden aus dem Internet abgerufen. Der *PC* benötigt dafür eine Internetverbindung, die ein *LTE Modem* bereitstellt. Dieses ist über eine *USB*-Schnittstelle angebunden und beinhaltet eine SIM-Karte, mit der es auf das Mobilfunknetz zugreifen darf. Da das Kartierungsgerät auch in entlegenen Gegenden mit schlechtem Mobilfunkempfang eingesetzt wird, ist das *LTE Modem* mit einer Antenne verbunden, welche die Empfangsleistung verstärkt.

Das Kartierungsgerät stellt eine Bedienoberfläche bereit, die mit einem Smartphone oder einem Tablet aufgerufen werden kann. Dafür müssen diese Geräte sich mit dem Kartierungsgerät verbinden, was durch einen *WLAN Access Point* ermöglicht wird. Dieser ist über eine interne *UART*-Schnittstelle mit dem *PC* verbunden. Über das WLAN verbundene Geräte können über den *PC* auf dessen Internetverbindung zugreifen.

Ein Lampe ist an einem Pin des *PCs* angeschlossen und zeigt, wenn das Kartierungsgerät einsatzbereit ist. Ein Schalter löst bei Betätigung einen *Interrupt* aus, worauf sich das Kartierungsgerät herunterfährt und abschaltet.

4.2.2 Software

In Abbildung 4.5 sind die Softwarekomponenten des Kartierungsgeräts dargestellt.

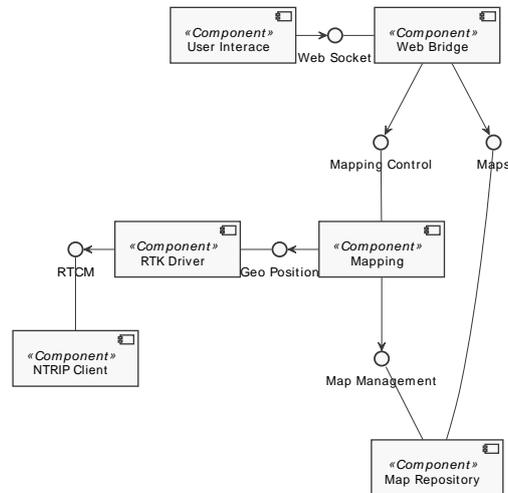


Abbildung 4.5: Komponentendiagramm über die Software des Kartierungsgeräts

Das Kartierungsgerät wird mit einem Smartphone oder Tablet-Computer bedient. Diese stellen die *User Interface*-Komponente in Form einer Webseite dar. Sie zeigt die Karten an, stellt sie für den Download zur Verfügung und nimmt Befehle zur Erstellung neuer Karten entgegen. Dafür greift das *User Interface* mittels des *Web Socket*-Protokolls auf die *Web Bridge*-Komponente zu. Diese leitet Anfragen an die *Mapping*-Komponente und die *Map Repository*-Komponente weiter.

Die *Mapping*-Komponente führt Kartierungen durch. Auf Kommando erstellt sie mit Beginn einer Kartierung einen ersten Pfad. Der Pfad beginnt mit einer Position, welche der derzeitigen Position des Kartierungsgeräts entspricht. Diese muss die *Mapping*-Komponente ermitteln, wofür sie auf den *RTK Driver* zugreift. Zur Bestimmung der Position steuert der *RTK Driver* einen Sensor an und liest die Position aus diesem aus. Um dessen Messverfahren zu verbessern, speist er *RTCM*-Korrekturdaten aus dem *NTRIP Client* in den Sensor. Der *NTRIP Client* fragt die Korrekturdaten aus dem Internet ab.

Dieser Vorgang wird wiederholt, bis der Bediener den Pfad als eingemessen markiert. Die *Mapping*-Komponente kann beliebig viele Berechnungspfade kartieren. Alle kartierten Pfade werden mit Abschluss der Kartierung kombiniert und als Shape-Datei durch die *Map Repository*-Komponente gespeichert.

5 | Realisierung

Im Folgenden wird beschrieben, wie die Berechnungsmaschine automatisiert wurde. Aus den Anwendungsfällen (siehe Kapitel 3) ist eine Reihe an erforderlichen Funktionen hervorgegangen, welche die Struktur dieses Kapitels vorgeben.

5.1 Grundlagen

In diesem Abschnitt wird die Basis der später beschriebenen Funktionen gelegt. Sie ist funktionsübergreifend und erleichtert die Entwicklung und Bedienung der automatisierten Berechnungsmaschine.

5.1.1 Robot Operating System

Das () ist eine Sammlung an Softwarebibliotheken und Konventionen, () genannt, welche die Entwicklung von Robotersystemen erleichtern. Dabei bietet es ein Publisher-/Subscriber-Messaging-System, was die Kommunikation von verschiedenen Komponenten untereinander über Hardware hinweg vereinfacht. Zusätzlich bietet es Werkzeuge, die zur Visualisierung und Analyse von Robotersystemen geschaffen wurden und eine gezielte Fehlersuche ermöglichen.

Das ROS steht der ARM-Prozessorarchitektur innerhalb der ubuntu-Paketquellen zur Verfügung, weshalb es auf einem Einplatinencomputer vom Typ RaspberryPi 3 Model B+ installiert wird.

5.1.2 Mathematisches Modell

Eine vollständige, mathematische Beschreibung der Berechnungsmaschine wäre zu komplex und übersteigt die Anforderungen bei weitem. Daher werden Berechnungen auf einem vereinfachten, mathematischen Modell der Berechnungsmaschine durchgeführt.

Die Berechnungsmaschine kann sich nicht frei in einem zweidimensionalen Koordinatensystem, ihrem Arbeitsraum, bewegen. Wenn sie beispielsweise in Richtung Osten fahren soll, dann muss sie sich entsprechend ausrichten, sofern sie nicht bereits nach Osten ausgerichtet ist. Damit sind die effektiven Freiheitsgrade der Berechnungsmaschine, die Richtungen in die sie sich bewegen kann, kleiner als die aktiven Freiheitsgrade, die Dimension ihres Arbeitsraums. Solche Systeme, die nicht zwischen allen benachbarten Positionen in einem Arbeitsraum wechseln können, werden als nicht-holonom bezeichnet. Aufgrund ihrer aus dem Kfz-Bereich bekannten Achsschenkellenkung lässt sich die Berechnungsmaschine

ähnlich klassifizieren. Die Art der Lenkung schränkt die Bewegungsmöglichkeiten der Beregnungsmaschine zusätzlich ein und erlaubt nur eine Ausrichtungsänderung zusammen mit einer Positionsänderung.

Ackermann Modell Das Ackermann Modell beschreibt nicht-holonome, Kfz-ähnliche Systeme und wird als Grundlage für ein mathematisches Modell der Beregnungsmaschine verwendet. Vergleichbare Anwendungen des Modells sind in [17, 18, 19] gegeben. In Abbildung 5.1 ist ein entsprechendes System, einschließlich der Geometrie des Modells, dargestellt. Das Modell setzt voraus, dass die Hinterachse starr ist und die Lenkung an der Vorderachse angebracht ist. Weiter wird vorausgesetzt, dass bei einer Lenkung beide Vorderräder einen gemeinsamen Drehpunkt Z auf der Hinterachsenlinie bilden. Die Lenkwinkel ϕ_L und ϕ_R sind somit nur gleich, wenn beide 0rad betragen, was einer Geradeausfahrt entspricht. Der Parameter L beschreibt den Abstand zwischen der Vorderachse und der Hinterachse. Zusammen mit dem Lenkwinkel ϕ_L ist die Bestimmung von r_L nach Gleichung 5.1 möglich.

$$r_L = \frac{L}{\tan \phi_L} \quad (5.1)$$

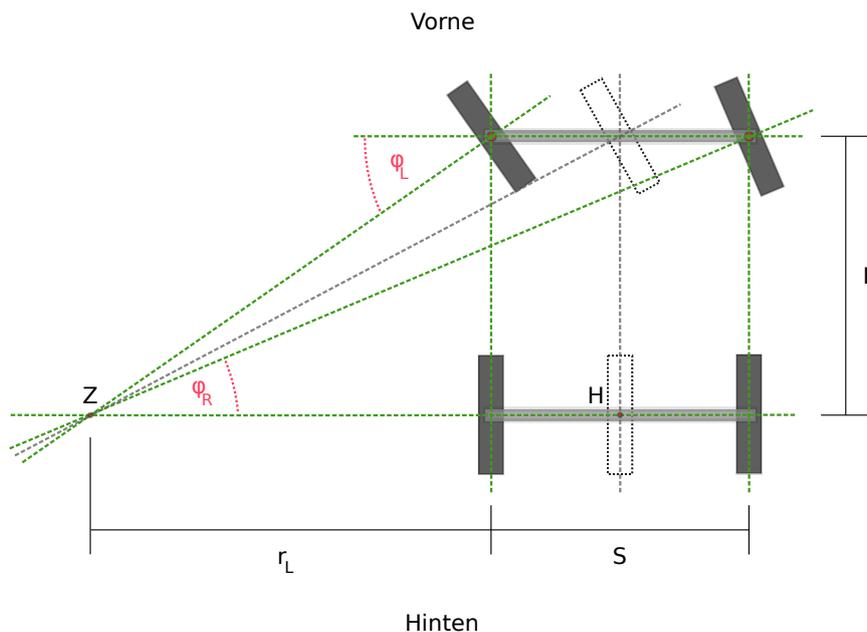


Abbildung 5.1: Die Geometrie des Ackermann Modells

Die Lineargeschwindigkeit in Vorwärtsrichtung v bezieht sich auf das in der Breite gesehene Zentrum. Bei einer Kurvenfahrt würde in Abhängigkeit der Kurvenrichtung die Geschwindigkeit auf der linken Seite bzw. rechten Seite der Beregnungsmaschine größer oder kleiner als im Zentrum ausfallen. Daher wird beim Ackermann Modell eine weitere Vereinfachung zu einem zweirädrigen System mit einer gelenkten Achse durchgeführt. Die linken sowie die rechten Räder werden in der Mitte der jeweiligen Achse vereint. In der Abbildung 5.1 sind die Konturen dieser Räder gestrichelt dargestellt. Der Mittelpunkt H , des zentrierten Hinterrades, wird als körperfester Koordinatenursprung für die Berechnung von Geschwindigkeiten genutzt. Der Radius r , mit dem sich der Koordinatenursprung um

den Drehpunkt Z bewegt, kann mit der Gleichung 5.2 berechnet werden.

$$r = r_L + \frac{S}{2} \quad (5.2)$$

In Kombination mit der Lineargeschwindigkeit v , welche nach vorne gerichtet und passend zum Mittelpunkt H ist, kann die Winkelgeschwindigkeit $\dot{\theta}$ mit der Formel 5.3, nach [19], berechnet werden.

$$\dot{\theta} = \frac{v}{r} \quad (5.3)$$

Lenkwinkelbestimmung Um den Wert von ϕ_L zu bestimmen, muss die Lenkung genauer betrachtet werden. Hierzu wurde diese mit ihren geometrischen Eigenschaften in Abbildung 5.2 skizziert. Die Räder der Achsschenkellenkung sind über ein Lenktrapez miteinander verbunden. Somit sind die Lenkwinkel ϕ_L und ϕ_R miteinander gekoppelt und bilden immer einen gemeinsamen Drehpunkt. Die Ansteuerung der Lenkung und somit die Anpassung von ϕ_L erfolgt über einen Linearantrieb. Seine Befestigungspunkte und das Gelenk des linken Vorderrades bilden ein Dreieck, bei dem die Seiten a und b eine konstante Länge aufweisen. Die Länge der Seite c entspricht der Länge des Linearantriebs inklusive dessen Kolben und kann somit verändert werden. Durch die Änderung der Länge c wird auch der Winkel γ gemäß der Gleichung 5.4 geändert.

$$\gamma = \arctan\left(\frac{a^2 + b^2 - c^2}{2 * a * b}\right) \quad (5.4)$$

Das Dreieck zwischen den Befestigungspunkten des Linearantriebs und dem Gelenk des linken Vorderrades gleicht nicht dem konstruierbaren Dreieck zwischen Lenkachse und Vorderachse. Daher müssen die festen Parameter ω_a und ω_b beachtet werden. Entsprechend der Zeichnung beschreiben diese den Winkel zwischen Lenkachse und der Seite a sowie den Winkel zwischen der Vorderachse und der Seite b . Zusammen bilden ω_a und ω_b einen Offset, welcher um π rad zu groß ausfällt und so zur Gleichung 5.5 führt.

$$\phi_L = \gamma + \omega_a + \omega_b - \pi \quad (5.5)$$

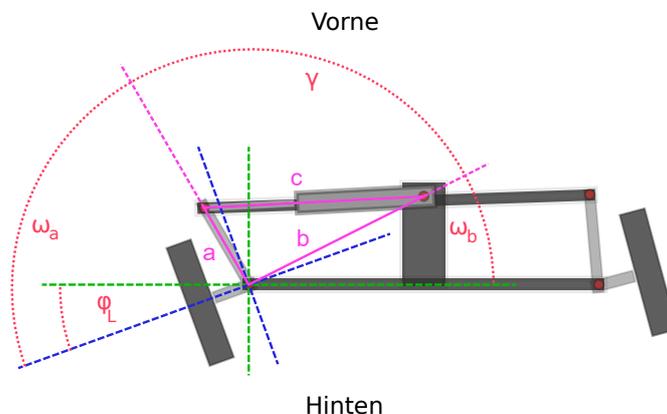


Abbildung 5.2: Bestimmung des Lenkwinkels ϕ_L

Geschwindigkeitsbestimmung Aus der Position des Linearantriebs c und der Lineargeschwindigkeit v , sowie den Parametern L , S , a , b , ω_a und ω_b lässt sich die Rotationsgeschwindigkeit $\dot{\theta}$ durch die Gleichung 5.6 ausdrücken.

$$\dot{\theta} = \frac{v}{\frac{L}{\tan(\arctan(\frac{a^2+b^2-c^2}{2*a*b})+\omega_a+\omega_b-\pi)} + \frac{S}{2}} \quad (5.6)$$

Die Lineargeschwindigkeit v ist in Bezug auf ein körperfestes Koordinatensystem angegeben. Um die Lineargeschwindigkeiten v_{x_k} und v_{y_k} in einem übergeordneten, nicht körperfesten Koordinatensystem k zu bestimmen, werden die Gleichungen 5.7 und 5.8 nach [19] verwendet. Dafür wird neben der Lineargeschwindigkeit v zusätzlich die Ausrichtung θ_k der Berechnungsmaschine im Koordinatensystem k benötigt.

$$v_{x_k} = v * \cos(\theta_k) \quad (5.7)$$

$$v_{y_k} = v * \sin(\theta_k) \quad (5.8)$$

5.1.3 Betriebsmodi

Die Berechnungsmaschine kann sich in den folgenden Betriebsmodi befinden:

- Aktiv: Die Elektronik ist mit der Spannungsquelle verbunden.
- Inaktiv: Die Elektronik ist von der Spannungsquelle getrennt.
- Bereit: Der Controller wartet auf ein Ereignis, um zunächst die Batterie zu testen und bei Erfolg alle Systeme zu starten. Er befindet sich im Tiefschlafmodus.
- Fehler: Eine Systemkomponente meldet einen Fehler.
- Berechnen: Die Elektronik ist mit der Spannungsquelle verbunden und alle Systeme sind hochgefahren.

In Abschnitt 5.2 ist beschrieben, warum zwischen den Zuständen *Aktiv* und *Bereit* unterschieden wird.

Zwischen den Betriebsmodi gibt es Übergänge, welche beschreiben, welches Ereignis notwendig ist, damit die Berechnungsmaschine in einen spezifischen Betriebsmodus wechselt. Diese Ereignisse können ein geänderter Ladestand der Batterie, ein Fehlerereignis oder eine Interaktion des Benutzers mit den Bedienelementen an der Maschine sein. Folgende Bedienelemente stehen dem Benutzer zur Verfügung:

- Hauptschalter: Schaltet die Batterie und verbindet/trennt dadurch die Spannungsquelle.
- Schlüssel: Aktiviert und deaktiviert die Berechnungsmaschine. Die Software fährt entsprechend hoch oder herunter.
- Notaus: Zwingt bei Betätigung die aktive Berechnungsmaschine in einen Fehlerzustand. Die Fehlerbehandlung wird in Abschnitt 5.1.4 beschrieben.

Die Betriebsmodi und die Übergänge sind in Abbildung 5.3 dargestellt.

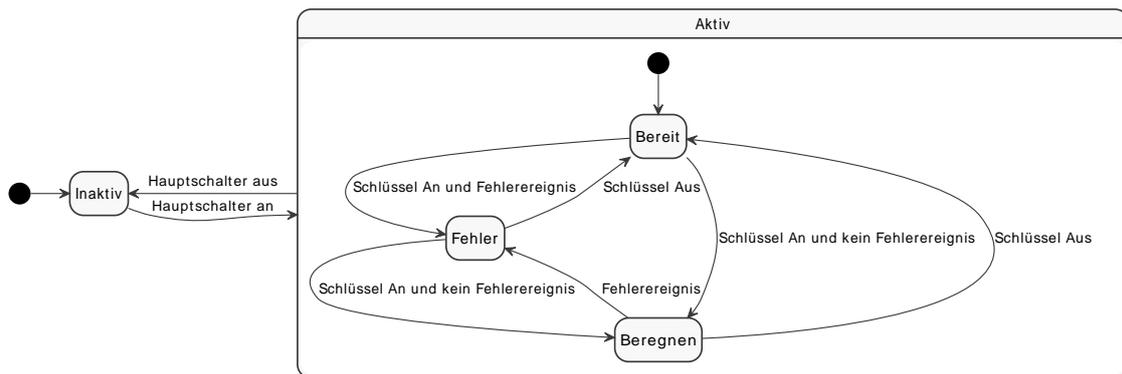


Abbildung 5.3: Zustandsdiagramm über die Power-States der Berechnungsmaschine

5.1.4 Fehlerverhalten

Der Programmcode des Controllers gliedert sich in Komponenten, die für sich gestartet und gestoppt werden können. Die meisten dieser Komponenten verfügen über einen Status, der Werte annehmen kann, die in Tabelle 5.1 aufgelistet sind.

Wert	Bedeutung	Signalfarbe	Maßnahme
0x00: Critical	ist ausgefallen	rot	Stoppe die Berechnungsmaschine und melde den Status
0x01: Warning	funktioniert mit Einschränkungen	gelb	Melde den Status
0x02: Okay	funktioniert gemäß Spezifikation	grün	Melde den Status
0x03: Off	ist abgeschaltet	-	-

Tabelle 5.1: Mögliche Statuswerte einer Komponente des Controller-Programmcodes

Zusätzlich bestimmt der Controller den Systemstatus, welcher mit drei Signallampen angezeigt wird, die, wie in Abschnitt 5.2.2 beschrieben, ein- oder ausgeschaltet werden. Der Status aller Komponenten wird in einer Liste vorgehalten. Sobald sich ein Status ändert, wird der Status mit dem niedrigsten Wert aus der Liste zum Systemstatus. Nach diesem Verfahren ist der Systemstatus kritisch, sobald eine Komponente ihren Status kritisch setzt.

Ein warnender Systemstatus hat mindestens eine der folgenden Ursachen:

- Die Batterieladung ist nur noch für eine kurze Betriebsdauer ausreichend.
- Der Neigungssensor ist nicht kalibriert.

Ein kritischer Systemstatus hat mindestens eine der folgenden Ursachen:

- Der Verbindung zwischen Controller und Computer wurde (noch) nicht hergestellt.
- Die Batteriespannung ist zu niedrig.
- Der Notausschalter ist betätigt worden.
- Die Berechnungsmaschine ist mehr als 30° geneigt.

- Einer der Motoren ist ausgefallen.
- Die Verbindung zum Neigungssensor konnte nicht hergestellt werden.

Wird der Systemstatus kritisch, dann kann die Berechnungsmaschine nicht mehr sicher betrieben werden. In diesem Fall sollen verschiedene Komponenten entweder Stoppen oder Maßnahmen ergreifen, um Schäden an der Berechnungsmaschine zu vermeiden. Alle Komponenten, die bei Auftreten eines kritischen Systemzustandes Maßnahmen ergreifen müssen, werden benachrichtigt. Weiter werden sie benachrichtigt, wenn der Zustand ausheilt. Die Tabelle 5.2 stellt die Maßnahmen der Komponenten dar, die ein kritischer Systemzustand auslöst.

Komponente	Maßnahme
Lenkung	Deaktiviere die Lenkung
Getriebe	Schalte in den Leerlauf
Wasserzufuhr	Schließe Wasserzufuhr
Hupe	Deaktiviere Hupe

Tabelle 5.2: Maßnahmen der Komponenten bei einem kritischen Systemzustand

5.2 Spannungsversorgung

Die Berechnungsmaschine ist entsprechend der Anforderung PR16 von der seed2soil GmbH & Co. KG. mit einer Solarzelle und einer 12 V-Bleibatterie ausgestattet worden. Ein Steuergerät überwacht und lädt die Batterie. Darüber hinaus stellt es einen 12 V-Anschluss und einen 5 V-USB-Anschluss zur Verfügung.

Die Berechnungsmaschine darf nur starten, wenn die Spannung der Batterie ausreichend ist. Wenn die Batterie entladen ist, bricht beim Start der Maschine die Spannung ein und die Maschine geht wieder aus. Infolgedessen wird diese nicht ordnungsgemäß heruntergefahren, was zu Schäden am Computer und dessen gespeicherten Daten führen kann. Außerdem verringert nach [20, S. 16 f.] eine Tiefenentladung die Lebensdauer der Batterie drastisch. Daher prüft die Berechnungsmaschine beim Start, ob sich die Batteriespannung in einem sicheren Bereich befindet und bricht diesen gegebenenfalls ab (siehe Abschnitt 5.1.3).

Die Berechnungsmaschine ist batteriebetrieben. Daher steht nur eine begrenzte Menge an Ladung zur Verfügung. Um die Laufzeit der Berechnungsmaschine zu maximieren, soll diese nicht benötigte Systeme abschalten, wenn keine Berechnungsaufgabe ansteht. Dies darf aber nicht den Controller selbst betreffen, da dieser so nicht in der Lage ist, die notwendigen Systeme wieder zu starten, wenn die nächste Berechnungsaufgabe ansteht. Stattdessen wird dieser in einen energiesparenden Schlafmodus versetzt, aus dem er zu gegebener Zeit aufwacht. Um diesen aufrecht zu erhalten, muss das Vorhandensein der Versorgungsspannung gewährleistet sein.

5.2.1 Batterieüberwachung

Der Ladestand einer Batterie kann aus der anliegenden Spannung geschätzt werden. Dieser Wert ist sehr ungenau, da die Spannung durch die anliegende Last beeinflusst wird. Außerdem fällt die Spannung nicht linear mit dem Ladestand ab, weshalb bei mobilen Geräten zusätzlich der Strom gemessen wird. Jedoch lädt die an der Berechnungsmaschine angebrachte Solarzelle die Batterie dauerhaft mit einem dem System unbekanntem Strom. Daher kann aus dem über die Zeit betrachteten Verbrauchsstrom alleine nicht der Ladestand der Batterie errechnet werden. Um den Ladestand präzise zu erfassen, müssen der Ladestrom, der Verbrauchsstrom und die Spannung gemessen werden.

Darauf wird hier verzichtet: Die Berechnungsmaschine kann eine Berechnungsaufgabe unterbrechen, wenn die Batterie einen niedrigen Ladestand erreicht. In diesem Fall bleibt sie auf der Berechnungsfläche stehen und lädt die Batterie mit der Solarzelle. Wenn die Batterie einen ausreichenden Ladestand erreicht, dann setzt die Berechnungsmaschine die Berechnungsaufgabe fort. Daher genügt es, die an der Batterie anliegende Spannung zu messen.

Spannungsteiler Spannungen lassen sich mit einem () messen. Dieser ist in den meisten Controllern bereits integriert. Allerdings kann ein solcher Baustein nur Spannungen messen, welche im Bereich der eigenen Logikspannung liegen. Negative Spannungen sind nicht messbar. Die meisten modernen Controller werden mit einer Spannung von 3,3V betrieben, weshalb die verwendeten ADCs eine Spannung zwischen 0V und 3,3V messen können. Um die maximal anliegende Spannung von 14,4V messen zu können, wird ein unbelasteter Spannungsteiler nach Abbildung 5.4 konstruiert, welcher die Versorgungsspannung linear reduziert.

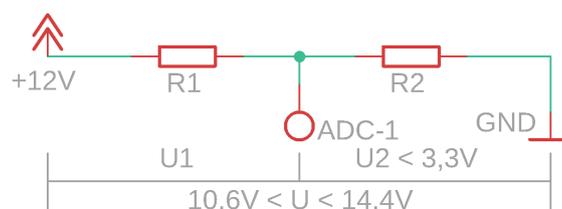


Abbildung 5.4: Aufbau eines Spannungsteilers in der Berechnungsmaschine

Dieser besteht aus zwei in Reihe geschalteten Widerständen. Die Spannung fällt hinter dem ersten Widerstand auf das zu erreichende Niveau ab. Hinter dem zweiten Widerstand beträgt die Spannung 0V. Damit zwischen den Widerständen eine Spannung zwischen 0V und 3V anliegt, müssen die Widerstände in einem zu bestimmenden Verhältnis zueinander gewählt werden, welches mit der Gleichung 5.9 nach [21, S. 130] berechnet wird. Dabei wird die Spannung auf 3V begrenzt, da viele ADCs oberhalb von 3V unpräzise werden.

Mit der Gleichung 5.9 wird errechnet, dass R_1 ungefähr den vierfachen Widerstandswert von R_2 aufweisen muss. Der Gesamtwiderstand des Spannungsteilers sollte nicht zu niedrig gewählt werden, da zwischen 12V und GND dauerhaft ein Strom fließt. Mit $R_1 = 15\text{ k}\Omega$ und $R_2 = 3,9\text{ k}\Omega$ ergibt sich bei 12V Batteriespannung nach Gleichung 5.10 ein konstanter

$$\begin{aligned}
U_2 = \frac{U}{R_1 + R_2} * R_2 &\iff \frac{U_2}{U} = \frac{R_2}{R_1 + R_2} \\
\implies \frac{3V}{14,4V} \approx \frac{1}{5} = \frac{R_2}{R_1 + R_2} &\iff 5 = \frac{R_1}{R_2} + 1 \\
&\iff R_1 = 4R_2
\end{aligned} \tag{5.9}$$

Strom von 0,6349 mA.

Die Widerstände können nicht beliebig groß gewählt werden. Wenn der Strom sehr klein ist, dann können durch Störungen induzierte Ladungen nur langsam abgeleitet werden, was die Spannung U_2 und damit die Messung verfälscht. Mit den für R_1 und R_2 gewählten Werten misst der Controller Spannungen, die in Teilen mit einem Spannungsmessgerät bestätigt werden können. Allerdings treten insbesondere im unteren und oberen Messbereich Abweichungen auf, die sich auf Charakteristiken des im Controller integrierten ADC zurückführen lassen. Daher ist eine nachträgliche Aufbereitung des Messergebnisses erforderlich.

$$I = \frac{U}{R_1 + R_2} \implies I = \frac{12V}{18900\Omega} = 0,6349 \text{ mA} \tag{5.10}$$

Aufbereitung der Spannungsmessung Der in Abschnitt 5.3.1 ausgewählte Controller verfügt nach [22, S. 23] über einen ADC mit einer Auflösung von zwölf Bits. Damit kann dieser den messbaren Spannungsbereich auf $2^{12} = 4096$ unterschiedliche Werte abbilden. Der messbare Bereich wird mit Gleichung 5.11 nach [21, S. 130] bestimmt und liegt zwischen 0 V und 16 V.

$$\begin{aligned}
U_2 = \frac{U}{R_1 + R_2} * R_2 &\iff \frac{1}{U} = \frac{R_2}{(R_1 + R_2) * U_2} \iff U = \frac{R_1 U_2}{R_2} + U_2 \\
&\implies U = \frac{15\text{k}\Omega * 3,3V}{3,9\text{k}\Omega} + 3,3V = 16V
\end{aligned} \tag{5.11}$$

Damit sollte der ADC in der Lage sein, die Spannung auf $\frac{16V}{4096} = 3,9\text{mV}$ genau zu bestimmen. Bei der Evaluation mit einem Spannungsmessgerät nach [23] stellte sich jedoch heraus, dass die Messung durch den ADC um bis zu 1 V abweicht. Darüber hinaus unterscheidet sich das Verhalten zwischen verschiedenen Pins. Die gemessenen Spannungen sind ungenau, aber wiederholbar. Daher bietet sich eine `LuT` als Lösung an. Diese speichert zu jedem messbaren Wert den tatsächlichen Wert. Der gemessene Wert ist dann der Index des tatsächlichen Wertes in der `LuT`, wodurch keine komplexen Berechnungen zur Korrektur des Fehlers notwendig sind.

Allerdings ist es sehr aufwändig eine solche `LuT` manuell zu erstellen, da insgesamt 4096 Messungen durchgeführt werden müssen. Diesem Problem hat sich Helmut Weber in [24] angenommen und ein Verfahren zur Erstellung einer Pin-bezogenen `LuT` mit dem ESP32-Controller erstellt. Dieser verfügt zusätzlich zu dem ADC über einen `ADC1`, mit welchem

Spannungen im Bereich von 0 V bis zur Logikspannung erzeugt werden können. Der integrierte DAC löst die zu erzeugende Spannung nach [22, S. 24] nur mit acht Bits auf. Die dadurch entstehende Ungenauigkeit ist dennoch deutlich kleiner als die Abweichung der Messung des ADC vom tatsächlichen Wert.

Helmut Weber verbindet einen analogen Ausgangspin mit dem zu kalibrierenden, analogen Eingangspin, erzeugt eine Spannung am Ausgangspin und notiert, wie weit die am Eingangspin gemessene Spannung abweicht. Da der ADC seine Messungen mit zwölf Bits auflöst, kann Helmut Weber nur für jeden 16. Wert eine Messung durchführen. Die übrigen Einträge der LuT schätzt er durch eine lineare Interpolation und gibt die gesamte Tabelle über ein serielles Interface aus. Die Tabelle wird als konstantes Array in den Programmcode des Controllers eingetragen.

Unter der Verwendung der erzeugten LuT weichen die Messungen nur um maximal 100 mV von den erwarteten Ergebnissen ab. Im Bereich zwischen 0 V und 1,2 V misst das System allerdings weiterhin 0 V und Spannungen oberhalb von 15 V sind nicht messbar.

Filtern von Störungen Während der ADC in einer Laborumgebung stabile Messungen liefert, treten vereinzelt Messungen mit Abweichungen von mehreren Volt auf, wenn das System an die Berechnungsmaschine angeschlossen ist. Wie nach [22, S. 23] empfohlen, glättet ein Keramik Kondensator mit einer Kapazität von 100 nF zwischen GND und dem analogen Pin die Messergebnisse. Aber dieser genügt nicht, um dauerhaft stabile Messergebnisse zu liefern. Im Fall der Batterie kann eine fehlerhafte Messung die Berechnungsmaschine kurzzeitig in einen Fehlerzustand versetzen. Dieses Verhalten lässt sich an allen analogen Pins beobachten und betrifft damit auch die Messungen der Positionen der Linearantriebe. Ein Medianfilter nach Algorithmus 1 löst das Problem.

```

input : integer n
input : integer pin
output: integer median
1 liste of size n;
2 for i ← 1 to n do
3   | liste[i] ← messeSpannung(pin);
4 end
5 sortiere(liste);
6 if n is even then median ← (liste[ $\frac{n}{2}$ ] + liste[ $\frac{n}{2} + 1$ ])/2 ;
7 else median ← liste[ $\frac{n}{2}$ ] ;
8 return median;

```

Algorithm 1: Medianfilter: *n* ist die Anzahl der durchzuführenden Spannungsmessungen am Pin mit der Nummer *pin*

Je größer *n* gewählt wird, desto stabiler ist der gemessene Wert. Allerdings dauert es verhältnismäßig lange, einen Wert aus dem ADC auszulesen. Daher muss *n* so groß wie nötig und so klein wie möglich gewählt werden. Um das jeweilige *n* zu finden, wird dieses sukzessiv erhöht, bis der Medianfilter stabile Werte liefert. Die Messungen aus den Potentialmetern nach Abschnitt 5.3.2 müssen besonders stabil sein, weshalb hier ein größeres *n* gewählt wird. In Tabelle 5.3 ist der ermittelte Parameter für die jeweilige Applikation

dargestellt.

	Batterie	Lenkung	Getriebe
n	8	32	8

Tabelle 5.3: Parameter für die Medianfilter der aufgelisteten Applikationen

5.2.2 Schaltung der Versorgungsspannung

Der Controller muss in der Lage sein, eine 12 V-Versorgungsspannung für Geräte oder Lampen zu schalten. Er verfügt über digitale Ausgangspins, die eine Logikspannung von 3,3 V ausgeben können. Dies ist aber nur ein Potenzial; ein solcher Pin kann nach [22, S. 43] nur einen Strom von 40 mA liefern. Daher wird ein Baustein benötigt, der auf ein 3,3 V-Signal reagiert und die 12 V-Versorgungsspannung dem Signal entsprechend schaltet. Dieses muss im Ruhezustand sperren. Fällt der Controller aus oder ist dieser noch nicht gestartet, darf das Bauteil die Versorgungsspannung nicht an den Verbraucher legen. Nur wenn der Controller ein entsprechendes Signal setzt und hält, darf das Bauteil die Spannung anlegen.

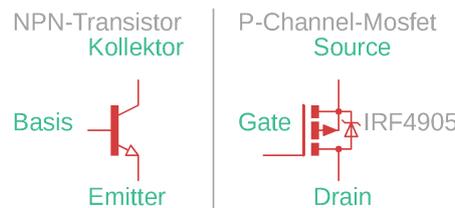


Abbildung 5.5: Schaltzeichen und Belegung zu NPN-Transistor und P-Channel-Mosfet

Ein Transistor funktioniert wie ein einstellbarer Widerstand. Wenn zwischen Basis und Emitter eines NPN-Transistors eine Spannung größer als die Schwellspannung U_{BE} angelegt wird, dann fängt dieser an zu leiten, sodass die Spannung zwischen Kollektor und Emitter abfällt. Ein P-Channel-Mosfet ist einem Transistor ähnlich, kann aber im Gegensatz zum NPN-Transistor sehr hohe Ströme leiten. Er benötigt eine negative Gate-Source-Spannung U_{GS} , um zu leiten. Nur für $U_{GS} = 0$ V sperrt dieser vollständig, sodass am Gate die Versorgungsspannung anliegen muss. Bei $U_{GS} \leq U_{DS}$ hat der P-Channel-Mosfet seinen geringsten Widerstand. Anhand dieser Bedingungen und Eigenschaften wird die in Abbildung 5.6 gezeigte Schaltung konstruiert.

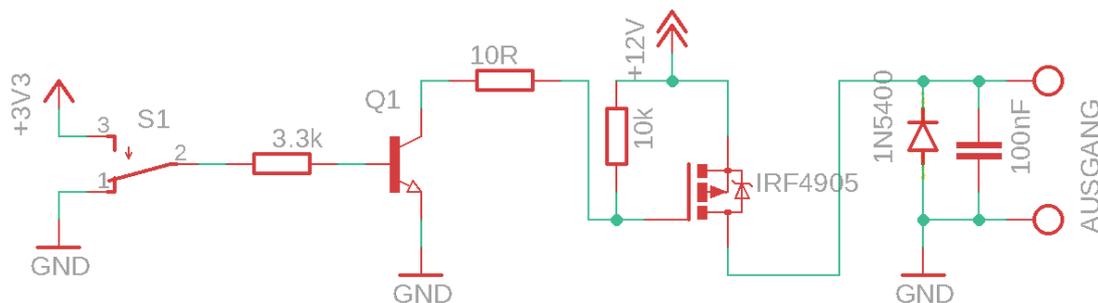


Abbildung 5.6: Aufbau der Schaltung zum Schalten einer 12 V-Versorgungsspannung

S1 repräsentiert den digitalen Ausgangspins des Controllers. Für den Transistor Q1 wird

ein Baustein mit der Bezeichnung ULN2001AN nach [25] verwendet, da er kompakt sieben NPN-Transistoren in einem Baustein kombiniert. Der P-Channel-Mosfet ist ein IRF4905 nach [26], welcher Ströme bis zu 74 A leiten kann.

Im Ruhezustand liegt am Transistor $Q1$ eine kleinere Spannung U_{BS} als 2,4 V, sodass dieser nach [25, S. 6] sperrt. Durch den 10 k Ω -Widerstand wird das Gate des P-Channel-Mosfets auf 12 V gezogen und U_{GS} ist 0 V. Der Mosfet sperrt und der Ausgang ist abgeschaltet.

Im eingeschalteten Zustand wird die Basis des Transistors $Q1$ mit Elektronen gesättigt. Der 3,3 k Ω -Widerstand begrenzt den Strom, um den Controller zu schützen. Der Transistor $Q1$ leitet und zieht das Gate des P-Channel-Mosfets auf GND, wobei der 10 Ω -Widerstand den Strom begrenzt. Dadurch wird $U_{GS} = -12$ V und es gilt $U_{GS} \leq U_{DS}$. Der P-Channel-Mosfet leitet und die Versorgungsspannung liegt am Ausgang an. Am Ausgang filtert ein 100 nF-Keramikkondensator vom Verbraucher ausgehende Störungen. Eine Freilaufdiode Type 1N5408 nach [27] führt vom Verbraucher erzeugte Spannungsspitzen ab.

5.3 Steuerung

Die Berechnungsmaschine verfügt über eine Lenkung und einen Kugelhahn, über welche die Maschine gesteuert wird. Mit der Lenkung lässt sich die Fahrtrichtung beeinflussen, mit den Kugelhahn lässt sich die Wasserzufuhr ein- und ausschalten. Weiter hat die seed2soil GmbH & Co. KG, entsprechend der Anforderung PR12, die Berechnungsmaschine mit einem Gangwahlhebel ausgestattet, mit welchem sich einstellen lässt, ob die Berechnungsmaschine vorwärts oder rückwärts fährt. Diese mechanischen Schnittstellen werden durch das System zur Automatisierung der Berechnungsmaschine angesteuert.

5.3.1 Hardware

Lenkung, Hahn und Hebel müssen durch elektrische Aktoren gestellt werden, damit diese durch ein informationstechnisches System gesteuert werden können.

Motoren Lenkung, Hebel und Hahn sollen mit elektrischen Aktoren in Form von Motoren gestellt werden können. Hierfür muss eine geeignete Hardware ausgewählt werden. Die Tabelle 5.4 zeigt Eigenschaften, welche Lenkung, Hebel und Hahn jeweils aufweisen.

	Lenkung	Hebel	Hahn
mögliche Stellungen	∞	3	2
Länge der Stellstrecke	40cm	15cm	10cm
notwendiger Kraftaufwand	hoch	hoch	gering
notwendige Geschwindigkeit	gering	gering	mittel

Tabelle 5.4: Mechanische Eigenschaften von Lenkung, Hebel und Hahn der Berechnungsmaschine

Da die Stellpositionen von Lenkung, Hebel und Hahn jeweils nach unten und nach oben begrenzt sind, eignen sich Linearantriebe, um diese zu bewegen. Ein Linearantrieb fährt einen Kolben aus einem Gehäuse, um eine am Ende des Kolbens angebrachte Mechanik

zu bewegen. Die Geschwindigkeit, mit der sich der Kolben bewegt, und die Kraft, welche dieser ausüben kann, hängen vom Untersetzungsverhältnis des im Linearantrieb verbauten Getriebes ab. Erreicht der Kolben die Maximalposition, wird ein Schalter betätigt und dadurch der Motor gestoppt. Bei umgekehrter Polarität der Versorgungsspannung fährt der Kolben wieder in das Gehäuse. Äquivalent zur Maximalposition stoppt der Motor auch in der Minimalposition.

Zur Ansteuerung des Kugelhahns genügt ein solcher Linearantrieb, welcher in der Minimalposition den Hahn schließt und in Maximalposition den Hahn öffnet. Damit wird der Anforderung PR13 entsprochen.

Ein Linearantrieb alleine reicht für die Ansteuerung der Lenkung nicht aus: Wenn nur ein einfacher Linearantrieb verwendet werden würde, dann könnte die Lenkung nur voll rechts oder voll links eingeschlagen werden. So ist zusätzlich ein Sensor zur Erfassung der Position des Kolbens notwendig. Mit diesem ist die ansteuernde Elektronik, im folgenden Controller genannt, in der Lage, die am Motor anliegende Spannung entsprechend der Abweichung der aktuellen Position von der Zielposition zu regeln und so den Kolben zu jeder erreichbaren Position zu fahren. Das Verfahren wird im Abschnitt 5.3.2 beschrieben. Linearantriebe können mit Potentiometern ausgestattet werden. Diese verändern auf einer Leitung einen ohmschen Widerstand, abhängig von der Position des Kolbens. Nach [28] nimmt dieser Widerstand mit dem Herausfahren des Kolbens linear ab und erreicht $0\ \Omega$ an der Maximalposition.

Das Potentiometer wird mit drei Adern angeschlossen: Versorgungsspannung, GND und Signal. In der Maximalposition hat die Signal-Ader gegenüber GND die Versorgungsspannung, in der Minimalposition besteht zu GND eine Spannung von 0 V . Die am Signal anliegende Spannung repräsentiert die aktuelle Position des Kolbens.

Für den Gangwahlhebel reichen drei Stellungen aus:

- Maximalposition: Rückwärtsgang,
- Mittelstellung: Leerlauf,
- Minimalposition: Vorwärtsgang.

Ein Potentiometer mit einer theoretisch unendlichen Genauigkeit zur Messung der Kolbenposition scheint hier überdimensioniert. Allerdings muss die Mittelstellung des Hebels genau erreicht werden, damit sich die Gangschaltung im Leerlauf befindet. Für den Leerlauf ist das präzise Anfahren einer Kolbenposition erforderlich. Weiter können so für die Lenkung entwickelte Hard- und Softwarekomponenten auch für die Gangschaltung verwendet werden, was den Entwicklungsaufwand insgesamt reduziert.

Zur Steuerung der Berechnungsmaschine werden folgende Hardwarekomponenten verwendet:

- 2x Linearmotor mit Endschaltern und verbautem Potentiometer nach [28] zur Positionsbestimmung für Lenkung und Gangwahlhebel nach Anforderungen PR12.1 und PR14,

- 1x Linearmotor mit Endschaltern für den Kugelhahn nach PR13,
- 3x Motortreiber zur Steuerung der Motorversorgungsspannung nach [29].

Motortreiber Die Motoren werden über zwei Adern mit 12 V-Gleichstrom versorgt. Um die Fahrtrichtung umzukehren, wird die Polarität des Gleichstroms umgekehrt. Die Geschwindigkeit des Motors, bzw. die vom Motor erzeugte Kraft, muss beeinflussbar sein. Wird ein stehender Motor mit der Versorgungsspannung versorgt, fließt im Einschaltmoment ein hoher Strom. Noch stärker ist dieser im Umschaltmoment, wenn der Motor seine Fahrtrichtung ändert. Der starke Strom beansprucht die elektrischen Bauteile und den Motor durch die entstehende Wärmeentwicklung und kann Störungen verursachen. Die entstandene Hitze ist Verlustwärme und erhöht den Energieverbrauch des Gesamtsystems. Ungedämpfte Ein- und Umschaltmomente sind auch für die Mechanik eine hohe Belastung, da auf diese schlagartig die maximale Kraft ausgeübt wird, die der Motor erzeugen kann. Daher sollen die Ein- und Umschaltmomente gedämpft werden.

Um einen Motor langsam zu beschleunigen, könnte man eine Spannung nahe 0 V anlegen und diese bis auf die Versorgungsspannung erhöhen. Eine flexible, steuerbare Spannungsquelle ist im System aber nicht gegeben. Stattdessen werden die Motoren über den sogenannten Tastgrad angesteuert. Dieser gibt nach [30] das Verhältnis von Impulsdauer zu Periodendauer eines Rechtecksignals an. Wird ein Motor mit einer 12 V-Versorgungsspannung und einem Tastgrad von 0,5 betrieben, dann liegt die Versorgungsspannung zur Hälfte der Periodendauer am Motor an. Der Motor ist dann zur Hälfte der Zeit eingeschaltet. Über die Periodendauer übt der Motor die Hälfte der möglichen Kraft aus (theoretisch, Verluste nicht betrachtet). Je geringer die Periodendauer, desto feiner lässt sich der Motor ansteuern.

Diese Funktionen können mit einer H-Brücke nach [31] erreicht werden. Eine H-Brücke ist eine elektrische Schaltung, welche eine Versorgungsspannung sowie die Eingangssignale zur Motoraktivierung und zur Richtungssteuerung in eine Versorgungsleitung für einen Motor übersetzt. Diese kann über das Signal der Motoraktivierung ein- und ausgeschaltet werden und ändert je nach Richtungssignal die Polarität der Versorgungsleitung. Kompakte H-Brücken stehen als fertige Schaltung zur Verfügung und nehmen die Rolle des Motortreibers ein. Es gilt ein fertiges Bauteil auszuwählen, welches folgende Kriterien erfüllt:

- Geeignet für 11 V-14,4 V Versorgungsspannung, da dies dem Spannungsbereich der bereits vorhandenen Blei-Batterie entspricht.
- Geeignet für Stromstärken bis 10A, da dies der höchstmöglichen Last entspricht, die von den verbauten Motoren erzeugt wird.
- Kompatibel zu 3,3 V Logikspannung. Moderne Controller können nur Signale mit höchstens dieser Spannung erzeugen.
- Geeignet für Rechtecksignale mit einer Frequenz von 25 kHz, um Geräusche im hörbaren Bereich durch Schwingungen der Motorspulen zu vermeiden.
- Geeignet für Gleichstrom-Bürstenmotoren.

- Verfügbarkeit einer Schnittstelle zur Richtungs- bzw. Polaritätssteuerung.
- Überlast- und Kurzschlusschutz, um die Konsequenzen von Fehlern im Entwicklungsprozess zu begrenzen.
- Kompakte Bauform, die sich im vorhandenen Schaltschrank unterbringen lässt.
- Steckbar in ein 2,54 mm Raster, damit die Komponente im Fall eines Defekts schnell ausgetauscht werden kann.

Nach dem Datenblatt [29] erfüllt der Pololu G2 18v17 Motortreiber diese Kriterien und wird zur Ansteuerung der Linearmotoren für Lenkung, Gangwahl und Wasserversorgung verwendet. Folgende zusätzliche Funktionen bietet dieser Baustein an, die verwendet werden sollen:

- Ein ausgehendes Fehlersignal zeigt eine Überlast oder einen Kurzschluss an.
- Ein eingehendes Sleep-Signal ermöglicht den Motor und die Motortreiberelektronik abzuschalten.

Ein Schaltplan wird erstellt, mit dem ein externer Dienstleister eine Platine anfertigen kann, auf welcher die ausgewählten Komponenten angeordnet und korrekt verbunden werden. Für den Schaltplan wird ein Baustein angelegt, welcher die zu belegenden Signale und Leitungen nach [29] darstellt. Dieser zeigt in Abbildung 5.7 die verfügbaren Pins.



Abbildung 5.7: Schaltsymbol mit den verfügbaren Pins am Pololu G2 Motortreiber nach [29]

Die Platine bietet vier Steckplätze für den ausgewählten Motortreiber. Mit einem späteren Prototypen möchte die seed2soil GmbH & Co. KG die Verwendung eines Elektromotors als Antrieb erproben und benötigt daher noch einen weiteren Steckplatz für einen Motortreiber. Daraus ergeben sich folgende Voraussetzungen an den noch auszuwählenden Controller:

- Der Controller muss vier voneinander unabhängige Rechtecksignale mit einer Frequenz von jeweils 25kHz erzeugen können.
- Der Controller benötigt acht digitale Ausgangsleitungen mit einer Logikspannung von 3,3V für die Signale zur Richtungssteuerung und die Abschaltung.
- Der Controller benötigt einen Interrupt-Pin zur Überlast- und Kurzschlussdetektion.

Die Anzahl der Interrupt-Pins, die ein Controller zur Verfügung stellen kann, ist begrenzt. Daher sollen die FLT-Signale aller Motortreiber in einem FLT-Kanal zusammengefasst werden. Der Controller erkennt weiterhin, dass ein Fehler aufgetreten ist, verliert aber

die Möglichkeit, den Fehler einem Motor zuzuordnen. Sobald ein Fehler an einem Motor auftritt, kann die korrekte und sichere Steuerung der Berechnungsmaschine nicht mehr sichergestellt werden. Deshalb wird die gesamte Motorelektronik deaktiviert und die Maschine geht in einen Fehlerzustand über. Das Verhalten im Fehlerfall ist also unabhängig vom betroffenen Motor, weshalb der Fehler nicht zugeordnet werden muss.

Die FLT-Pins der gewählten Motortreiber sind Ausgänge [29]. Wenn diese verbunden werden, zieht im Fehlerfall ein Motortreiber die FLT-Ausgänge der anderen auf eins, oder die anderen Motortreiber ziehen den Fehlerkanal auf null. Dadurch würde ein ausgelöstes Fehlersignal ausgelöscht werden. Das ist so nicht vorgesehen, weshalb nach Abbildung 5.8 zwischen dem jeweiligen FLT-Ausgang und dem Fehlerkanal eine Halbleiterdiode vom Typ 1N4148 [32] platziert wird. Diese erlaubt nur eine Signalrichtung vom FLT-Pin zum Fehlerkanal, wodurch kein Signal vom Fehlerkanal in einen Motortreiber gelangen kann. Diese Halbleiterdiode ist sehr kompakt und reduziert die Spannung kaum, kann dafür aber nur sehr kleine Ströme von maximal 0,2A führen. Die Ströme sind in Logiknetzen sehr gering, weshalb sich diese Diode für diese Anwendung sehr gut eignet.

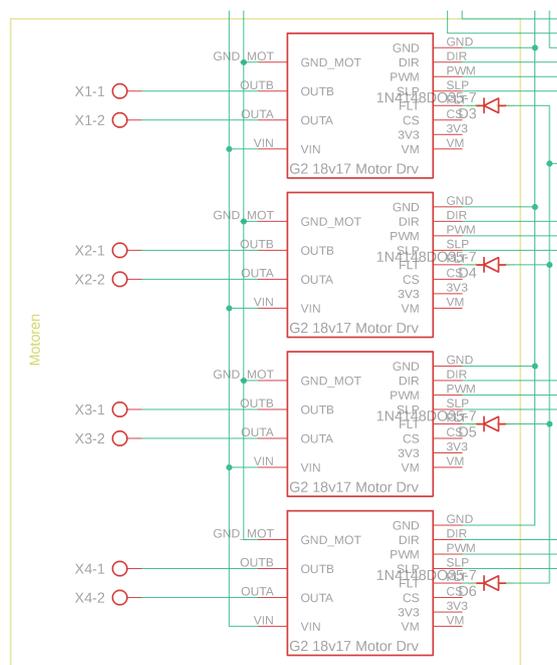


Abbildung 5.8: Verschaltung der Motortreiber

Controller Der Markt bietet eine große Vielfalt an Controllern. Es gilt ein geeignetes Modell aus diesen auszuwählen, welches folgende Kriterien erfüllen muss:

1. Betrieb mit 3,3V Logikspannung nach Abschnitt 5.3.1,
2. vier voneinander unabhängige Pins zur Erzeugung von Rechtecksignalen mit einer Frequenz von 25kHz nach Abschnitt 5.3.1,
3. 14 digitale Ausgangspins nach Abschnitt 5.3.1 und Tabelle 5.6,

4. Unterstützung einer Tiefschlaffunktion nach Abschnitt 5.1.3,
5. vier Interrupt-Pins für Power- und Notausschalter nach Abschnitt 5.1.3 und den Inkrementaldrehgeber nach Abschnitt 5.4.1,
6. drei Analoge Eingänge, um Spannungen innerhalb der Logikspannung messen zu können. Zu messen sind die Batteriespannung nach Abschnitt 5.2.1 und die Motorstellungen von Lenkung und Hebel nach Abschnitt 5.3.1,
7. I2C-Schnittstelle, um Sensoren zur Neigungserkennung nach Abschnitt 5.4.4 anbinden zu können,
8. UART oder USB-Schnittstelle mit Unterstützung einer hohen Baudrate,
9. gute, offene Dokumentation,
10. steckbar in ein 2,54mm Raster, damit der Controller bei einem Ausfall schnell ausgetauscht werden kann,
11. langfristig verfügbar am Markt, damit der Controller verfügbar ist, wenn aus dem Prototyp ein Produkt werden sollte,
12. Kompatibilität mit der ROS-Serial-Bibliothek, damit eine Kommunikation mit einer ROS-Umgebung leicht umgesetzt werden kann,
13. programmierbar via USB, damit die Software des Controllers mit geringem Aufwand geändert werden kann.

Durch Voraussetzung 12 wird nach [33], Abschnitt *Client Libraries* die Liste infrage kommender Controller stark reduziert. Viele der verbleibenden Controller verwenden dieselbe oder sehr ähnliche Recheneinheiten. Die Voraussetzungen 1 bis 8 sind abhängig von der auf dem Controller verwendeten Recheneinheit. Deshalb werden diese in Tabelle 5.5 zunächst verglichen.

Chip	Takt	Strom	Voraussetzung							
			1	2	3	4	5	6	7	8
ATmega328P [34]	16MHz	14mA	-	-	-	x	-	x	x	x
ATmega2560 [35]	16MHz	14mA	-	-	x	x	x	x	x	x
ATmega32u4 [36]	16MHz	27mA	-	-	-	x	x	x	x	x
AT91SAM3X8E [37]	84MHz	150mA	x	x	x	x	x	x	x	x
Kinetis KL26 [38]	48MHz	6,2mA	x	x	-	x	x	x	x	x
Kinetis K66 [39]	180MHz	80mA	x	x	x	x	x	x	x	x
STM32F1 [40]	72MHz	47,3mA	x	x	x	x	x	x	x	x
Xtensa LX106 [41]	160MHz	80mA	x	x	x	x	x	x	x	x
Xtensa LX6 [22]	240MHz	68mA	x	x	x	x	x	x	x	x

Tabelle 5.5: Vergleich von geeigneten Controllern mit den in Abschnitt 5.3.1 definierten Kriterien

Der Xtensa LX6 sticht heraus. Nach [22] wird dieser auf dem ESP32-Controller verlötet, welcher die Voraussetzungen 9 bis 13 erfüllt. Ebenfalls eignet sich der Kinetis K66, welcher zusätzlich einen CAN-Bus zur Anbindung weiterer Peripherie bietet. Dennoch soll

der ESP32 als Controller verwendet werden, da dieser abseits der Voraussetzungen WLAN und Bluetooth unterstützt. Diese Schnittstellen ermöglichen eine kabellose Anbindung von Komponenten, wie Sensoren oder Aktoren, und damit ein hohes Maß an Flexibilität bezüglich deren Anbringung. Weiter bietet der ESP32 zwei Kerne und erreicht damit eine Taktrate von 2x240MHz.

Für den Schaltplan wird ein Baustein gezeichnet (siehe Abbildung 5.9), welcher die verfügbaren Pins übersichtlich darstellt.



Abbildung 5.9: Schaltsymbol mit den verfügbaren Pins am ESP32-Controller nach [22]

Pin-Erweiterung Die Anzahl der Pins des ESP32-Controllers reicht, wie in Abschnitt 5.3.1 beschrieben, aus. Alle verwendeten Bausteine sollen auf einer Platine verlötet werden, um eine physikalisch robuste, geschlossene Einheit zu schaffen. Eine nachträgliche Änderung dieser Platine ist nicht möglich. Daher werden auf der Platine Anschlüsse für zusätzliche Komponenten geschaffen. So kann dem System beispielsweise ein weiterer Motor sowie Motortreiber hinzugefügt werden, sollte in der Entwicklungsphase festgestellt werden, dass dies erforderlich ist. In Tabelle 5.6 sind die Komponenten mit den erforderlichen Pins dargestellt.

Komponente	digital Pins	weitere Pins
Lenkmotor	2	2
Getriebemotor	2	2
Wasserhahn	2	1
Motor Backup	2	2
Motor Fehlersignal	0	1
Hupe und Lampen	4	0
Host-System	1	2
12V-Spannungsversorgung	1	0
5V-Spannungsversorgung	1	0
Encoder	0	2
Encoder Backup	0	2
Button Power	0	1
Button Notaus	0	1
Button Backup	0	1
Orientierungssensor	0	2
Debug-Header	1	2
<i>Summe</i>	<i>14</i>	<i>21</i>

Tabelle 5.6: Auflistung über alle Systemkomponenten, welche an den Controller angeschlossen werden und welche Anzahl an Pins diese benötigen. Weitere Pins sind analoge, interrupt- oder Daten-Pins oder Pins zur Erzeugung von Rechtecksignalen.

Nach dem Datenblatt [22] können 26 Pins des EPS32-Controllers verwendet werden. Damit verfügt der ESP32-Controller über eine nicht ausreichende Anzahl an Pins, wenn auf der Platine Spielraum für weitere Komponenten gelassen werden soll. Allerdings lässt sich die Anzahl digitaler Pins durch einen zusätzlichen Baustein erhöhen. Muss der ESP32-Controller nur die nicht-digitalen Pins ansteuern, genügt dieser.

Der MCP23017-Baustein bietet nach [42] 16 digitale Pins, welche über eine I2C-Schnittstelle angebunden werden. Zwei Register mit einer Länge von acht Bits können beschrieben werden. Jedes Bit repräsentiert den Zustand eines digitalen Pins.

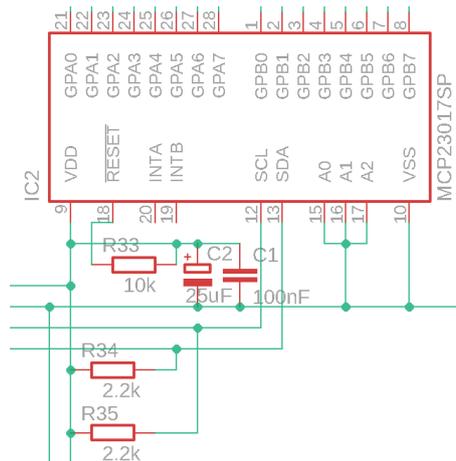


Abbildung 5.10: Schaltsymbol des MCP23017 Bausteins, erstellt und angebunden nach [42]

Der MCP23017-Baustein muss nach [42] angebunden werden, damit dieser korrekt funktioniert. Wie in Abbildung 5.10 dargestellt, werden die Pins A0, A1 und A2 auf GND (VSS) gezogen, um dem Baustein die I2C-Adresse 0x00 zuzuweisen. Der Reset-Pin wird durch einen PullUp-Widerstand (R33) beim Einschalten auf Versorgungsspannung (VDD) gezogen, was den Baustein starten lässt. Zwischen GND (VSS) und der Versorgungsspannung (VDD) werden ein Keramik-Kondensator (C2) und ein Elektrolyt-Kondensator (C1) geschaltet. Der Keramik-Kondensator filtert hoch frequentierte Störungen mit niedriger Amplitude, der Elektrolyt-Kondensator niedrig frequentierten Störungen mit hoher Amplitude.

Der I2C-Bus (SDA/SCL) ist eins, wenn keine Daten gesendet werden und wird daher mit PullUp-Widerständen (R34/R35) auf die Versorgungsspannung (VDD) gezogen. Je höher der Widerstand, desto geringer der Stromfluss im Logiknetz. Allerdings führt ein hoher Widerstand nach [43] zu einer erhöhten Störungsempfindlichkeit, da induzierte Ladungen weniger schnell abgeführt werden können. Innerhalb des Systems gibt es viele Störungsquellen. Darum wurden die Widerstandswerte der PullUp-Widerstände (R34/R35) sukzessive verringert, bis mit 2,2 k Ω ein Wert gefunden wurde, bei dem keine Störungen am Baustein feststellbar sind.

5.3.2 Positionsregelung der Linearantriebe

Die Linearantriebe der Lenkung und des Getriebes müssen präzise zu einer vorzugebenen Position fahren. Dafür gilt es zunächst, die aktuelle Position des Kolbens zu messen. In

Abschnitt 5.3.1 ist die dafür erforderliche Hardware beschrieben. Der Controller misst eine Spannung am Pin, an dem das Potentiometer angeschlossen ist. Aus dem nach 5.2.1 aufbereiteten Messergebnis wird eine Position errechnet. Diese Position und ggfs. Positionen aus der Vergangenheit sind die Datenbasis für einen Regler, welcher die an den Motor anzulegende Leistung bestimmt.

Berechnung der Kolbenposition Die Kolbenposition ist definiert als der Abstand zwischen den beiden Befestigungspunkten des Linearantriebs. Die minimale und maximale Positionen werden aus den Datenblatt [28] zu den Linearantrieben entnommen und in Tabelle 5.7 gegenübergestellt.

	min. Pos. a	max. Pos. b	Strecklänge l	Auflösung r
Lenkung	292 mm	442 mm	150 mm	0,0366 mm
Getriebe	242 mm	342 mm	100 mm	0,0244 mm

Tabelle 5.7: Eigenschaften der Linearantriebe für Lenkung und Getriebe nach [28]

Dem Controller stellt sich die Kolbenposition als Zahl zwischen 0 und 4095 dar. Dabei misst er die am zuständigen Pin anliegende Spannung. Diese steigt linear von 0 V mit dem Herausfahren des Kolbens auf 3,3 V an. Ein Regler arbeitet typischerweise mit einer sehr hohen Frequenz. Jede gemessene Position in das metrische System umzurechnen kostet Rechenzyklen und bringt keinen Mehrwert. Stattdessen wird die Kolbenzielposition auf einen Wert zwischen 0 und 4096 normiert und gespeichert. Dabei wird davon ausgegangen, dass das Ziel seltener gesetzt wird, als die Kolbenposition gemessen wird. Unter dieser Annahme werden Rechenzyklen gespart. Außerdem ist die reale Kolbenposition eine Fließkommazahl. Rechenoperationen auf Fließkommazahlen sind deutlich aufwändiger als Rechenoperationen auf Ganzzahlen. Die gemessene Kolbenposition m und reale Kolbenposition k lassen sich mit der minimalen Position a und der Auflösung r nach Gleichung 5.12 und 5.13 ineinander überführen.

$$k = m * r + a \tag{5.12}$$

$$m = \frac{k - a}{r} \tag{5.13}$$

Berechnung der Kolbengeschwindigkeit Wie in Abschnitt 5.6.2 beschrieben, bestimmt das System anhand der Kolbengeschwindigkeit die zukünftige Kolbenposition, welche notwendig ist, um einen Kurs vorherzusagen. Daher muss der Controller auch die Kolbengeschwindigkeit bestimmen.

Neben der aktuellen Kolbenposition kennt der Controller auch die vergangene Zeit seit der vorherigen Messung, da er die Position immer in einer festen Rate bestimmt. Jedes Mal, wenn der Controller die Position pos bestimmt, bestimmt er auch den Abstand zur vorherigen Position (*distance*) und die seit der letzten Messung vergangene Zeit (*duration*). Nach Algorithmus 2 kombiniert er die Position, den Abstand, die vergangene Zeit und einen Zeitstempel $time$ zu einem Element, fügt es das Ende einer Liste an und löscht das

Element am Anfang der Liste.

Weil der Controller die Position mit einer sehr hohen Rate misst, leidet die Messgenauigkeit, was zu einer sich ständig verändernden, gemessenen Geschwindigkeit führt, selbst wenn die tatsächliche Geschwindigkeit konstant wäre. Daher werden mehrere Geschwindigkeitsmessungen nach dem Sliding-Window-Verfahren in eine Liste eingetragen. Erst wenn die Geschwindigkeit benötigt wird, wird aus allen Einträgen der Liste der Mittelwert gebildet, was zu einer Glättung der gemessenen Geschwindigkeit führt.

```
input : integer pos
input : list history of time, duration, pos, distance
output: float speed

1 integer time ← now();
2 integer duration ← time − history.last.time;
3 integer distance ← pos − history.last.pos;
4 history.append(time, duration, pos, distance);
5 history.remove(history.first);
6 integer duration_sum ← 0;
7 integer distance_sum ← 0;
8 for element : history do
9   | duration_sum ← duration_sum + element.duration;
10  | distance_sum ← distance_sum + element.distance;
11 end
12 speed ←  $\frac{\textit{distance\_sum}}{\textit{duration\_sum}}$ ;
13 return speed;
```

Algorithm 2: Bestimme die Kolbengeschwindigkeit nach dem Sliding-Window-Verfahren

Aufgaben der Regelung Dem Regler werden Ist- und Soll-Kolbenposition übergeben. Er bildet daraus einen Tastgrad und ein Richtungssignal, mit dem ein Linearantrieb angesteuert werden soll. Ist- und Soll-Kolbenposition sind Werte zwischen 0 und 4095, der Tastgrad ist mit acht Bit aufgelöst und liegt daher zwischen 0 und 255. Bei einem digitalen Regler wird eine Routine dauerhaft mit einer vorgegebenen Rate ausgeführt. In der Routine wird der Soll- mit dem Ist-Wert verglichen und das Ausgangssignal entsprechend angepasst. Dadurch wird die Ausgangsgröße ständig neu berechnet und auf Änderungen reagiert. Der hier implementierte Regler ist ein PID-Regler nach [44], welcher einen Proportional-Regler, einen Integral-Regler und Differenzierer kombiniert.

Der Regler muss folgende Probleme lösen:

- Die Messgröße und die Zielgröße liegen in unterschiedlichen Wertebereichen.
- Die Linearantriebe sollen beschleunigen.
- Die Linearantriebe sollen bereits vor Erreichen der Zielposition entschleunigen.
- Die Linearantriebe sollen ihre Zielposition erreichen, solange die aufzubringende Last innerhalb der Motorspezifikation liegt.

Kolbenpositionsregelung Der gesamte Regelkreis ist in Abbildung 5.11 dargestellt.

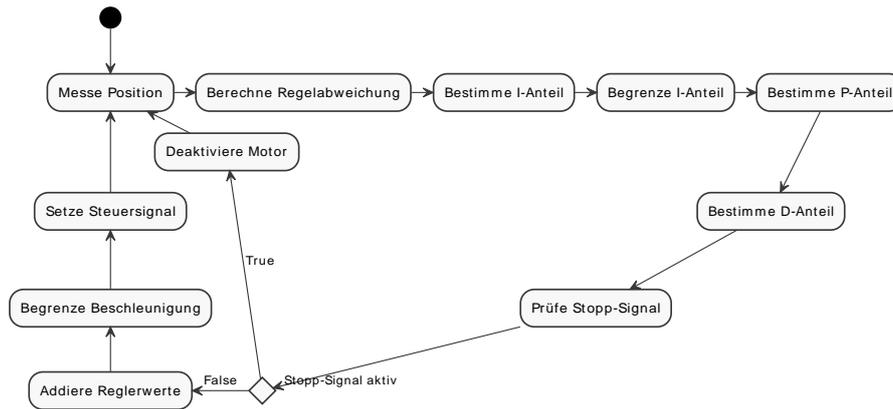


Abbildung 5.11: Regelkreis zur Ansteuerung der Linearantriebe der Lenkung und des Getriebes

Ein klassischer PID-Regler begrenzt nicht zwangsläufig die Beschleunigung des Motors. Wird eine neue Zielposition gesetzt, die weit von der aktuellen Position entfernt ist, dann reagiert der Regler mit einem Sprung auf die maximale Motorleistung. Genau das soll aber verhindert werden, um die Mechanik und die Elektronik zu schonen. Deshalb wird die Ausgabe des PID-Reglers von einem Beschleunigungsbegrenzer nach oben und unten begrenzt. Dieser wirkt beim Be- und Entschleunigen identisch. Er begrenzt den Bereich, in welchem sich die Änderung der Reglerausgabe innerhalb einer Regleriteration befindet. Jedes Mal, wenn der Regelkreis iteriert, wird dessen Ausgabe gespeichert. Die Ausgabe des Reglers wird mit der vorherigen Ausgabe des Regelkreises verglichen. Ist die Änderung zu groß, wird die maximal erlaubte Änderung in Änderungstendenz auf das vorherige Ergebnis des Regelkreises addiert und als neue Ausgabe des Regelkreises gespeichert. Sonst wird die Ausgabe des Reglers übernommen.

Dieser Begrenzer simuliert eine Trägheit des Motors, weshalb er bei der Kalibrierung des PID-Reglers aktiviert sein muss. Sonst provoziert er ein schwingendes Verhalten, bei dem das System nie den Sollwert erreicht.

Wird ein Fehlerereignis ausgelöst, wird die Berechnungsmaschine gestoppt, wie in Abschnitt 5.1.4 beschrieben. Der Regelkreis nimmt ein Stopp-Signal entgegen und deaktiviert den Motor. In diesem Fall wird die Entschleunigung nicht berücksichtigt.

Parametrisierung des Reglers nach [44] Der P-Regler arbeitet proportional. Er bestimmt die Abweichung der Ist-Kolbenposition k von der Soll-Kolbenposition m , multipliziert diese mit einem p -Faktor und errechnet den P -Anteil:

$$P = (m - k) * p \quad (5.14)$$

Ist der p -Faktor zu groß gewählt, fährt der Linearantrieb über die Zielposition hinaus; ist er zu klein gewählt, dann kommt der Linearantrieb mit großem Abstand zur Zielposition zum stehen. Nach der Schwingungsmethode zur Einstellung eines PID-Reglers von Ziegler und Nichols nach [44, Dimensionierung des Reglers] wird ein p_{crt} -Faktor so lange erhöht, bis das

Regelsystem stabil oszilliert. Der Kolben schwingt um die Zielposition und die Amplitude nimmt weder zu noch ab. Die Periodendauer dieser Schwingung T_{krit} wird erfasst und für die Bestimmung des i - und d -Faktors notiert. Der p -Faktor ist wie folgt zu setzen:

$$p = 0,6 * p_{crit} \quad (5.15)$$

Mit dem P-Regler allein wird die Zielposition nicht erreicht; der Kolben bleibt mit etwas Distanz vor der Zielposition stehen. Diese Distanz wird größer, wenn die Last am Lineartrieb erhöht wird.

Der I-Regler bildet ein Integral über die Zeit und die Abweichung. Im Diskreten bedeutet das, dass der I-Anteil über die Regeliteration hinaus gespeichert wird und pro Regeliteration der Fehler auf den I-Anteil addiert wird.

$$I = I + (m - k) * i, \text{ mit } -127 \leq I \leq 127 \quad (5.16)$$

Allerdings muss der I-Anteil in seinem negativen, minimalen bzw. in seinem positiven, maximalen Wert begrenzt werden. Ist der Fehler sehr groß, dann benötigt der Kolben sehr lange, um die Zielposition zu erreichen. Dadurch wächst der I-Anteil über die Zeit so weit an, dass der Kolben über die Zielposition hinausfährt. Hier wurde der Wertebereich des I-Anteils auf $[-127; 127]$ begrenzt, was jeweils der Hälfte des maximalen Ausgabewertes in beide Richtungen entspricht. Der Kolben kommt allein durch den P-Regler der Zielposition sehr nahe, sodass der I-Regler den maximalen Tastgrad nicht anlegen braucht. Fährt der Kolben über das Ziel hinaus, wird der I-Anteil schnell abgebaut und negativ, dass sich der Kolben sehr schnell auf die Zielposition einschwingt.

Der i -Faktor wird entsprechend Ziegler und Nichols nach [44, Dimensionierung des Reglers] gesetzt:

$$i = 0,5 * T_{crit} \quad (5.17)$$

Der D-Regler reagiert auf Änderungen der Abweichung zwischen Soll- und Ist-Position (Regelabweichung).

$$D = ((m - k) - (m' - k')) * d \quad (5.18)$$

Wenn die Regelabweichung zunimmt, dann versucht der D-Regler dagegen zu wirken. Wenn die Regelabweichung abnimmt, dann bremst der D-Regler und kann das Überschießen des I-Reglers reduzieren.

Der d -Faktor wird entsprechend Ziegler und Nichols nach [44, Dimensionierung des Reglers] gesetzt:

$$d = 0,12 * T_{crit} \quad (5.19)$$

P-, I- und D-Anteil werden aufsummiert und das Ergebnis auf den Wertebereich $[-255; 255]$ begrenzt.

$$Out_{[-255;255]} = P + I + D, \text{ mit } -255 \leq P \leq 255 \quad (5.20)$$

Das Vorzeichen von *Out* repräsentiert den Zustand des Richtungssignals; der Betrag von *Out* ist der Tastgrad des erzeugten Rechtecksignals.

Der Regler erzeugt gegebenenfalls einen Tastgrad, der so nahe an null liegt, dass die Leistung nicht ausreicht, um den Motor zu bewegen. Dann fließt ein Strom, welcher lediglich den Motor erwärmt und dennoch die Laufzeit der Batterie reduziert. Ein weiterer Parameter gibt vor, wie weit der Tastgrad von null entfernt sein muss. Jeder Wert, der diesen Parameter unterschreitet, wird auf null gesetzt. Bleibt der Tastgrad über eine definierte Zeit null, dann wird der Motor samt Steuerelektronik durch ein Zurücknehmen des enable-Signals deaktiviert.

Die Schwingungsmethode zur Einstellung eines PID-Reglers von Ziegler und Nichols nach [44, Dimensionierung des Reglers] liefert gute Ausgangswerte für *p*, *i* und *d*. Allerdings müssen diese weiter angepasst werden, bis die Regler das gewünschte Verhalten zeigen. Dafür wurden *p* und *i* inkrementell verändert, bis die Regler mit den Werten aus Tabelle 5.8 das gewünschte Verhalten zeigen.

	p	i	d	Rate	max. Beschl.	min. Tastgrad	Leerlaufzeit
Lenkung	2,25	1	1,5	50 Hz	384 steps/m ²	48	2 s
Getriebe	1,5	0,4	0,75	50 Hz	384 steps/m ²	48	2 s

Tabelle 5.8: Einstellungen der Regelsysteme für Lenkung und Getriebe

5.3.3 Schaltung der Wasserversorgung

Wie in Abschnitt 5.3.1 beschrieben, wird die Wasserversorgung der Berechnungsmaschine mit einem Kugelhahn geöffnet bzw. geschlossen. Dieser wird durch einen Linearantrieb ohne Potentiometer vom Typ TA2-1F des Herstellers TiMOTION angetrieben. Da die Position des Kolbens nicht gemessen werden kann, kann dieser nur komplett eingefahren oder komplett ausgefahren werden. Erreicht der Linearantrieb eine dieser Positionen, trennt nach [45, S. 9] ein im Motor verbauter Schalter die Versorgungsspannung, sodass der Motor stoppt. Mit der Polarität der Motorversorgungsspannung wird vorgegeben, welche der beiden Positionen angefahren werden soll. Dabei ist der Kugelhahn bei eingefahrenen Kolben geschlossen und bei ausgefahrenen Kolben geöffnet.

Ein Motortreiber nach Abschnitt 5.3.1 wird verwendet, um Steuersignale vom Controller zu einer entsprechenden Motorversorgungsspannung zu übersetzen. Um eine hohe Belastung der Mechanik und der Elektronik zu vermeiden, wird auch dieser Linearantrieb be- und entschleunigt (siehe Abschnitt 5.3.2, Paragraf Kolbenpositionsregelung).

Die ansteuernde Softwarekomponente repräsentiert die Wasserversorgung. Sie nimmt einen Zustand an, welcher den aktuellen Status des Kugelhahns möglichst präzise repräsentiert. Die möglichen Zustände und die Zustandsübergänge sind in Abbildung 5.12 dargestellt. Der

Kolben benötigt eine Zeit d , um die anzufahrende Position zu erreichen. Ist die Zeit d nach dem letzten Befehl vergangen, wird der Motor und die Motorsteuerelektronik abgeschaltet. Der Öffnungs- bzw. Schließvorgang kann mit der asynchronen öffnen-Funktion bzw. schließen-Funktion gestartet werden.

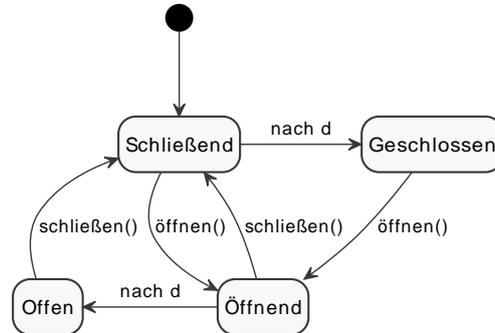


Abbildung 5.12: Zustände, welche die Wasserversorgung annehmen kann; d ist die Zeit, die der Kolben für die gesamte Fahrstrecke benötigt

Befindet sich das System nach Abschnitt 5.1.4 in einem kritischen Systemstatus, dann ist die öffnen-Funktion deaktiviert, wodurch die Wasserzufuhr nicht geöffnet werden kann. Außerdem löst der kritische Systemstatus die schließen-Funktion aus, sodass die Wasserzufuhr geschlossen wird und bleibt.

Mit der Realisierung der Ansteuerung des Kugelhahns ist die Anforderung CR33 erfüllt.

5.4 Lokalisierung

Damit die Berechnungsmaschine einem Berechnungspfad folgen kann, muss sie ihre eigene Pose in Relation zum Berechnungspfad setzen. Wie in Anforderung CR27 gefordert, bestimmt diese dazu ihre Pose global. Die Lokalisierung kombiniert ein absolutes Lokalisierungsverfahren mit einem relativen Lokalisierungsverfahren, um etwaige Schwächen des einen mit den Stärken des anderen auszugleichen. Als absolutes Lokalisierungsverfahren wird RTK eingesetzt. Es bestimmt die Position der Berechnungsmaschine in Form einer global eindeutigen WGS84-Koordinate. Ein zweiter RTK-Empfänger ermöglicht die Berechnung der globalen Ausrichtung. Die relative Lokalisierung erfolgt über die Berechnung der zurückgelegten Strecke. Hierzu wird die Lineargeschwindigkeit der Berechnungsmaschine gemessen und die Winkelgeschwindigkeit mit dem mathematischen Modell aus Abschnitt 5.1.2 berechnet. Über die Zeit integriert wird, ausgehend von einer Ausgangspose, auf eine Endpose geschlossen. Die Zusammenführung der absoluten und relativen Lokalisierung erfolgt durch einen (). Dessen Ergebnis wird als Pose der Berechnungsmaschine angenommen.

Koordinatensysteme Eine Pose bezieht sich immer auf ein Koordinatensystem. Dieses beschreibt eine Achsenkonfiguration, welche als Referenz für die darin liegenden Posen dient. Somit sind Posen relativ zum Ursprung des jeweiligen Koordinatensystems definiert. Zwei Koordinatensysteme werden über Transformationen in Beziehung gesetzt. Transformationen beschreiben die Rotation und die Verschiebung des einen Koordinatensystems,

sodass dessen Achsenkonfiguration dem zweiten gleicht. Durch die Anwendung einer Transformation auf eine Pose wird diese aus dem Ursprungskordinatensystem in das Zielkoordinatensystem übertragen. Die inverse Transformation wirkt in die entgegengesetzte Richtung.

In der Robotik werden Koordinatensysteme über Transformationen in einer Baumstruktur angeordnet. Dadurch kann eine Pose eines Koordinatensystems in jedes andere der Struktur überführt werden. Zur Verwaltung einer solchen Struktur wird das *tf*-Paket eingesetzt. Dieses erlaubt Koordinatensysteme in die gegebene Baumstruktur einzubinden sowie die beschriebene Transformation von Posen. Die Abbildung 5.13 skizziert die zentralen Koordinatensysteme der Berechnungsmaschine. Das *UTM*-Koordinatensystem ist weltfixiert und wird zur Transformation von WGS84-Koordinaten in das *Map*-Koordinatensystem verwendet. Die Startpose der Berechnungsmaschine entspricht dem Ursprung des ebenfalls weltfixierten *Map*-Koordinatensystems. Die Koordinaten eines abzufahrenden Berechnungspfad und der Berechnungsmaschine werden in das *Map*-Koordinatensystem transformiert und in Relation gesetzt. Das *Base Link*-Koordinatensystem ist körperfixiert und hat seinen Ursprung in der aktuellen Pose der Berechnungsmaschine im *Map*-Koordinatensystem. Dadurch sind deren Achsenkonfiguration zum Startzeitpunkt gleich. Alle drei Koordinatensysteme halten die Konventionen der REP 105 [46] ein.

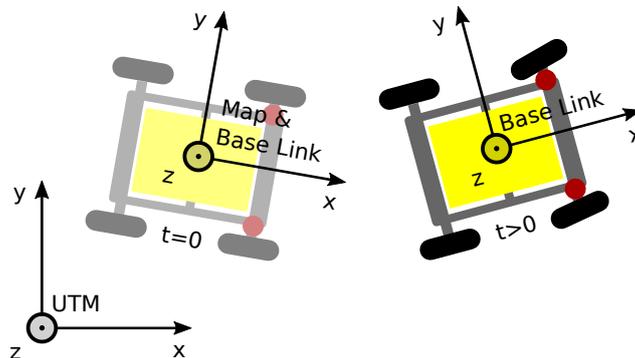


Abbildung 5.13: Aufbau der zentralen Koordinatensysteme UTM, Map und Base Link

Vom *Base Link*-Koordinatensystem ausgehend sind vier körperfixierte Koordinatensysteme auf der Berechnungsmaschine durch statische Transformationen angegeben. Wie in Abbildung 5.14 dargestellt, beschreiben sie jeweils eine Pose an der Berechnungsmaschine, die für Berechnungen verwendet werden. Das *Front GNSS Antenna*- und das *Back GNSS Antenna*-Koordinatensystem geben die Positionen der GNSS-Antennen an. Mit ihnen werden die empfangenen Koordinaten zu einer Koordinate kombiniert und diese ins *Base Link*-Koordinatensystem transformiert (siehe Abschnitt 5.4.2). Die Koordinatensysteme *Front Axle* und *Back Axle* sind jeweils mittig auf der Vorderachse und der Hinterachse mit der Ausrichtung nach vorne platziert. Bei der Abfahrt eines Berechnungspfad werden sie, der Fahrtrichtung entsprechend, als aktuelle Pose der Berechnungsmaschine genutzt (siehe Abschnitt 5.6).

Die Transformationen vom *Base Link*-Koordinatensystem zum *Front GNSS Antenna*-

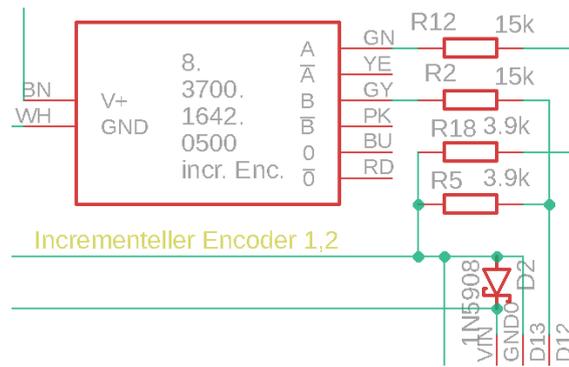


Abbildung 5.15: Verschaltung des Inkrementaldrehgebers mit dem Controller

entsteht bei konstanter Drehgeschwindigkeit größer $0 \frac{\text{rad}}{\text{s}}$ ein perfektes Rechtecksignal. So erreicht der Inkrementaldrehgeber eine Auflösung r_{rot} von $0,18^\circ$.

Aus der gemessenen Rotation muss eine gerichtete Strecke berechnet werden. Mit dem Umfang U_{enc} des rotierenden Kreises wird berechnet, welche Distanz der Inkrementaldrehgeber zwischen zwei Signalen misst. Der rotierende Kreis ist ein gummiertes Rad, welches auf der Welle des Inkrementaldrehgebers befestigt ist und von der Haspel bewegt wird. Der Umfang U_{enc} ist nicht bekannt, weshalb der Durchmesser von $d_{enc} = 60 \text{ mm}$ manuell gemessen wurde. Dieser wird in den Umfang übertragen und die lineare Auflösung r des Inkrementaldrehgebers nach Gleichung 5.21 berechnet.

$$r = \frac{\pi * d_{enc}}{2000} = 0,09425 \text{ mm} \quad (5.21)$$

Jedes Signal löst im Controller einen Interrupt aus. Wie in Abbildung 5.16 dargestellt, sind die beiden Kanäle um eine halbe Phase zueinander verschoben. Dadurch ist mit Empfang eines Signals die Rotationsrichtung bekannt. Ausgehend von der Position $0,09^\circ$ (grüne, vertikale Linie) bedeutet ein Signal auf Kanal A eine Vorwärtsrotation, während ein Signal auf Kanal B eine Rückwärtsrotation angibt.

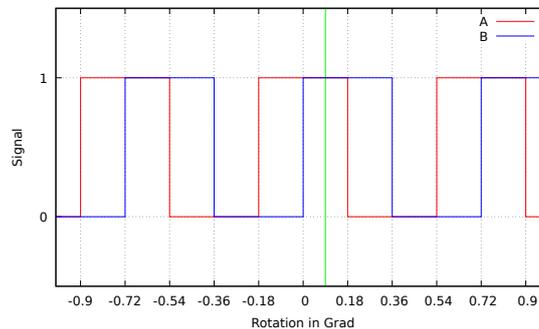


Abbildung 5.16: Signale des Inkrementaldrehgebers

Der Controller merkt sich den Kanalzustand. Wird ein Interrupt ausgelöst, dann ändert sich der Zustand des Kanals, der den Interrupt ausgelöst hat. Am geänderten Kanal wird abgeleitet, in welche Richtung sich der Inkrementaldrehgeber bewegt hat. Pro ausgelöstem Interrupt wird die Position k inkrementiert oder dekrementiert, abhängig von der Rotationsrichtung.

Das System benötigt die Linear- und Rotationsgeschwindigkeiten, um den Kurs der Berechnungsmaschine vorherzusagen. Die Änderung der Position des Inkrementaldrehgebers ist die Geschwindigkeit v_{enc} , mit der sich die Haspel dreht. Aus dieser können, zusammen mit weiteren Sensoren der Odometrie, die benötigten Geschwindigkeiten berechnet werden.

Innerhalb einer Interruptbehandlungsroutine kann die Geschwindigkeit v_{enc} nicht berechnet werden, da die Routine mit einer viel zu hohen Rate ausgelöst wird und nur wenig Programmcode enthalten darf, um die Ausführung anderer Routinen nicht zu blockieren. Stattdessen betrachtet eine Routine mit einer Rate von 30 Hz die aktuelle Position k des Inkrementaldrehgebers, bestimmt die Änderung Δk und berechnet die Geschwindigkeit. Da die Zeit zwischen der Änderungsbetrachtung konstant bzw. bekannt ist, kann die Geschwindigkeit v_{enc} nach Gleichung 5.22 berechnet werden.

$$\begin{aligned} v_{enc} &= \frac{s}{t} = \frac{\Delta k * r}{\frac{1}{f}} = \Delta k * r * f \\ &= \frac{\Delta k * \pi * 0,06 \text{ m} * 30 \text{ Hz}}{2000} \end{aligned} \quad (5.22)$$

Lineargeschwindigkeit der Berechnungsmaschine Der Inkrementaldrehgeber misst die Geschwindigkeit v_{enc} am äußeren Rand der Haspel. Der Schlauch ist jedoch im Haspelinneren aufgewickelt und hat damit einen geringeren Umfang U_{tube} . Ein weiterer Faktor u berücksichtigt die Reduktion des Umfangs u nach Gleichung 5.23.

$$u = \frac{U_{tube}}{U_{haspel}} \quad (5.23)$$

U_{tube} entspricht dem durchschnittlichen Schlauchumfang. Wird eine genauere Odometrie erforderlich, dann kann dieser Faktor mit der Länge des abgewickelten Schlauchs adaptiert werden. Die Geschwindigkeit v , mit der der Schlauch ab- oder aufgewickelt wird, wird nach folgender Gleichung 5.24 berechnet.

$$v = \frac{\Delta k * \pi * u * 6 \text{ m}}{10000 \text{ s}} \quad (5.24)$$

Winkelgeschwindigkeit der Berechnungsmaschine Die Gleichung 5.6 aus dem Abschnitt 5.1.2 wird mit der nach Abschnitt 5.3.2, Paragraf Berechnung der Kolbenposition ermittelten Kolbenposition der Lenkung und der Lineargeschwindigkeit der Berechnungsmaschine genutzt, um die Winkelgeschwindigkeit zu bestimmen. Die übrigen Variablen der Gleichung 5.6 sind fest und werden daher durch eine Konfigurationsdatei gesetzt.

5.4.2 RTK

Aufgrund der Anforderung CR27.1 muss die Lokalisierung der Berechnungsmaschine auf mindestens 15 cm genau erfolgen. Die derzeit verfügbaren GNSS erlauben nur eine Lokalisierungsgenauigkeit im Meterbereich. Beim DGPS-Verfahren wird ein GNSS-Empfänger an einer bekannten Position platziert. Somit ist dieser in der Lage, die Abweichung zwischen der über GNSS ermittelten Position und der bekannten Position zu berechnen. Die

Abweichung wird an einen anderen GNSS-Empfänger übertragen und als Korrektur auf die von ihm ermittelte Position angewendet. Störungen, die zu Ungenauigkeiten führen, treten vor allem in der Atmosphäre auf. Solange beide GNSS-Empfänger eine Lokalität von ca. 100 km aufweisen, sind die atmosphärischen Störungen ähnlich. Durch die angewendete Korrektur kann mit dem DGPS-Verfahren eine Genauigkeit im Dezimeterbereich erreicht werden. Das RTK-Verfahren baut auf dem DGPS-Verfahren auf und wertet neben den GNSS-Signalen auch deren Trägersignal aus. Es wird die Anzahl der vollständigen Wellenlängen, welche das Trägersignal zwischen Satellit und Empfänger benötigt, mit der bekannten Wellenlänge des Trägersignals multipliziert und der Wellenrestanteil addiert [48]. Das Verfahren erreicht eine Genauigkeit im Zentimeterbereich und erfüllt so die Anforderung CR27.1.

Es existiert bereits die Infrastruktur zum Betrieb von RTK-Empfängern und somit erfordert es keine Instrumentalisierung der Berechnungsfläche. Andere auf künstlichen Landmarken basierende, absolute Lokalisierungsverfahren, wie zum Beispiel Bluetooth-Beacons, würden entgegen der Anforderung PCR5 eine Instrumentalisierung der Berechnungsfläche benötigen. RTK-Controller benötigen lediglich ein Korrektursignal, um die Position berechnen zu können. Auf natürliche Landmarken basierende, absolute Lokalisierungsverfahren benötigen viel Rechenleistung und weitere Sensoren, wie zum Beispiel Laserscanner. Auch wären vorliegende Fahrspuren in Form von Shape-Dateien nicht mit absoluten Lokalisierungsverfahren, die auf natürlichen Landmarken basieren, kompatibel.

Die Einbindung des RTK-Controller C94-M8P vom Hersteller u-blox in das System erfolgt durch das *ublox*-Paket [49]. Dieses kommuniziert über eine serielle Schnittstelle mit dem RTK-Controller und konfiguriert diesen anhand gesetzter Parameter. Das Paket stellt außerdem eine Schnittstelle bereit, über die die aktuelle Koordinate mitgeteilt wird.

Korrektursignal Das RTK-Verfahren basiert auf dem DGPS-Verfahren und benötigt daher ein Korrektursignal von einer Referenzstation, die eine Lokalität von ca. 100 km aufweist. DGPS- und RTK-Korrektursignale werden in Form von RTCM-Nachrichten von NTRIP-Servern versendet. Mit einem NTRIP-Client werden die RTCM-Nachrichten empfangen und an den jeweiligen DGPS- bzw. RTK-Controller übertragen.

Es gibt Anbieter, wie den (), die sich auf die Bereitstellung von Korrektursignalen spezialisiert haben. Diese betreiben Referenzstationen und bieten die Korrektursignale über das Internet an. Im Verlauf des Projektes wurde vom niedersächsischen Ministerium für Ernährung und Verbraucherschutz eine Pressemitteilung herausgegeben, in der über eine kostenlose Nutzung der SAPOS-Dienste ab den 01.10.2019 informiert wurde [50]. Die Niederlassung des SAPOS in Niedersachsen betreibt mehrere Referenzstationen, sodass innerhalb von Niedersachsen zwischen RTK-Empfänger und Referenzstation eine Distanz unter 100 km vorliegt. Aus diesen Gründen wird ein Referenzsignal der niedersächsischen Niederlassung des SAPOS verwendet.

Der Empfang der Korrekturdaten vom NTRIP-Server des SAPOS erfolgt durch einen ent-

wickelten NTRIP-Client über ein Mobilfunknetz. Der Client übergibt die über GNSS empfangenen Koordinaten bei der Anmeldung an den Server. Ausgehend von den Koordinaten wählt der NTRIP-Server das Korrektursignal einer in der Nähe liegenden Referenzstation aus und überträgt dieses an den Client. Dem System stellt der Client die Korrektursignale über eine Schnittstelle zur Verfügung. Die konfigurierbare Adresse des NTRIP-Servers ermöglicht den leichten Wechsel zu einem anderen NTRIP-Anbieter.

Das *ublox*-Paket unterstützt nicht den Empfang von RTCM-Nachrichten. Daher wurde das Paket um diese Möglichkeit ergänzt. Sobald eine Schnittstelle existiert, die RTCM-Nachrichten zur Verfügung stellt, leitet das angepasste *ublox*-Paket diese über die serielle Schnittstelle weiter an den RTK-Controller.

Dual RTK Es werden zwei RTK-Controller verwendet, um neben der Position auch die Ausrichtung zu bestimmen (siehe Abbildung 5.17). Die GNSS-Antennen der Controller sind möglichst weit voneinander entfernt, damit sich etwaige Lokalisierungsfehler nicht zu stark auf die bestimmte Ausrichtung auswirken. Ein zu großer Lokalisierungsfehler führt nach Anforderung CR29 zum Stopp der Berechnungsmaschine und muss daher bei der Ausrichtungsbestimmung nicht berücksichtigt werden.

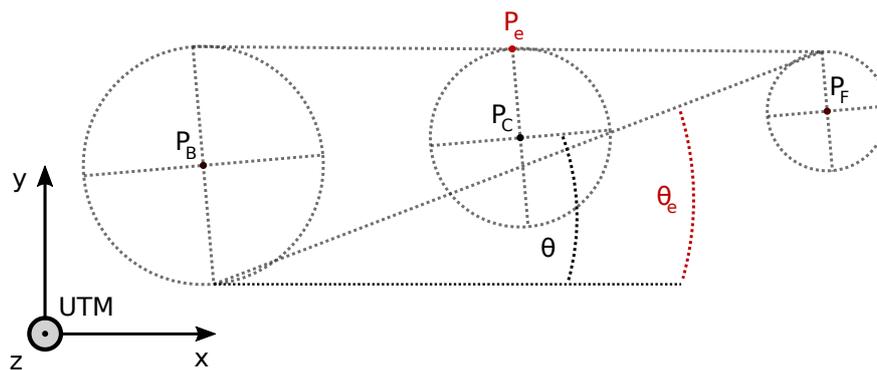


Abbildung 5.17: Geometrie der RTK-Koordinaten

Um dem System eine Pose bereitzustellen, werden die ermittelten Koordinaten vom hinteren RTK-Controllers mit denen des vorderen RTK-Controllers kombiniert. In Abbildung 5.17 sind die beiden ermittelten Koordinaten P_B und P_F dargestellt. Der Mittelpunkt P_C bildet die kombinierte Koordinate. Um die jeweiligen Punkte repräsentieren Kreise deren Standardabweichung. Für die kombinierte Koordinate P_C wird die Standardabweichung durch den Abstand zum Punkt P_e bestimmt. Die Ausrichtung θ entspricht der Steigung der Geraden im *UTM*-Koordinatensystem, die durch die Koordinaten P_B , P_F und P_C verläuft. Als Standardabweichung für θ wird dessen Differenz zu θ_e genutzt. Wobei θ_e den Winkel beschreibt, welcher unter Beachtung der Standardabweichungen von P_B und P_F die größte Distanz zu θ aufweist.

Ausgehend von einer Distanz von 2,5 m zwischen beiden GNSS-Antennen und einer Standardabweichung von 2 cm für die vordere sowie hintere Koordinate, ergibt sich eine Standardabweichung für die kombinierte Koordinate von 2 cm und für die Ausrichtung von ca. 0,016 rad. Wie in Abbildung 5.17 zu erkennen, ist die Standardabweichung der kombinier-

ten Koordinate immer kleiner als die größere Standardabweichung aus P_B und P_F .

Anfängliche Versuche, die Ausrichtung durch einen IMU mit integrierten Magnetometer zu bestimmen, blieben ohne Erfolg. Die vom Sensor erfassten Werte wiesen bei einer nahezu konstanten Winkelgeschwindigkeit bis zu 0,157 rad große Sprünge auf. Dies führte zu einem oszillierenden Verhalten bei der Abfahrt eines Berechnungspfades. Die Lösung mit zwei RTK-Controllern weist kleinere Sprünge auf und muss im Gegensatz zu einer IMU nicht kalibriert werden.

5.4.3 Sensorfusion

Um die berechneten Geschwindigkeiten der Odometrie 5.4.1 und die Posen aus der absoluten Lokalisierung 5.4.2 zu einer Pose zusammenzuführen, wird ein EKF verwendet. Dieser durchläuft in einer geschlossenen Schleife die Schritte: Vorhersagen und Korrigieren. Im ersten Schritt wird anhand eines Modells das Verhalten vorhergesagt. Der zweite Schritt korrigiert dieses mit Messdaten. Die Korrektur beachtet dabei die Varianzen der vorhergesagten Werte und die der Messungen. Mit jeder Vorhersage erhöhen sich die Varianzen der vorhergesagten Werte. Der EKF benötigt, im Gegensatz zu anderen Sensorfusionen, nur eine geringe Rechenleistung und wurde bereits in vielen Anwendungen erfolgreich eingesetzt [51, S. 53].

Das eingesetzte *robot_localization*-Paket [52] implementiert einen flexibel parametrierbaren und ROS kompatiblen EKF. Dieser ist so konfiguriert, dass er die Transformation zwischen dem *Map*-Koordinatensystem und dem *Base Link*-Koordinatensystem erzeugt. Die Parametrierung der Varianzen für Vorhersageschritte erfolgt über eine Matrix. Deren Werte sind empirisch anhand von Simulationen bestimmt worden. Die Simulationen basieren hierbei auf Aufzeichnungen von Sensordaten. Das Abspielen dieser erlaubt, zusammen mit einer Visualisierung, das Verhalten in der Simulation zu bewerten und die Matrix iterativ anzupassen.

Auch implementiert das *robot_localization*-Paket eine Komponente, die zur Transformation von WGS84-Koordinaten in das *UTM*-Koordinatensystem verwendet wird. WGS84-Koordinaten entsprechen weder der REP 103, noch der REP 105 und können somit nicht vom *tf*-Paket verarbeitet werden. Die Komponente aus dem *robot_localization*-Paket rechnet die WGS84-Koordinaten in ()-Koordinaten um. Außerdem ergänzt die Komponente die Baumstruktur des *tf*-Paket um eine Transformation vom *Map*-Koordinatensystem in das *UTM*-Koordinatensystem.

5.4.4 Lageerkennung

Nach Anforderung CR30 muss die Berechnungsmaschine erkennen, wenn sie sich zu stark neigt. Dafür ist ein Lagesensor erforderlich. Die BNO055-IMU vom Hersteller Bosch ist nach dessen Datenblatt [53] ein ganzes System aus Sensoren und einem Chip, um die Messungen aufzubereiten. Dieses kann die eigene Ausrichtung absolut entlang allen drei Achsen erfassen, hat sich aber im Verlauf des Projekts als zu unpräzise bei der absoluten Ausrichtungserfassung auf der gegebenen Berechnungsmaschine herausgestellt. Dennoch verfügt es

über ein Beschleunigungssensor, welcher sich eignet, um den Neigungswinkel der Maschine zu bestimmen. Zur Neigungserkennung ist jeder Beschleunigungssensor geeignet. Der Neigungswinkel ist definiert als Winkel zwischen einem senkrecht auf der Beregnungsmaschine stehenden Vektor und dem negativen Erdanziehungsvektor. Er muss nicht präzise gemessen werden, da nur starke Neigungen erkannt werden müssen.

Der Beschleunigungssensor wird über eine I2C-Schnittstelle an den Controller angebunden. Um den Datenbus gegenüber Störungen abzusichern, werden die beiden Datenleitungen mit vier $4,7\text{k}\Omega$ -Widerständen auf $3,3\text{V}$ gezogen. Die Widerstände werden, wie in Abbildung 5.18 dargestellt, möglichst nahe am Controller und Sensor verlötet. So können die Datenleitungen des I2C-Bus auf bis zu 30 cm verlängert werden, ohne dass die Signalqualität leidet.

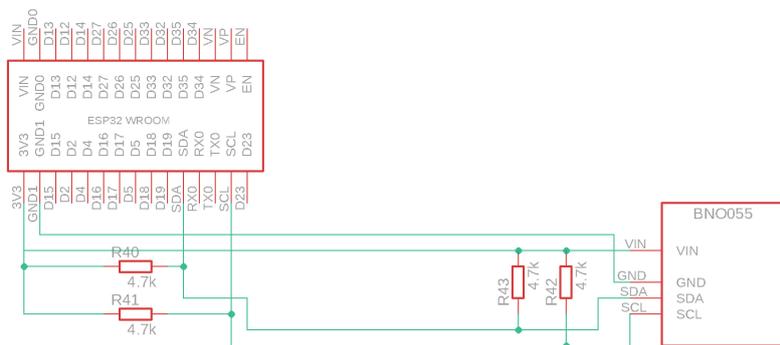


Abbildung 5.18: Anbindung des Beschleunigungssensors

Aufbereiten der Messung Ein Beschleunigungssensor misst nicht die tatsächlichen Beschleunigungen. Die Messwerte weichen, neben unvermeidbarem Rauschen, um einen festen Betrag von der Wirklichkeit ab. Diese Abweichung kann durch Fertigung, Abweichung im Material oder durch Einflüsse der Lokalität des Sensors entstehen. Durch Kalibrierungsroutinen wird ein fester Betrag, Offset genannt, näherungsweise bestimmt und fortan auf alle Messungen addiert. Der BNO055-IMU hat diese Kalibrierungsroutinen bereits integriert und führt diese beim Start aus. Die ermittelten Offsets werden vom Sensor selbst auf die Messungen addiert und können gelesen werden, sobald die Kalibrierung abgeschlossen ist. Solange die Kalibrierung nicht abgeschlossen ist, liefert der Sensor keine guten Werte.

Um eine gute Kalibrierung des Beschleunigungssensors zu erreichen, muss dieser nach [53, S. 48] in sechs unterschiedliche, stabile Positionen gebracht werden. Es ist nicht möglich diese Schritte bei jedem Start der Beregnungsmaschine auszuführen. Stattdessen wird, wie in Abbildung 5.19 dargestellt, der Sensor im nicht-montieren Zustand kalibriert, die ermittelten Offsets auf dem Controller gespeichert und bei jedem Startvorgang in den Sensor geschrieben. Beim Starten erkennt der Controller, ob bereits Offsets hinterlegt wurden, indem er überprüft, ob die Sensor-ID an der zugehörigen Speicherstelle steht. Steht hier ein anderer Wert, wurden noch keine Offsets hinterlegt. Nachdem die Offsets einmal hinterlegt wurden, muss keine manuelle Kalibrierung mehr ausgeführt werden.

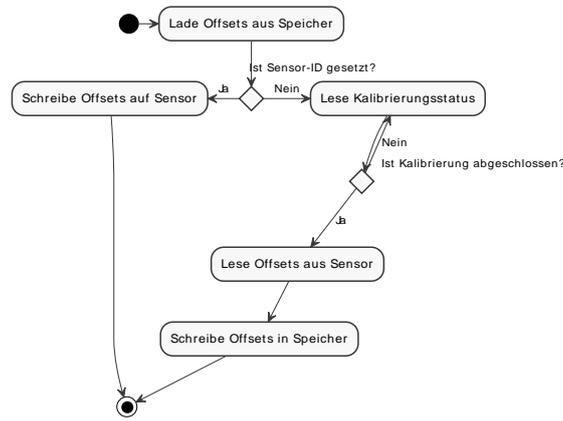


Abbildung 5.19: Lesen und Schreiben der BNO055-Sensoroffsets

Auswertung der Messung Der Beschleunigungssensor misst auf seiner z-Achse die Erdbeschleunigung. Bei einem Neigungswinkel von 0° sollte dieser einen Wert von ungefähr $9,81 \frac{m}{s^2}$ messen. Liegt die Berechnungsmaschine auf der Seite, dann hat sie einen Neigungswinkel von 90° und der Beschleunigungssensor misst auf der nun geneigten z-Achse einen Wert von ungefähr $0 \frac{m}{s^2}$. Durch diese Punkte wird eine Gerade gezogen, welche durch die Funktion 5.25 beschrieben und in der Abbildung 5.20 dargestellt wird.

$$f(x) = \frac{-90}{9,81}x + 90 \quad (5.25)$$

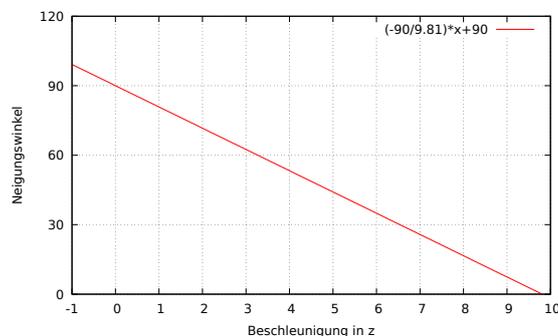


Abbildung 5.20: Funktion zur Berechnung des Neigungswinkels aus der Beschleunigung entlang der z-Achse der Berechnungsmaschine

Diese Funktion ist nur eine Approximation. Sie liefert einen negativen Neigungswinkel für eine Beschleunigung größer als $9,81 \frac{m}{s^2}$, welcher per Definition nicht existieren kann. Der Neigungswinkel wird aber gut genug bestimmt, um ein Umkippen festzustellen. Die Berechnungsmaschine gilt ab einem Winkel von 30° als umgekippt. Bei diesem Winkel ist die Berechnungsmaschine nicht mehr in der Lage Fahrmanöver auszuführen.

Es wäre ineffizient, den Neigungswinkel bei jeder Messung erneut zu berechnen. Stattdessen wird einmalig die Beschleunigung berechnet, welche bei einem kritischen Neigungswinkel auftritt und die Messungen mit dieser verglichen. Die Grenze wird mit der Funktion 5.26 bestimmt.

$$f(x) = \frac{-9,81}{90}x + 9,81 \quad (5.26)$$

Nach dieser gilt die Beregnungsmaschine ab einer gemessenen Beschleunigung kleiner $6,54 \frac{m}{s^2}$ als umgekippt. In diesem Fall wird im Controller ein Fehlerereignis ausgelöst und die Beregnungsmaschine entsprechend des Abschnitts 5.1.4 gestoppt. Damit wird die Anforderung CR31 erfüllt.

5.5 Kartierung

Die Kartierung von Fahrspuren erfolgt mit dem Kartierungsgerät, welches ein Kartiersystem gemäß der Anforderung CR20 bildet. Dieses ist mit einem RTK-Empfänger ausgestattet, sodass es sich selbst lokalisieren kann. Um die benötigten Korrektursignale für das RTK-Verfahren zu empfangen, ist es außerdem mit einem LTE-Modem verbunden. Für den Betrieb des RTK werden die entsprechenden Software-Komponenten der Beregnungsmaschine eingesetzt.

Die Bedienung des Kartiergerätes erfolgt über eine Webseite, die über dessen WLAN-Access-Point aufgerufen wird. Die Bedienoberfläche verbindet sich über das Web Socket Protokoll mit dem Kartierungsgerät und kann so Aktionen anstoßen und Informationen abrufen. Die in Abbildungen 5.21 und 5.22 sind Bildschirmaufnahmen der Bedienoberfläche des Kartierungsgerätes und zeigen die Kartierungsoberfläche und die Kartenliste.

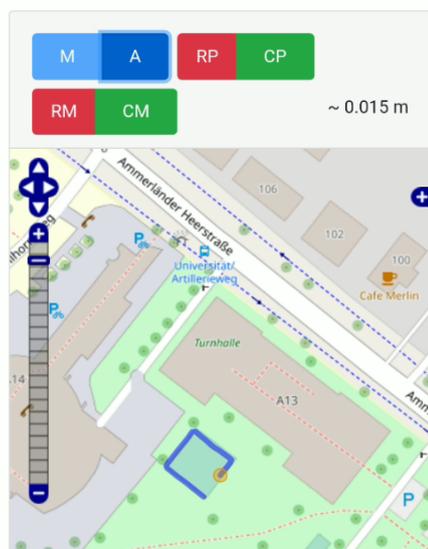


Abbildung 5.21: Bildschirmaufnahme der Kartierungsoberfläche

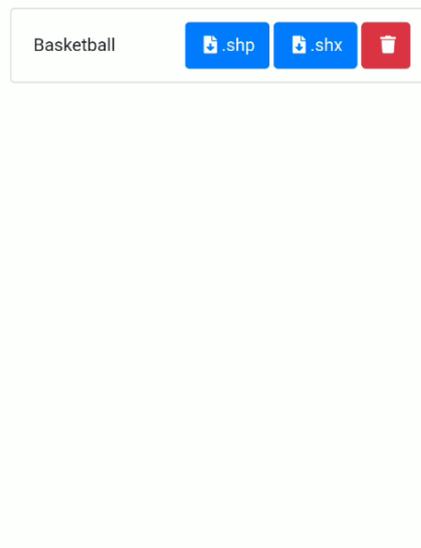


Abbildung 5.22: Bildschirmaufnahme der Kartenliste

Über die Kartierungsoberfläche kann ein Bediener mehrere Pfade kartieren und in einer Shape-Datei zusammenfassen. Sie zeigt dem Bediener die Standardabweichung der aktuellen Koordinate an, damit dieser die Qualität der Daten einschätzen kann. Bei der Kartierung hat der Bediener die Wahl, ob er die Koordinaten einzelnen erfassen will oder ob diese automatisch durch eine Aufzeichnung erfasst werden. Die erfassten Koordinaten werden in einer Ebene über einer Karte angezeigt und mit blauen Linien in ihrer Erfassungsreihenfolge verbunden. Dadurch kann der Bediener einschätzen, wie gut der jeweilige Pfad kartiert wurde. Sobald alle Koordinaten eines Pfades erfasst wurden, kann der Bediener dies über

die Oberfläche bestätigen und somit das Ende eines Pfades entsprechend der Anforderung CR20.4.1 markieren. Die gesammelten Koordinaten werden in einem Pfad zusammengefasst und in die aktuelle Karte übernommen. Als Rückmeldung werden alle Pfade, die übernommen wurden, in der Karte durch rote Linien dargestellt. Falls ein Pfad nicht übernommen werden soll, kann der aktuelle Pfad zurückgesetzt werden. Nachdem alle Pfade kartiert wurden, kann der Bediener diese bestätigen und es wird eine Shape-Datei erzeugt, welche die kartierten Pfade durch Polygonzüge beschreibt. Die Shape-Dateien können über die Kartenliste der Bedienoberfläche, nach Anforderung CR20.4.2, heruntergeladen werden.

Ein Clustering-Verfahren und eine anschließende Glättung reduzieren die Anzahl der Koordinaten, die bei der Kartierung erfasst werden in Realzeit. Dies stellt sicher, dass der Bediener das Ergebnis zu jedem Zeitpunkt sieht und gegebenenfalls den Pfad zurücksetzen kann. Mit dem Clustering-Verfahren werden doppelt erfasste Koordinaten entfernt, die auftreten, wenn der Kartierende sich während der automatischen Aufzeichnung nicht bewegt. Das Verfahren beachtet dabei die Reihenfolge der Koordinaten und fasst jene zusammen, die aufeinander folgen. Um jede Koordinate wird ein Radius der Größe der Standardabweichung gezogen. Zusätzlich kann die Größe des Radius durch einen parametrierbaren Faktor und einen parametrierbaren Summand über die Bedienoberfläche angepasst werden. Letzteres kann verwendet werden, wenn die RTK-Antenne an einem erhöhten Ort liegt und Schwankungen unterworfen ist. Liegt eine Koordinate im Radius einer anderen, werden die Koordinaten im Zentrum vereint. Die anschließende Glättung erfolgt nach dem Douglas-Peucker-Algorithmus. Bei diesem werden ebenfalls die Standardabweichungen und die zuvor genannten Parameter genutzt, um den Grad der Glättung zu bestimmen. Kleine Lokalisierungsfehler werden von der Glättung entfernt und so nicht im Pfad beachtet.

5.6 Berechnungspfad folgen

Das Folgen eines Berechnungspfades ist eine Teilaufgabe der automatisierten Abfahrt einer Berechnungssequenz und somit notwendig, um die Anforderung CR26 zu erfüllen. Berechnungspfade sind nach Definition 1 durch eine Liste von Koordinaten, die entsprechend ihrer Abfahrreihenfolge sortiert sind, beschrieben. Aufgrund der Anforderung UR2.3 liegen Berechnungssequenzen in Form von Shape-Dateien vor und setzen sich aus durch Polygonzüge beschriebenen Berechnungspfaden zusammen. Die Aufgabe bei der Verfolgung eines Berechnungspfades liegt darin, dass die Berechnungsmaschine so gesteuert wird, dass diese möglichst genau diesen Polygonzügen folgt.

Definition von Berechnungssegmenten Die Definition von Berechnungspfaden 1 behandelt nicht die nach den Anforderungen UR35.1 und UR28.1 geforderte Parametrisierung der präferierten Geschwindigkeit und der Mindestgenauigkeit, mit der einem Berechnungspfad gefolgt werden soll. Da entlang eines Berechnungspfades unterschiedliche Gegebenheiten vorliegen können, ist es sinnvoll, diesen zu unterteilen. Auf den Teilstücken können dann die Parameter, entsprechend des jeweiligen Umfelds, gewählt werden. Mit der Definition 6 von Berechnungssegmenten wird dies erreicht. Sie erlauben die Parametrisierung von präferierter Geschwindigkeit und maximaler Abweichung auf Segmenten,

welche durch zwei aufeinanderfolgende Koordinaten beschrieben werden. Wenn zwei aufeinanderfolgende Koordinaten zu weit auseinander liegen und sich das dazwischenliegende Segment über unterschiedliche Gegebenheiten erstreckt, kann ohne Verletzung der Definition 1 ein Berechnungspfad um weitere Koordinaten ergänzt werden. Somit entstehen neue Berechnungssegmente welche jeweils unterschiedlich parametrisiert werden können.

Definition 6 *Berechnungssegment*

Ein **Berechnungssegment** $bseg$ ist ein Quadrupel mit $bseg = (k_{start}, k_{end}, v_{pref}, \delta_{max})$, v_{pref} ist eine präferierte Geschwindigkeit, δ_{max} ist die maximale Abweichung, k_{start} sowie k_{end} sind Koordinaten aus einem Berechnungspfad $bpf = \langle k_1, \dots, k_n \rangle$ und $n \in \mathbb{N}$, sodass $k_{start} = k_i$ und $k_{end} = k_{i+1}$ mit $i \in [1, n - 1]$ gelten.

Model Prediction Regelung Bei der Erstellung eines Berechnungspfades werden keine Bewegungseinschränkungen der Berechnungsmaschine berücksichtigt. Daher ist nicht sichergestellt, dass die Berechnungsmaschine diesen ohne Abweichungen abfahren kann. Durch eine Vorverarbeitung der Berechnungspfade könnte dies zwar sichergestellt werden, jedoch würde es diesen verfälschen. Auch könnten weiterhin Lokalisierungsfehler dazu führen, dass der vorverarbeitete Berechnungspfad nicht mehr eingehalten werden kann. Übliche Regelsysteme nach [19], welche zur Pfadverfolgung eingesetzt werden, setzen einen durch eine oder mehrere Kurven beschriebenen Pfad voraus oder schneiden Ecken durch frühzeitiges Lenken auf der Innenseite mit entsprechenden Abweichungen ab. Ein Algorithmus zur Einhaltung eines Berechnungspfades muss jedoch Polygone, welche Ecken aufweisen können, mit möglichst kleiner Abweichung folgen. Ein Lenkmanöver muss daher so erfolgen, dass ein Kompromiss zwischen innenliegender Abweichung und außenliegender Abweichung gefunden wird, wenn diese unvermeidbar ist. So verteilt sich die Abweichung auf beide Seiten und die maximale Abweichung zum Berechnungspfad wird reduziert. Ein geeignetes Verfahren ist die Model Prediction Regelung, die in einer geschlossenen Schleife realisiert wird. Gegenüber üblichen Regelsystemen vergleicht sie nicht den Ist- mit dem Soll-Zustand, sondern trifft auf Basis eines Vorhersagemodells Aussagen über die Regelabweichung über die Zeit und agiert dieser entsprechend. Die wesentlichen Schritte der Model Prediction Regelung sind in Abbildung 5.23 dargestellt.

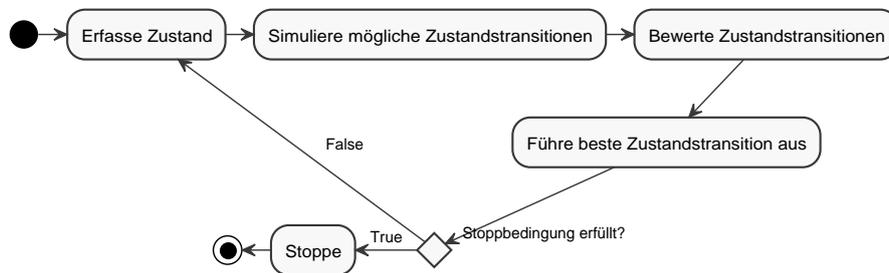


Abbildung 5.23: Aktivitätsdiagramm einer Model Prediction Regelung

Zu Beginn eines Zyklus erfasst die Regelung den Ist-Zustand des Systems. Darauf aufbauend werden mögliche Zustandstransitionen durch ein Vorhersagemodell simuliert. Eine mögliche Zustandstransition wäre die Änderung der Pose. Anschließend werden die simulierten Zustandstransitionen nach ihrer Regelabweichung über die Zeit bewertet. Die Zustands-

transition mit der kleinsten Regelabweichung über die Zeit wird am besten bewertet und ausgeführt. Wenn die Stoppbedingung nicht erfüllt ist, beginnt der Zyklus erneut.

Algorithmus zur Einhaltung eines Berechnungspfades Der Algorithmus zur Verfolgung eines Berechnungspfades implementiert den Zyklus einer Model Prediction Regelung und ergänzt diesen den Anforderungen entsprechend um weitere Schritte. In Abbildung 5.24 sind diese in einem Aktivitätsdiagramm dargestellt. Der Zyklus beginnt mit der Erfassung des Zustands. Dieser umfasst die globale Pose und ihre Standardabweichung, die lokale Geschwindigkeit und die Stellung sowie die Geschwindigkeit des Linearantriebes der Lenkung.

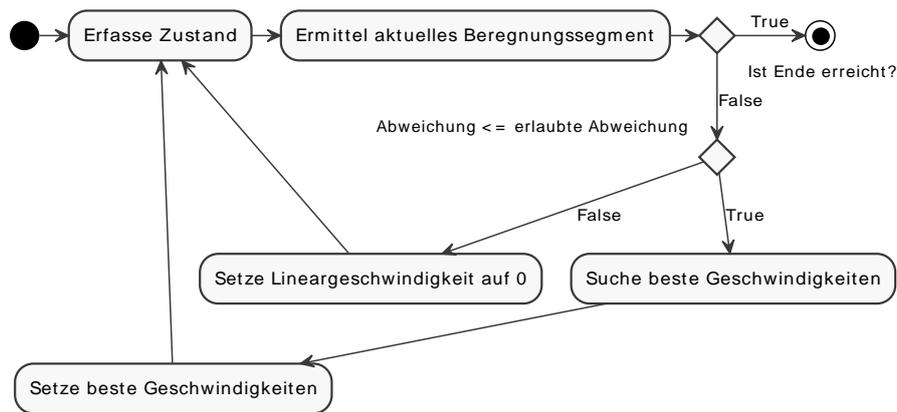


Abbildung 5.24: Aktivitätsdiagramm der Berechnungspfadverfolgung

Anhand der Pose der Berechnungsmaschine und dem im vorherigen Zyklus bestimmten Berechnungssegment, wird das aktuelle Berechnungssegment ermittelt. Der Punkt auf dem aktuellen Berechnungssegment, welcher der Position der Berechnungsmaschine am nächsten liegt, wird als Referenzpunkt gesetzt und bildet die Referenzposition auf dem Berechnungspfad.

Wenn es sich bei dem aktuellen Berechnungssegment um das letzte in einem Berechnungspfad handelt, wird die Distanz zwischen dem Referenzpunkt und dem Endpunkt des Berechnungssegments berechnet. Ist diese kleiner als ein parametrierbarer Wert, gilt der Berechnungspfad als abgefahren. Dann ist die Stoppbedingung erfüllt und der Algorithmus endet, nachdem die Lineargeschwindigkeit auf $0 \frac{m}{s}$ gesetzt wurde. Falls die Distanz größer als der parametrierbarer Wert ist, wird der Zyklus weiter durchlaufen.

Um die Anforderung CR29 zu erfüllen, wird zunächst die Abweichung zwischen dem aktuellen Berechnungssegment und der Berechnungsmaschine bestimmt. Dafür wird die Distanz zwischen dem Referenzpunkt und der Position der Berechnungsmaschine berechnet und um die Standardabweichung der Position ergänzt. Das aktuelle Berechnungssegment gibt nach Definition 6 die erlaubte Abweichung vor. Ist diese kleiner als die Abweichung zwischen der Berechnungsmaschine und dem Berechnungssegment, dann wird die Lineargeschwindigkeit auf $0 \frac{m}{s}$ gesetzt und der Zyklus beginnt erneut. So wird verhindert, dass sich die Berechnungsmaschine zu weit von einem Berechnungspfad entfernt. Die Betrachtung der Standard-

abweichung der Pose stellt sicher, dass ein zu großer Lokalisierungsfehler dazu führt, dass der Schwellwert überschritten wird. Verringert sich der Lokalisierungsfehler wieder, wird der Schwellwert wiederum unterschritten und die Berechnungsmaschine setzt die Berechnung fort.

Der Ablauf der Suche nach den besten Geschwindigkeiten wird in der Abbildung 5.25 dargestellt. Es gilt die beste Kombination aus Lineargeschwindigkeit und Winkelgeschwindigkeit zu finden, sodass über die Zeit die Abweichung zwischen Berechnungsmaschine und Berechnungspfad möglichst gering gehalten wird.

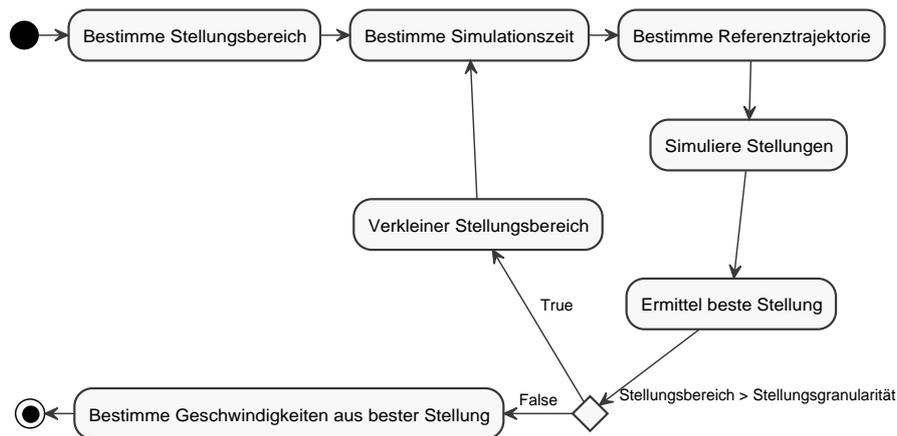


Abbildung 5.25: Aktivitätsdiagramm der Suche nach optimalen Geschwindigkeiten

Die Suche bestimmt zuerst jeweils einen Stellungsbereich für den Antrieb und für die Lenkung. Die Stellungsgebiete geben an, welche Lineargeschwindigkeiten und Lenkpositionen möglich sind. Beide Bereiche sind durch Parameter konfigurierbar, wobei der Stellungsgebiet des Antriebes durch die dynamisch ermittelten Lineargeschwindigkeitsgrenzen beschrieben werden kann.

Ein Fahrzeug muss vorausschauend manövriert werden, denn die Menge der möglichen Fahrmanöver ist durch die Agilität des Fahrzeugs beschränkt. Enge Kurven erfordern daher eine angemessene Voraussicht. Die Simulationszeit bestimmt, wie vorausschauend der Algorithmus operiert. Sie wird durch benötigte Stellzeiten (vgl. Agilität des Fahrzeugs) und den auftretenden Winkeln, auf den noch nicht abgefahrenen Teilen des Berechnungspfad (vgl. Kurven), bestimmt.

Eine Kombination der Stellungsgebiete bildet einen zweidimensionalen Stellungsraum. Aus diesem werden pro Dimension parametrierbar viele, gleich verteilte Stellungskombinationen ausgewählt. Deren Annahme wird ausgehend vom Ist-Zustand über die Dauer der Simulationszeit simuliert. Die Simulation erfolgt je Stellungskombination iterativ über die Zeit und umfasst neben den Stellungen auch die Pose der Berechnungsmaschine. Daraus ergeben sich zeitlich geordnete Zwischenzustände, welche zusammen eine Trajektorie, den Fahrweg der Berechnungsmaschine, über die Simulationszeit beschreiben.

Die Trajektorien werden im weiteren Verlauf gegenüber einer Referenztrajektorie bewertet. Diese bildet einen Teil des Berechnungspfades im zeitlichen Bereich ab und schafft somit eine Vergleichbarkeit zwischen Trajektorien und Berechnungspfad. Ähnlich zur Arbeit [54] wird ein virtuelles Referenzfahrzeug geschaffen. Dieses ist vollständig frei in der Bewegungsmöglichkeit und unterliegt keinen physikalischen Einschränkungen. Somit kann es jeden Berechnungspfad ohne Abweichung folgen. Die initiale Pose des Referenzfahrzeugs entspricht dem Referenzpunkt auf dem Berechnungspfad und der Ausrichtung des aktuellen Berechnungssegments. Eine Simulation beginnt bei der initialen Pose und folgt dem Berechnungspfad für eine der Simulationszeit und der Geschwindigkeit entsprechenden Strecke. Die initiale Pose, die Pose am Ende der Simulationszeit und die Posen am Ende eines jeden Berechnungssegments bilden zusammen mit den entsprechenden Zeitpunkten innerhalb der Simulation die Referenztrajektorie. Während der Simulation nimmt das Referenzfahrzeug die jeweils präferierte Geschwindigkeit an. Dies führt bei einem späteren Vergleich mit anderen Trajektorien dazu, dass jene Trajektorien besser bewertet werden, deren Geschwindigkeit zur Referenztrajektorie ähnlich ist. Dadurch nimmt die Berechnungsmaschine eine zur präferierten Geschwindigkeit ähnliche Geschwindigkeit an und erfüllt somit die Anforderung CR36. Wenn das Referenzfahrzeug das Ende des Berechnungspfades erreicht, dann wird für den Rest der Simulationszeit eine Geschwindigkeit von $0 \frac{\text{m}}{\text{s}}$ angenommen. Entsprechend werden die Trajektorien besser bewertet, welche um den jeweiligen Zeitpunkt herum ebenfalls eine Geschwindigkeit von $0 \frac{\text{m}}{\text{s}}$ annehmen. Da Trajektorien unter Berücksichtigung von physikalischen Einschränkungen erstellt werden, führt dies zu einer kontrollierten Entschleunigung der Berechnungsmaschine.

Die Referenztrajektorie sowie die Trajektorien aus der Simulation der Stellungskombinationen sind bezüglich ihrer zeitlichen Dimension entsprechend der Simulationszeit gleich und können somit über diese in Relation gesetzt werden. Daher wird die Abweichung über die Zeit zwischen einer Trajektorie und der Referenztrajektorie als Bewertungskriterium verwendet. Die Stellungskombination, die zur besten Trajektorie führt, gilt als beste Stellungskombination.

Solange einer der Stellungsbereiche noch größer als die ihm zugeordnete, parametrierbare Stellungsgranularität ist, werden beide Bereiche um getrennt parametrierbare Faktoren verkleinert und um die bisher beste Stellung zentriert. In den neuen Bereichen beginnt die Suche nach der besten Stellung erneut. Mit jeder Iteration nähern sich die zur Simulation ausgewählten Stellungskombinationen aneinander an. Sobald beide Stellungsbereiche kleiner oder gleich der jeweiligen Stellungsgranularität sind, wird die beste Stellung aus der letzten Iteration genutzt, um daraus Lineargeschwindigkeit und Winkelgeschwindigkeit zu bestimmen.

Die ermittelte Lineargeschwindigkeit und Winkelgeschwindigkeit werden gesetzt und der Zyklus der Regelung beginnt erneut.

5.6.1 Bestimmung des aktuellen Berechnungssegment

Die Referenztrajektorie wird anhand des aktuellen Berechnungssegments bestimmt. Daher muss für eine korrekte Bewertung der Stellungen das richtige Berechnungssegment ausgewählt werden. Es genügt nicht das Berechnungssegment allein anhand der kleinsten Distanz zur Position der Berechnungsmaschine auszuwählen. Wenn zwei nicht aufeinanderfolgende Berechnungssegmente nah beieinander liegen, dann kann es durch Lokalisierungsfehler dazu kommen, dass das falsche Berechnungssegment eine kleinere Distanz aufweist als das eigentlich aktuelle Berechnungssegment. Daher wird zur Bestimmung des aktuellen Berechnungssegments neben der Position der Berechnungsmaschine auch das im vorherigen Zyklus bestimmte Berechnungssegment herangezogen. Von diesem ausgehend erfolgt eine iterative Vorwärts- und Rückwärtssuche nach dem Berechnungssegment mit der kleinsten Distanz zur Position der Berechnungsmaschine. Wenn die Distanz eines Berechnungssegments zur Berechnungsmaschine kleiner als die des zuvor betrachteten ist, dann wird dieses vorläufig ausgewählt. Sobald ein Berechnungssegment eine größere Distanz als dessen vorheriges aufweist, wird die Suche in die jeweilige Richtung abgebrochen. Nachdem in beide Richtungen die potentiellen Berechnungssegmente bestimmt sind, wird das mit der kleinsten Distanz zur Position der Berechnungsmaschine gewählt.

5.6.2 Berechnung von Stellungsparameter

Die Stellung des Antriebs sowie der Lenkung benötigen Zeit. Somit liegt erst mit einer zeitlichen Verzögerung die jeweils gesetzte Stellung vor. Wenn diese Trägheit bei der Simulation ignoriert wird, dann können erhebliche Abweichungen zwischen Simulation und Realität auftreten. Daher beachtet das genutzte Vorhersagemodell die Trägheit.

Die Modellierung der Trägheit erfolgt auf Basis von parametrierbaren Eigenschaften, die der Stellgröße entsprechen:

- minimale sowie maximale Position,
- minimale sowie maximale Geschwindigkeit und
- minimale sowie maximale Beschleunigung.

Aufgrund der relativ geringen Geschwindigkeiten wird von einer konstanten Beschleunigung ausgegangen. Die Geschwindigkeitsgrenze bzw. die Geschwindigkeitsstellung wird von der Lenkung (siehe Abschnitt 5.3.2, Paragraf Kolbenpositionsregelung) bzw. vom Antrieb schnell erreicht. Somit liegt nur für kurze Zeit eine Beschleunigung an, welche annähernd konstant ist.

Der benötigte Zeitraum zum Setzen einer Stellung wird in Phasen unterteilt. Aufgrund der konstanten Beschleunigung werden die Phasen so gewählt, dass diese jeweils durch einen Beschleunigungswert und eine Beschleunigungsdauer beschrieben sind. Die zur Stellung benötigten Phasen werden in einem Plan nach der jeweiligen Stellgröße zusammengefasst. Wenn eine Phase während einer Stellung nicht benötigt wird, wird deren Beschleunigungsdauer auf 0s gesetzt.

Phasen der Geschwindigkeitsstellung Wenn die aktuelle Geschwindigkeit entgegengesetzt der Stellgeschwindigkeit ist, muss zuerst eine Entschleunigung erfolgen. Diese muss solange anliegen, bis keine Geschwindigkeit mehr vorliegt. Daraufhin kann das Gelenk auf die gewünschte Geschwindigkeit beschleunigt werden. Anschließend muss die Geschwindigkeit bis zur nächsten Stellung gehalten werden. Da der Zeitraum bis zur nächsten Stellung nicht bekannt ist, wird die Dauer auf ∞ gesetzt. Somit ergeben sich für die Geschwindigkeitsstellung drei Phasen: Entschleunigungsphase, Beschleunigungsphase und Bleibephase.

Phasen der Positionsstellung Bei der Stellung einer Position ist neben der Position auch die Geschwindigkeit zu beachten. Wenn sich die Stellposition entgegengesetzt der aktuellen Geschwindigkeit befindet, muss zunächst eine Entschleunigung erfolgen. Dasselbe gilt für den Fall, wenn die aktuelle Geschwindigkeit in Richtung der Stellposition zeigt, jedoch der notwendige Entschleunigungswert, um auf der Position zu stoppen, außerhalb der Beschleunigungsgrenzen liegt. Wenn aktuell keine Geschwindigkeit vorliegt, dann kann die Stellposition direkt durch Beschleunigen, Halten der Geschwindigkeit und Entschleunigen angefahren werden. Dies ist auch der Fall, wenn die aktuelle Geschwindigkeit in Richtung der Stellposition zeigt und der zum Stoppen notwendige Entschleunigungswert innerhalb der Beschleunigungsgrenzen liegt. Jedoch muss die Geschwindigkeitsgrenze auf der Strecke zwischen aktueller Position und Stellposition bestimmt und eingehalten werden. Diese wird durch das Minimum bzw. Maximum der konfigurierten Geschwindigkeitsgrenzen und der minimalen bzw. maximalen Geschwindigkeit, die innerhalb der Beschleunigungsgrenzen und der Streckendistanz erreicht und wieder entschleunigt werden kann, bestimmt. Sobald die Stellposition erreicht ist, wird diese mit einer Dauer von ∞ gehalten. Somit ergeben sich für die Positionsstellung fünf Phasen: Wendephase, Beschleunigungsphase, Haltephase, Entschleunigungsphase und Bleibephase.

Kombination von Geschwindigkeitsstellung und Positionsstellung Die Stellung von Antrieb und Lenkung erfolgt simultan, weshalb die beiden Pläne zu einem kombiniert werden. Hierzu wird jede Phase aus dem Plan zur Stellung des Antriebs mit jeder Phase aus dem Plan zur Stellung der Lenkung vereint. Zusammen bilden zwei Phasen eine kombinierte Phase, welche aus den beiden Beschleunigungswerten der Phasen und einer Beschleunigungsdauer besteht. Die Beschleunigungsdauer der kombinierten Phase entspricht der minimalen Beschleunigungsdauer der beiden Phasen. Nach einer Vereinigung von zwei Phasen wird deren Beschleunigungsdauer um die Beschleunigungsdauer der kombinierten Phase reduziert. Insgesamt bildet die Stellung von Antrieb und Lenkung 15 Phasen.

5.6.3 Berechnung der Simulationszeit

Die Simulationszeit bestimmt, wie vorausschauend der Algorithmus agiert. Wenn diese zu klein gewählt wird, entstehen bei Ecken mit einem größeren Winkel große Abweichungen auf der Außenseite. Ist sie zu groß gewählt, werden Ecken mit einem kleinen Winkel möglicherweise nicht beachtet. Außerdem benötigt eine größere Simulationszeit mehr Rechenressourcen, was zu einer geringen Frequenz der Regelung führt und somit zu einer höheren Reaktionsverzögerung. Um einen situationsbedingten Kompromiss zu finden, wird die Simulationszeit adaptiv bestimmt. Die Einflussgrößen sind die Stellbereiche sowie die

Winkel der Ecken auf dem bisher noch nicht abgefahrenen Teilstück des Berechnungspfades.

Bei der Suche nach der besten Stellung werden erst Stellungen aus den Stellungsbereichen ausgewählt und danach simuliert. Aufgrund der Trägheit benötigen die Stellungen im Regelfall unterschiedliche Stellzeiten. Wenn die Simulationszeit kleiner als die Stellzeit einer simulierten Stellung ist, dann wird die Annahme der Stellung nicht simuliert und die Bewertung dieser verfälscht. Daher muss die Simulationszeit so gewählt werden, dass alle Stellungen aus dem Stellungsbereichen in einer Simulation angenommen werden. Die maximalen und minimalen Stellungen aus den Stellungsbereichen bilden vier Stellungskombinationen, deren maximale Stellzeit die aller anderen Stellungen übertrifft und damit das Minimum für die Simulationszeit setzt.

Um den Einfluss der Ecken auf dem restlichen Teilstück des Berechnungspfades zu bestimmen, werden die Winkel dieser ermittelt. Die Winkel von aufeinanderfolgenden Ecken, die in dieselbe Richtung zeigen, werden addiert. Es folgt eine Simulation zur Berechnung der maximal benötigten Distanz, um die größten Winkel je Richtung anzunehmen. Diese gibt an, welche Distanz auf dem restlichen Teilstück des Berechnungspfades in der Simulation beachtet werden muss. Die Distanz bezieht jedoch Winkel mit ein, die erst am Ende eines Berechnungspfades auftreten können, während die aktuelle Position noch am Anfang liegt. Daher werden die Winkel der Ecken auf dem restlichen Teilstück des Berechnungspfades bis zur berechneten Distanz ausgewertet und um den Winkel zwischen der Ausrichtung der Berechnungsmaschine und dem aktuellen Berechnungssegment ergänzt. Entsprechend der ersten Winkelbestimmung werden die Winkel aufaddiert und deren Maximalwerte je Richtung bestimmt. Anschließend erfolgt eine Simulation zur Bestimmung der benötigten Distanz für die maximalen Winkel. Durch die Simulation des Referenzfahrzeuges auf dem Berechnungspfad über die benötigte Distanz, wird die benötigte Zeit für die auftretenden Winkel bestimmt, welche ein weiteres Minimum für die Simulationszeit bildet.

5.6.4 Berechnung von Trajektorien

Das unter Abschnitt 5.1.2 beschriebene mathematische Modell beachtet nicht die unter Abschnitt 5.6.2 eingeführten Stellungsparameter zur Beschreibung der Trägheit der abstrahierten Gelenke. Außerdem betrachtet das mathematische Modell keine zeitlichen Aspekte, die aber für die Simulation notwendig sind. Daher wird das mathematische Modell entsprechend ergänzt und als Vorhersagemodell verwendet.

Die Lenkung wird über Positionsstellungen gesteuert. Da von einer konstanten Beschleunigung \ddot{c} ausgegangen wird (siehe Abschnitt 5.3.2, Paragraf Kolbenpositionsregelung) und die aktuelle Position c_0 und die Geschwindigkeit \dot{c}_0 bekannt sind, kann das Verhalten über einen Zeitraum δ_t mit der Funktion 5.27 angenähert werden.

$$c(c_0, \dot{c}_0, \ddot{c}, \delta_t) = c_0 + \int_0^{\delta_t} \dot{c}_0 + \ddot{c} * t dt = c_0 + \dot{c}_0 * t + \frac{\ddot{c}}{2} * t^2 \quad (5.27)$$

Das Verhalten des Antriebes kann ebenfalls durch eine Funktion über einen Zeitraum δ_t angenähert werden. Die Funktion 5.28 bildet diese Beschreibung in Abhängigkeit der

aktuellen Geschwindigkeit v_0 und der angenommenen Beschleunigung \dot{v} ab.

$$v(v_0, \dot{v}, \delta_t) = v_0 + \int_0^{\delta_t} \dot{v} dt = v_0 + \dot{v} * t \quad (5.28)$$

Mit der Ersetzung von c durch $c(c_0, \dot{c}_0, \ddot{c}, \delta_t)$ und v durch $v(v_0, \dot{v}, \delta_t)$ im mathematischen Modell, sind Funktionen gegeben, die das Verhalten der Berechnungsmaschine über einen Zeitraum unter der Berücksichtigung der Trägheit annähern.

Durch das mathematischen Modell sind Geschwindigkeiten beschrieben, welche in der Simulation über einen Zeitraum δ_t betrachtet werden, um entsprechende Trajektorien zu bestimmen. Hierzu wurden die Funktionen 5.29, 5.30 und 5.31 auf Basis der ergänzten Geschwindigkeitsbeschreibungen aufgestellt.

$$\theta(\theta_0, v_0, \dot{v}, c_0, \dot{c}_0, \ddot{c}, \delta_t) = \theta_0 + \int_0^{\delta_t} \frac{v(v_0, \dot{v}, t)}{\frac{L}{\tan(\arctan(\frac{a^2+b^2-c(c_0, \dot{c}_0, \ddot{c}, t)^2)}{2*a*b}) + \omega_a + \omega_b - \pi)} + \frac{S}{2}} dt \quad (5.29)$$

$$x(x_0, \theta_0, v_0, \dot{v}, c_0, \dot{c}_0, \ddot{c}, \delta_t) = x_0 + \int_0^{\delta_t} v(v_0, \dot{v}, t) * \cos(\theta(\theta_0, v_0, \dot{v}, c_0, \dot{c}_0, \ddot{c}, t)) dt \quad (5.30)$$

$$y(y_0, \theta_0, v_0, \dot{v}, c_0, \dot{c}_0, \ddot{c}, \delta_t) = y_0 + \int_0^{\delta_t} v(v_0, \dot{v}, t) * \sin(\theta(\theta_0, v_0, \dot{v}, c_0, \dot{c}_0, \ddot{c}, t)) dt \quad (5.31)$$

Je Phase aus Abschnitt 5.6.2 lassen sich die Funktionen 5.29, 5.30 und 5.31 rekursiv über die Phase und den vor der jeweiligen Phase geltenden Zustand parametrieren. Die so aufgebaute Funktion der letzten Phase beschreibt somit den Fahrweg der Berechnungsmaschine über die Zeit unter Berücksichtigung der Trägheit.

Es konnte jedoch für kein Integral der Funktionen 5.29, 5.30 und 5.31 eine aus elementaren Funktionen zusammengesetzte Funktion gefunden werden. Daher werden zur Bestimmung des Fahrwegs der Berechnungsmaschine die Stellungparameter aus den Phasen verwendet, um die Integrale je Phase diskret anzunähern. Die dabei entstehenden Zwischenzustände $(v_0, c_0, \dot{c}_0, \theta_0, x_0, y_0)$ werden mit dem jeweiligen Zeitpunkt kombiniert und in Form einer Liste zu einer Trajektorie zusammengefasst.

5.6.5 Bewertung von Trajektorien

Die Bewertung der Trajektorien erfolgt unter der Bedingung, dass diese bezüglich ihrer zeitlichen Dimension gleich sind. Dies erlaubt den Vergleich der Trajektorien über die Zeit. Die Positionen aus den Trajektorien werden in x- und y-Koordinaten aufgeteilt und mit dem jeweiligen Zeitpunkt des Zwischenzustands kombiniert. Dies führt zu zwei injektiven Funktionen $x(t)$ und $y(t)$, die jeder x- bzw. y-Koordinate einen eindeutigen Zeitpunkt zuordnen. Ein durch $x(t)$ bzw. $y(t)$ beschriebener Polygonzug gibt den Verlauf der x- bzw. y-Koordinate über die Zeit an. Der Flächeninhalt zwischen zwei dieser Polygonzüge entspricht der diskretisierten Abweichung bezüglich der x- bzw. y-Koordinate zwischen zwei

Trajektorien über die Zeit und wird daher als Bewertungskriterium verwendet. Am besten wird die Trajektorie mit der kleinsten Abweichung bezüglich der x- und y-Koordinate über die Zeit zur Referenztrajektorie bewertet.

5.7 Berechnung

Die automatisierte Berechnungsmaschine muss eine Berechnungsfläche berechnen. Diese Funktion wird durch die orchestrierende *Sprinkling*-Komponente realisiert. Sie lädt die Berechnungssequenzen, extrahiert die Berechnungspfade und weist die *Path Following*-Komponente an, einem Pfad zu folgen.

Initial ist *Sprinkling* im *idle*-Zustand. Damit die Berechnungsmaschine mit der Berechnung beginnt, muss das *sprinkle*-Kommando aufgerufen werden. Diesem wird eine Berechnungssequenz als Parameter übergeben, welche die Berechnungsmaschine anschließend lädt. Die *Sprinkling*-Komponente geht dafür in den *loading map*-Zustand. Wenn die Berechnungssequenz geladen ist, wird der erste Berechnungspfad der Berechnungssequenz der *Path Following*-Komponente übergeben und die Berechnungsmaschine setzt sich in Bewegung, sofern alle Systeme bereit sind. Die *Sprinkling*-Komponente befindet sich während der Berechnung im *sprinkling*-Zustand. Mit dem Eintritt in den *Sprinkling*-Zustand gibt die Berechnungsmaschine ein warnendes, akustisches Signal von sich.

Soll die Berechnung pausiert werden, muss das *pause*-Kommando ausgeführt werden, welches die *Sprinkling*-Komponente in den *paused*-Zustand versetzt. Die Berechnungsmaschine schließt daraufhin die Wasserversorgung und stoppt. Die *Sprinkling*-Komponente merkt sich den Index des zu befahrenden Berechnungspfades aus der Berechnungssequenz und stoppt die *Path Following*-Komponente. Mit dem *resume*-Kommando kann die Berechnung fortgesetzt werden, welches die *Sprinkling*-Komponente zurück in den *sprinkling*-Zustand versetzt. Dafür wird der Berechnungspfad mit dem Index des zuvor pausierten Pfades der *Path Following*-Komponente übergeben, woraufhin die Berechnungsmaschine die Wasserzufuhr wieder öffnet und die Berechnung fortsetzt.

Mit dem *abort*-Kommando kann die Berechnung abgebrochen werden. Dann schließt die Berechnungsmaschine die Wasserversorgung und stoppt. Die *Path Following*-Komponente wird beendet und die *Sprinkling*-Komponente nimmt den *idle*-Zustand an. Ein Fortsetzen der Berechnung ist nach dem *abort*-Kommando nicht mehr möglich, da der Index des zuletzt bearbeiteten Berechnungspfades nicht gespeichert wird.

Die *Sprinkling*-Komponente stellt eine Schnittstelle bereit, über die ihr Zustand sowie der Fortschritt einer Berechnung angegeben wird. Der Fortschritt umfasst den Index des aktuellen Berechnungspfades, die Anzahl der Berechnungspfade, die auf dem aktuellen Berechnungspfad zurückgelegte Strecke und die Länge des aktuellen Berechnungspfades.

Es ist möglich, dass die Berechnungsmaschine stoppt, obwohl nach Abschnitt 5.1.4 kein kri-

tischer Systemzustand vorliegt und die *Sprinkling*-Komponente sich im Zustand *sprinkling* befindet. Die *Path Following*-Komponente befindet sich beim Abfahren eines Berechnungspfades im Zustand *follow path*. In den übrigen, folgenden Zuständen ist die Berechnung gestoppt:

- *waiting for transformation*: Die *Localization*-Komponente wartet auf eine ausreichende Lokalisierungsgenauigkeit, um das *UTM*-Koordinatensystem aufzuspannen.
- *transforming path*: Die Koordinaten der Berechnungspfade werden in das *map*-Koordinatensystem transformiert.
- *waiting for localization-accuracy*: Die *Localization*-Komponente meldet keine ausreichende Genauigkeit, um dem Pfad sicher zu folgen.
- *path is out of range*: Der Pfad ist zu weit von der Maschine entfernt.
- *tilt prevention*: Die Maschine ist zu stark geneigt.

6 | Test & Optimierung

Die automatisierte Beregnungsmaschine wurde mehrfach auf drei sehr unterschiedlichen Flächen getestet:

- gepflasterte Straße im urbanen Gebiet,
- leicht geneigte, betonierte Fläche auf einem ländlich-gelegenen Betrieb,
- offen-gelegene Wiese.

Testbedingungen Der Rotor der Beregnungsmaschine wird entfernt und durch einen Akkuschauber ersetzt. Dessen Akku wird mit zwei Handgriffen alle zehn Minuten während der Beregnung ersetzt, damit die Beregnungsmaschine eine möglichst gleichbleibende Geschwindigkeit erreicht. Die durchgeführten Tests erstrecken sich über die Wintermonate und werden ohne eine Wasserversorgung ausgeführt. Die Beregnungsfläche wird mit dem Kartiergerät kartiert. Um die Beregnung zu starten, wird die Spannungsversorgung hergestellt, die Systeme hochgefahren, die Shape-Datei aus dem Kartiergerät übertragen und ein *sprinkle*-Kommando abgesetzt. Die Beregnungsmaschine ist mit einem Laptop kabellos verbunden, der über den Verlauf der Beregnung alle gemessenen und berechneten Werte aufzeichnet.

Urbanes Gebiet Die Beregnungsmaschine steht mittig zwischen zwei Gebäuden mit ca. 4 m-Abstand zu beiden Seiten. Mit dem Start der Beregnung verharrt sie zunächst auf ihrer Startposition. Es dauert ca. 25 Minuten, bis die Sensoren zur globalen Lokalisierung Messungen von ausreichender Qualität liefern. Bei sehr schlechtem Wetter (wolkenverhangener Himmel bei starkem Regen und Wind) ist es vorgekommen, dass die Sensoren auch nach 90 Minuten keine Werte mit ausreichender Genauigkeit lieferten und der Versuch abgebrochen wurde. Startet die Beregnungsmaschine die Abfahrt der Pfade, dann beendet sie die Beregnung erfolgreich. Führt der Pfad sehr nahe (ca. 2 m-Abstand) an einem Gebäude vorbei, dann lässt die Lokalisierungsgenauigkeit nach und die Beregnungsmaschine pendelt mit ein sinus-förmigen Kurve über den Pfad.

Der Antrieb der Beregnungsmaschine versagt bei einer links-gerichteten Rückwärtsfahrt. Hierbei liegt das angetriebene Rad im Kurveninneren. Bei der Vorwärtsfahrt muss das Antriebsrad für die Rückwärtsfahrt gelöst werden, da die Mechanik die Vorwärtsfahrt stark bremst. In Kurven zieht der Schlauch sich bei der Vorwärtsfahrt (Schlauch wird aufgewickelt) in das Kurveninnere, wodurch dieser seitlich über die Fläche gezogen wird.

Ländlicher Betrieb Die Berechnungsmaschine steht auf einer leicht geneigten Betonfläche. Rechts von der Maschine befindet sich mit einem Abstand von ca. 10 m ein Futterlager. Die Sensoren zur Lokalisierung liefern nach ca. 15 Minuten Werte von ausreichender Qualität, sodass sich die Berechnungsmaschine in Bewegung setzt.

Muss die Berechnungsmaschine rückwärts eine Steigung bewältigen, dann kann der Antrieb nicht genügend Kraft aufbringen und die Maschine steht still. Wenn die Maschine vorwärts (Schlauch wird aufgewickelt) auf abschüssigen Gelände fährt, dann ist Fahrgeschwindigkeit sehr ungleichmäßig. Der Schlauch wird gespannt und die Maschine beschleunigt. Die Schlauchspannung lässt nach und die Maschine rollt ein Stück mit entspanntem Schlauch. Sie erreicht kurzzeitig eine unerwartet hohe Geschwindigkeit und entfernt sich vom Pfad. Daraufhin lenkt sie ein, der Schlauch wird wieder gespannt und die Maschine beschleunigt erneut. Nun kommt sie auf der anderen Seite des Pfades zum stehen. Dadurch pendelt sie über den Pfad und fährt nicht die beabsichtigte, gerade Linie. Dieses Verhalten setzt sich fort, solange das abschüssige Gelände andauert.

Bei der Vorwärtsfahrt wird der Schlauch in das Kurveninnere gezogen. Das Antriebsrad für die Rückwärtsfahrt muss auch hier gelöst werden.

Wiese Die benötigte Lokalisierungsgenauigkeit ist bereits nach ca. 15 Minuten erreicht. Jedoch genügt die Kraft des Antriebs nicht, um die Berechnungsmaschine zu bewegen. Das gilt für die Vorwärts- und Rückwärtsfahrt. Der Test wird daraufhin abgebrochen.

Probleme des Testverfahrens Bei keiner der Tests ist die Berechnung von einer Person eingerichtet und gestartet worden, die nicht oder nur wenig mit dem System vertraut ist. Außerdem hat die Berechnungsmaschine keinen einzigen Test absolviert, der länger als zwei Stunden dauerte. Die automatisierte Berechnungsmaschine ist noch nie durch Wasserdruck angetrieben worden.

Positive Effekte des Testverfahrens Die Tests erstrecken sich über die gesamte Entwicklungsphase. So wurde inkrementell getestet. Grundlegende Komponenten, die sehr früh entwickelt werden mussten, da weitere Komponenten auf ihnen aufbauen, wurden so sehr häufig getestet. Auftretende Fehler wurden bearbeitet und die Lösung beim nächsten Test überprüft. So ist die Basis der Berechnungsmaschine vielfach getestet und optimiert worden.

Abschließende Testresultate Die Lokalisierungs-komponenten sind nicht direkt einsatzbereit und benötigen immer eine Aufwärmphase. Diese kann aber in einer Berechnungsaufgabe berücksichtigt werden und schränkt die nach einem Berechnungsplan operierende, automatisierte Berechnungsmaschine nicht ein. Eine Berechnung auf Anfrage kann aber nur mit einer Verzögerung stattfinden.

Der Antrieb der Berechnungsmaschine muss überarbeitet werden. Im derzeitigen Zustand benötigt die Berechnungsmaschine mehrfach manuelle Eingriffe, um eine realistische Berechnungssequenz zu berechnen. Für viele Bereiche, in denen Berechnungsmaschinen eingesetzt

werden, ist der Antrieb ungeeignet.

Wird die Berechnungsmaschine von hohem Objekt überragt, dann leidet die Lokalisierungs-
genauigkeit. In Obstplantagen könnte dies ein Problem darstellen.

Wenn im Test keins der genannte Probleme auftritt, dann folgt die Berechnungsmaschine
dem Pfad sehr präzise. Von einem Pfad, dem die Maschine physikalisch folgen kann, weicht
sie selbst in Kurven nicht mehr als 10 cm ab. Von einem Pfad, der eine 90°-Ecke enthält,
weicht die Maschine maximal 50 cm ab.

7 | Zusammenfassung

Im Rahmen dieser Masterarbeit wurde eine Berechnungsmaschine vom Modell Minimobil 40 des Herstellers Fasterholt automatisiert. Dafür wurde es mit einem informationstechnischen System ausgestattet, welches die Kontrolle über die Lenkung, ein Getriebe und die Wasserzufuhr übernimmt.

Die Automatisierung schränkt die Einsatzmöglichkeiten der Berechnungsmaschine nicht ein und bedarf keiner Instrumentalisierung der zu berechnenden Fläche. Sie fährt selbstständig auf zuvor definierten Fahrspuren und verlässt diese nicht. Kommt es zu einem Fehler, wird dieser protokolliert und signalisiert. Die automatisierte Berechnungsmaschine stoppt und schließt die Wasserzufuhr. Ein solcher Fehler wäre ein zu hoher Neigungswinkel, ein zu schlechtes Positionssignal oder das Betätigen des Notaus-Schalters.

Vom Bediener werden die Fahrspuren mit einem mobilen Kartierungsgerät aufgezeichnet. Dieses bietet eine Bedienoberfläche, die der Bediener mit seinem Smartphone benutzen kann. Mit der Bedienoberfläche bringt der Bediener die Fahrspuren in die abzufahrende Ordnung und erzeugt daraus eine Shape-Datei. Diese kann die automatisierte Berechnungsmaschine einlesen.

Die automatisierte Berechnungsmaschine bestimmt ihre Position und Ausrichtung, setzt diese in Relation zur Fahrspur und errechnet die erforderlichen Manöver, um dieser zu folgen. Sie folgt den definierten Spuren, ändert selbstständig die Fahrrichtung und beendet den Berechnungsvorgang, nachdem sie die Fahrspuren in der definierten Ordnung abgefahren hat.

Die automatisierte Berechnungsmaschine ist modular aufgebaut. Die verwendeten Komponenten können ausgetauscht oder ergänzt werden. Sie bietet die Schnittstellen, um weitere Aktoren oder Sensoren einzubinden. Weiter ist die Integration einer Bedienoberfläche für die automatisierte Berechnungsmaschine vorgesehen und vorbereitet.

Die automatisierte Berechnungsmaschine berechnet eine komplette Berechnungsfläche, ohne dass nach der Einrichtung ein Zutun des Bedieners erforderlich ist.

8 | Ausblick

Die im Kapitel 6 ermittelten Probleme können durch Austausch von Hardware gelöst werden.

Lokalisierung Für eine präzise Lokalisierung ist immer eine Aufwärmphase notwendig. Ihre Dauer kann aber durch den Einsatz von neuer Technologie verringert werden. Während des Projekts kam der ZED-F9P-Chip nach [55] vom Hersteller Ublox auf den Markt. Gegenüber dem M8P-Chip verwendet der ZED-F9P zwei GNSS-Antennen, wodurch die Berechnungsmaschine nur noch ein einzigen Chip für die absolute Lokalisierung benötigen würde. Die zweite Antenne ermöglicht dem ZED-F9P multipath-Fehler auszugleichen. Diese werden durch Objekte verursacht, welche die GNSS-Antenne überragen. Mit dem ZED-F9P könnte die Berechnungsmaschine auch in Gebieten eingesetzt werden, in denen ein größerer Teil des Himmels verdeckt ist. Damit wird eine kleine Berechnungsmaschine denkbar, die im urbanen Gebiet in privaten Gärten eingesetzt werden kann. Das C099-F9P-Modul nach [56] integriert den ZED-F9P-Chip und kann verwendet werden, um diesen für die gewünschte Applikation zu erproben.

Antrieb Der Antrieb bringt nicht die notwendige Kraft auf, um die Berechnungsmaschine unter realistischen Bedingungen einzusetzen. Die seed2soil & Co. KG entwickelt bereits einen rein elektrischen Antrieb, welcher die Hinterräder der Berechnungsmaschine antreiben soll. Dieser soll die Berechnungsmaschine sowohl vorwärts als auch rückwärts antreiben. Ein Ziehen am Schlauch ist dann nicht mehr erforderlich, wodurch dieser nicht mehr in das Kurveninnere gezogen würde. Außerdem hätte die Flexibilität des Schlauchs keinen Einfluss mehr auf die Geschwindigkeit der Berechnungsmaschine. Die Geschwindigkeit ließe sich sogar für jeden Berechnungspfad festlegen, wodurch sich die aufzutragende Menge an Wasser steuern ließe. Die Basis für eine solche Geschwindigkeitsregelung wurde sowohl in der Hardware als auch in der Software der automatisierten Berechnungsmaschine bereits geschaffen.

Solarzelle Das volle Potenzial der Solarzelle wird nicht genutzt. Bisher genügt die Leistung, um die Batterie auf hohem Ladestand zu halten. Wird die Berechnungsmaschine jedoch um einen elektrischen Antrieb ergänzt, reicht die Leistung wahrscheinlich nicht aus. Die Leistung erhöht sich stark, wenn man die Solarzelle auf die Sonne ausrichtet. Da die Berechnungsmaschine über eine Internetverbindung verfügt, kann sie den Standort der Sonne ermitteln. Mit einem Linearantrieb kann die Neigung der Solarzelle beeinflusst und so die Ladeleistung erhöht werden.

Bedienoberfläche Damit die Berechnungsmaschine von einem Bediener und Kunden komfortabel bedient werden kann, ist eine Bedienoberfläche erforderlich. In einer solchen könnte dieser einen Berechnungsplan definieren und bei Bedarf ändern. Außerdem könnte der Bediener Berechnungssequenzen direkt in der Bedienoberfläche definieren oder ändern. Dies ist allerdings nur möglich, wenn eine große Abweichung vom Berechnungspfad tolerierbar ist, da das Kartenmaterial im Internet nicht so präzise wie das Kartierungsgerät ist. Für den Bediener wichtige Einstellungen, wie die aufzutragende Wassermenge, könnten ebenfalls über eine Bedienoberfläche gesetzt werden. Die Schnittstellen zur Anbindung einer Bedienoberfläche sind bereits vorhanden.

Größenadaptierung Die seed2soil & Co. KG denkt über größere und kleinere automatisierte Berechnungsmaschinen nach. Die Hard- und Software der automatisierten Berechnungsmaschine sind vollkommen parametrisierbar, sodass das System direkt auf kommende Modelle übertragen werden kann.

A | Interview

Datum: 05.06.2019, 16:00
Interviewer: Florian Schmariede
Protokollant: Marvin Banse
Interviewte: Michael Pokriefke
Malte Hamann
Justus Lange

Tätigkeitsfeld der seed2soil GmbH & Co. KG:

- Engineering im Bereich Landwirtschaft
- Schnittstellen zur Anwendertechnik

Zuständigkeiten der Beschäftigten:

- Malte: IT-Infrastruktur, Außenwirkung, Kontruktion
- Michael: Projektmanagement, Patente, Finanzen und Verträge, Personal, Aufträge
- Justus: Konstruktion, Versuche, Präsentationen, Außenwirkung
- Florian: Konstruktion, Versuche, Metallbau, Prototyping

Voraussichtliche Einsatzumgebung der Beregnungsmaschine:

- Gemüseanbau (Sonderkulturen)
- Baumschulen
- Sportanlagen (Grünanlagenpflege)
- Gartenbaubetrieb
- Beregnungsanwendung in Bereichen mit kleinem Budget

Eigenschaften der automatisieren Beregnungsmaschine

- höchste, gewünschte Genauigkeit: +/-5cm
- höchste, tolerierbare Genauigkeit: +/-15cm
- maximale, unterstützte Geschwindigkeit: 2km/h
 - Eine hohe Geschwindigkeit begrenzt die aufgetragene Wassermenge
- erwartete Geschwindigkeit: 1km/h bei $60m^3/h$ Wassermenge
- Später unterschiedliche Geschwindigkeit auf Beregnungspfaden denkbar
- Lenkungsspiel: 2mm
- Spurbreite: 30-40 cm

Verhalten im Fehlerfall:

- Beregnungsmaschine soll unterbrochen werden können -> stehenbleiben

- Fahrt später fortsetzen
- oder Return-to-Home

• Steckenbleiben

- detektieren, abschalten und melden
- Schlauchroller als Informationsquelle

• Schlauch wird aus der Fahrspur gezogen (Kurvenfahrten)

- ggfs. Lanze als Kontroll-/Korrekturinstrument
- Problematik soll durch Definition eines maximalen Kurvenradius umgangen werden
- Problematik kann durch Führungen auf der Beregnungsfläche behandelt werden
- Problematik senkt Präzision des Schlauchrollers

Bedienoberfläche:

- Übersicht in einer OpenStreetMap
- Daten Im Kontext einer Maschine:
 - Beregnungssequenz (mit Kurvenradien) als Overlay
 - Beregnungspfade als Rote oder Grüne Fahrgasse mit Genauigkeitsvoraussetzung
 - Wo liegt gerade Schlauch?
 - Wo ist es nass?
 - Time-to-Finish (grob geschätzt)
- Für eine Maschine sollen verzögerte Startzeiten hinterlegt werden können
- Beregnungssequenzen können mittels Shape-Dateien erstellt werden
 - Shape-Dateien sind sehr verbreitet und ein Standard in dem Bereich.

Hardwarekomponenten, welche angesteuert werden müssen:

- Kugelhahn zum Schließen der Wasserversorgung
 - Magnetventil schließt Wasserleitung zu schlagartig -> Druckanstieg im Schlauch im Schließmoment
- 4x Linak Linearmotoren 12V
- RS online Drehgeber / Drehwinkelsensor

B | Anforderungsliste

- PCR1** Eine automatisierte Berechnungsmaschine muss Berechnungsflächen, welche im Außenbereich liegen, berechnen können. 11
- PCR1.1** Eine automatisierte Berechnungsmaschine muss unter den selben Wetterbedingungen wie eine nicht automatisierte Berechnungsmaschine arbeiten können. 11
- UR2** Auf einer Berechnungsfläche müssen für die Berechnung durch eine Berechnungsmaschine erforderliche Fahrspuren definiert sein. 11, 17, 23
- UR2.1** Fahrspuren sollen als Polygonzüge in einer Shape-Datei mit Koordinaten aus dem WGS84 beschrieben werden. 18, 20, 89, 90
- CR2.2** Fahrspuren, die durch Polygonzüge in einer Shape-Datei mit Koordinaten aus dem WGS84 beschrieben sind, sollen vom System verarbeitet werden können. 18
- UR2.3** Der Bediener muss Fahrspuren so kartieren, dass bei einem Export als Shape-Datei diese einer Berechnungssequenz nach Definition 3 entsprechen. 21, 22, 70
- CR2.4** Das System muss eine Schnittstelle anbieten, über die Shape-Dateien, welche Berechnungssequenzen nach Definition 3 entsprechen, angenommen werden. 21
- CR3** Eine automatisierte Berechnungsmaschine darf sich nur auf zuvor definierten Fahrspuren selbstständig bewegen. 11, 17, 23
- UR4** Für den Betrieb einer automatisierten Berechnungsmaschine muss ein Wasseranschluss im Aktionsradius der Berechnungsmaschine vorhanden sein. 11
- PCR5** Der Instrumentalisierungsaufwand einer Berechnungsfläche darf bei einer automatisierten Berechnungsmaschine nicht höher sein als bei einer nicht automatisierten Berechnungsmaschine. 11, 64
- PCR6** Eine automatisierte Berechnungsmaschine muss mit Notausschaltern ausgestattet sein. 12
- PCR7** Eine automatisierte Berechnungsmaschine muss den Beginn einer Berechnung signalisieren. 12

- UR8** Fahrspuren dürfen während einer Beregnung nicht durch Lebewesen oder Objekte blockiert sein. 12
- PR9** Es muss eine Beregnungsmaschine vom Hersteller Fasterholt vom Modell Minimobil 40 gestellt werden. 12
- PCR10** Als Grundlage der automatisierten Beregnungsmaschine soll eine Beregnungsmaschine vom Hersteller Fasterholt mit der Produktbezeichnung Minimobil 40 verwendet werden. 12
- CR11** Das Minimobil 40 muss durch ein informationstechnisches System ergänzt werden. 13
- PR12** Das Minimobil 40 muss durch eine Komponente, welche die Vorwärts- sowie die Rückwärtsfahrt erlaubt, ergänzt werden. 14, 28, 46
- PR12.1** Die Komponente zur Fahrtrichtungsbestimmung muss eine Schnittstelle, geeignet für informationstechnische Systeme, zur Steuerung der Fahrtrichtung bieten. 14, 24, 28, 47, 90
- PR13** Der Wasserhahn des Minimobil 40 soll durch eine für informationstechnische Systeme geeignete Schnittstelle ergänzt werden. 14, 24, 28, 47, 90
- PR14** Die Lenkung des Minimobil 40 soll durch eine für informationstechnische Systeme geeignete Schnittstelle ergänzt werden. 14, 24, 28, 47, 90
- UR15** Die Deichsel des Minimobil 40 darf während der Beregnung nicht auf dem Schlauch aufliegen. 14
- PR16** Das Minimobil 40 muss um eine entsprechend dimensionierte Stromquelle ergänzt werden. 15, 28, 41
- PCR17** Alle zusätzlich angebrachten Komponenten und Umbauten müssen gegenüber Wasser resistent sein. 15, 29
- PR18** Das Minimobil 40 muss um einen trockenen Bereich ergänzt werden. 15, 28
- CR20** Es muss ein Kartierungssystem zur Kartierung von Fahrspuren entwickelt werden. 20, 29, 69
- CR20.1** Das Kartierungssystem muss so klein und leicht sein, dass es von Hand getragen werden kann. 20, 29
- CR20.2** Das Kartierungssystem muss so flexibel sein, dass es von Hand getragen werden kann sowie an einem Fahrzeug befestigt werden kann. 20, 29
- CR20.3** Das Kartierungssystem muss den Export dieser gemäß Anforderung UR2.1 ermöglichen. 20

- CR20.4** Das Kartierungssystem muss über eine Bedienoberfläche verfügen. 20
- CR20.4.1** Die Bedienoberfläche des Kartierungssystems muss es dem Bediener ermöglichen, den Anfang sowie das Ende einer Fahrspur zu markieren. 20, 69
- CR20.4.2** Die Bedienoberfläche des Kartierungssystems muss es dem Bediener ermöglichen, die nach Anforderung UR2.1 exportierten Daten abzurufen. 20, 69
- PR20.5** Die benötigten Komponenten für die Entwicklung und den Bau des Kartierungssystems müssen gestellt werden. 20
- CR21** Das System muss eine Schnittstelle anbieten, über die Beregnungssequenzen hinterlegt werden können. 21
- CR22** Das System muss eine Schnittstelle anbieten, über die eine Beregnung mit einer Beregnungssequenz gestartet werden kann. 22, 25
- CR23** Das System soll eine Schnittstelle anbieten, über die eine Beregnung einer Beregnungssequenz abgebrochen werden kann. 22
- CR24** Das System soll eine Schnittstelle anbieten, über die eine Beregnung einer Beregnungssequenz pausiert werden kann. 22
- CR25** Das System soll eine Schnittstelle anbieten, über die eine pausierte Beregnung einer Beregnungssequenz fortgesetzt werden kann. 22
- CR26** Die automatisierte Beregnungsmaschine muss eine Beregnungssequenz abfahren können. 22, 25, 70
- CR27** Die automatisierte Beregnungsmaschine muss ihre Pose global bestimmen können. 23, 29, 59
- CR27.1** Die automatisierte Beregnungsmaschine muss sich auf mindestens 15 cm genau lokalisieren können. 23, 63
- CR28** Das System muss eine Möglichkeit bieten, eine einzuhaltende Genauigkeit für einen Beregnungspfad zu definieren. 24
- UR28.1** Der Bediener muss eine einzuhaltende Genauigkeit für einen Beregnungspfad definieren. 24, 70
- CR29** Die automatisierte Beregnungsmaschine muss, sobald die vorgegebene Genauigkeit nicht mehr eingehalten werden kann, stoppen. 24, 65, 72
- CR30** Das System muss ein Umkippen der Beregnungsmaschine detektieren. 24, 66
- CR31** Die Beregnungsmaschine muss stoppen, wenn sie umkippt. 24, 68
- CR32** Das System soll die Schnittstelle, nach Anforderung PR12.1, zur Steuerung der Fahrtrichtung verwenden. 24, 29

- CR33** Das System soll die Schnittstelle, nach Anforderung PR13, zur Steuerung der Wasserzufuhr verwenden. 24, 29, 59
- CR34** Das System soll die Schnittstelle, nach Anforderung PR14, zur Steuerung der Lenkung verwenden. 24, 29
- CR35** Das System soll eine Möglichkeit bieten, eine einzuhaltende Geschwindigkeit für einen Berechnungspfad zu definieren. 25
- UR35.1** Der Bediener soll eine einzuhaltende Geschwindigkeit für einen Berechnungspfad definieren. 25, 70
- CR36** Die automatisierte Berechnungsmaschine soll, wenn möglich, vorgegebene Geschwindigkeiten einhalten. 25, 73
- CR37** Das System soll so entwickelt werden, dass die Ergänzung um eine Ansteuerung einer alternativen Antriebstechnologie möglich ist. 25
- CR38** Das System muss Störungen erkennen und identifizieren. 25
- CR38.1** Das System muss Informationen zu Störungen über eine Schnittstelle bereitstellen. 25
- CR39** Das System muss die aktuell ermittelte Pose über eine Schnittstelle bereitstellen. 26
- CR40** Das System muss die Koordinaten des aktuellen Berechnungspfades über eine Schnittstelle bereitstellen. 26
- CR41** Das System muss den Fortschritt innerhalb der aktuellen Berechnungssequenz über eine Schnittstelle bereitstellen. 26
- CR42** Das System muss die ermittelte Abweichung über eine Schnittstelle bereitstellen. 26
- CR43** Das System muss den ermittelten Lokalisierungsfehler über eine Schnittstelle bereitstellen. 26
- CR44** Das System soll so entwickelt werden, dass die Ergänzung um weitere Sensoren möglich ist. 26

C | Konstruktionszeichnungen

C.1 Schaltplan

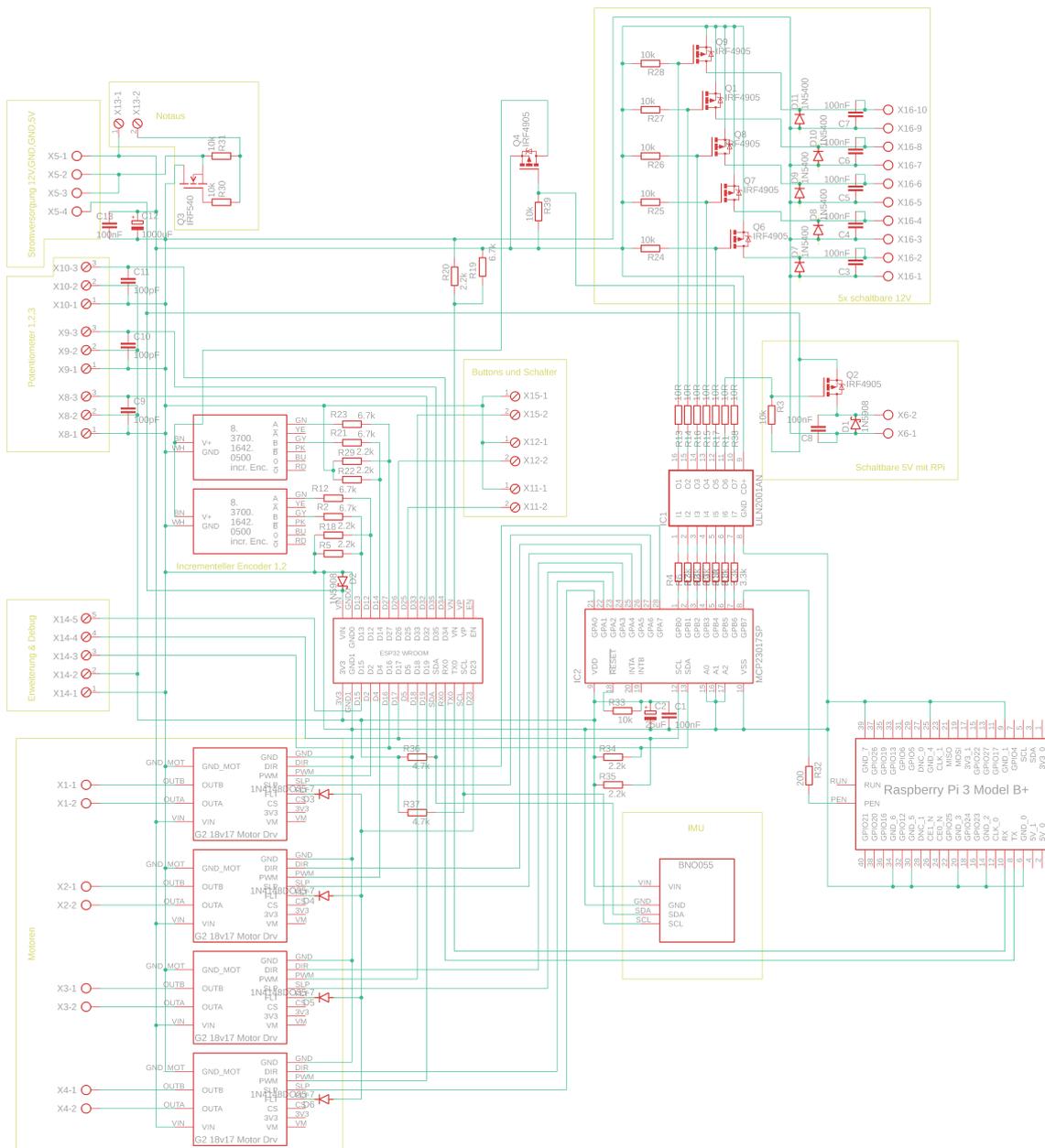


Abbildung C.1: Verschaltung aller auf der Hauptplatine der Berechnungsmaschine angebrachten Komponenten und Schnittstellen

C.2 Platinenlayout

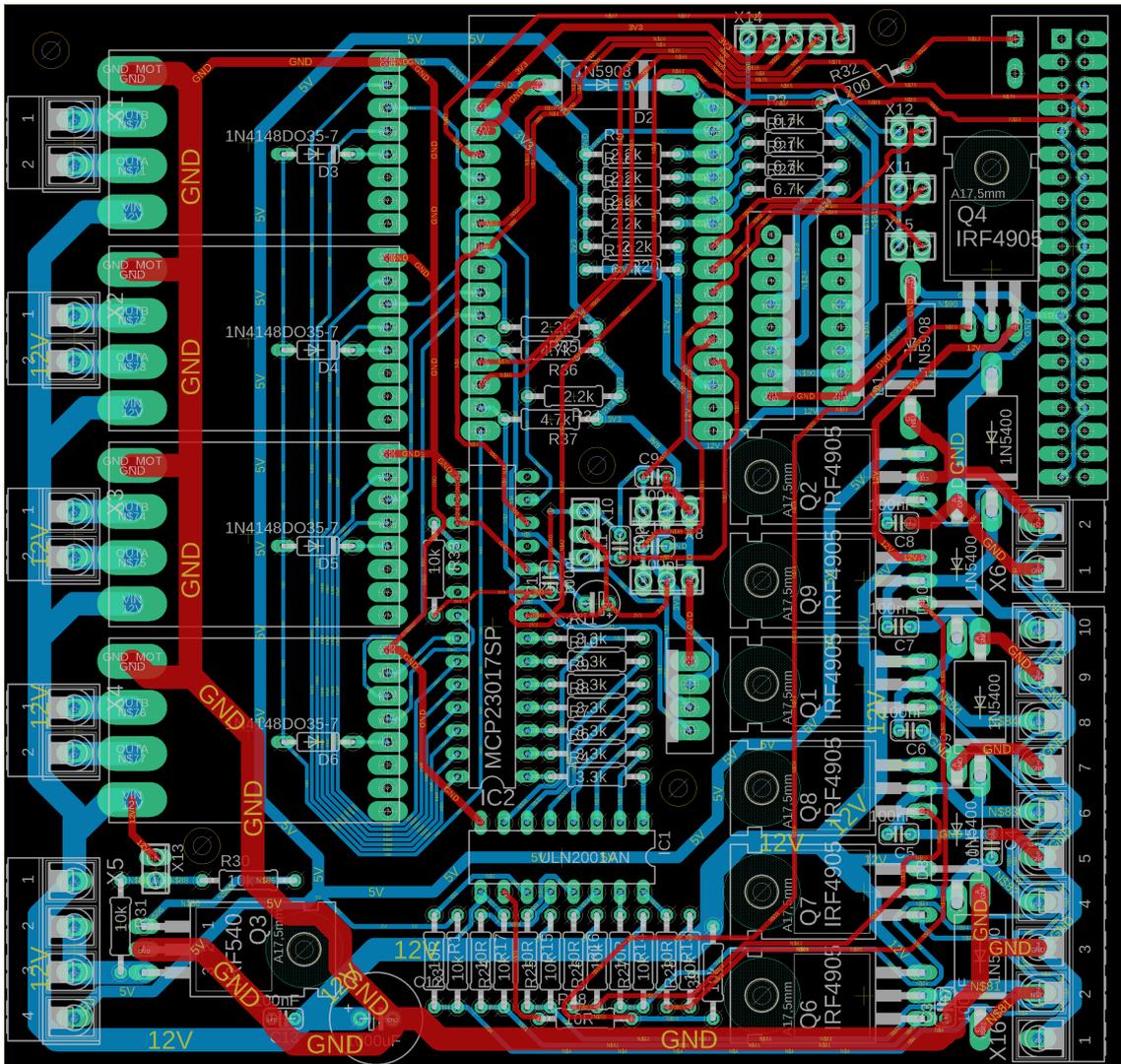


Abbildung C.2: Das Platinenlayout, welches auf dem Schaltplan der Hauptplatine aus Anhang C.1 basiert

C.3 Kartiergerät

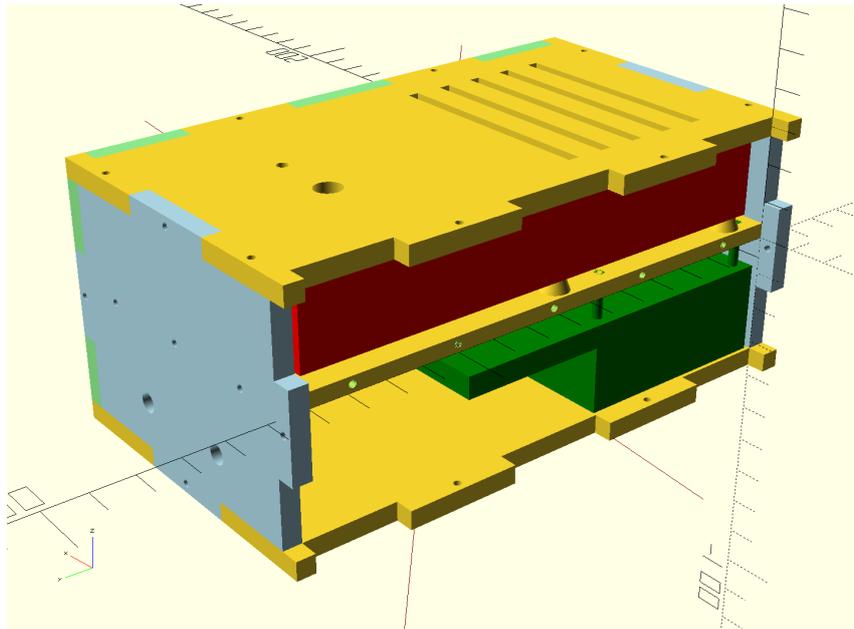


Abbildung C.3: Das Kartiergerät, von links betrachtet, mit geöffneter, linken Seitenwand

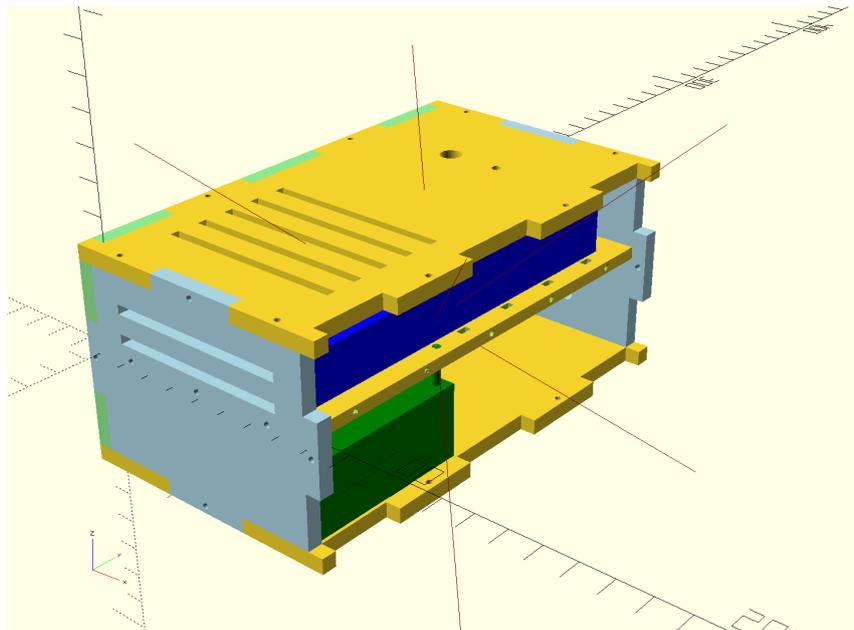


Abbildung C.4: Das Kartiergerät, von rechts betrachtet, mit geöffneter, rechten Seitenwand

Literatur

- [1] Ekkehard Fricke, Angela Riedel. *Einführung in die Feldberechnung*. Vortrag im Rahmen des vierten Treffens der Netzwerke Wasser und Berechnung. Landwirtschaftskammer Niedersachsen. 31. Mai 2017. URL: <https://www.lwk-niedersachsen.de/download.cfm/file/28180.html> (besucht am 29.03.2019).
- [2] Heinz Sourell. „Technik der Feldberechnung“. In: *Grundlagen Landtechnik Bd. 28 Nr. 4* (1978). URL: <http://440ejournals.uni-hohenheim.de/index.php/Grundlagen/article/download/512/458> (besucht am 01.04.2019).
- [3] *ESRI Shapefile Technical Description*. J-7855. Environmental Systems Research Institute. 380 New York Street Redlands, CA USA, Juli 1998. URL: <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf> (besucht am 01.04.2019).
- [4] FASTERHOLT. *minimobil-40-1.jpg*. URL: <https://www.fasterholt.dk/images/Produkte/Minimobil-40/minimobil-40-1.jpg> (besucht am 25.05.2019).
- [5] *MiniMobil 32-40 Operator Instructions and Spare Parts Catalogue*. FASTERHOLT. A/S FASTERHOLT MASKINFABRIK EJSTRUPVEJ 22, DK-7330 BRANDE DENMARK, 2012. URL: https://www.fasterholt.dk/images/PDF/Betjeningsvejledninger/Betjening_-_engelsk/minimobil32-40-2011-engelsk.pdf (besucht am 06.02.2020).
- [6] Marvin Banse, Patrick Burke, Daniel Bollmann, Yannick Habecker, Kai Hartel, Dario Imsande, Dennis Lisiecki, Stephan Mävers, Florian Schmalriede, Judith Segerath, Jannik Spieker. *LIO - Lawn In Order. Projektdokumentation*. Projektgruppe WiSe18/SoSe19. Universität Oldenburg, Department für Informatik, Abt. Systemsoftware und verteilte Systeme + Software Engineering. 30. Sep. 2018.
- [7] ublox. *C94-M8P - u-blox RTK application board package - product summary*. 2017. URL: https://www.u-blox.com/sites/default/files/C94-M8P-AppBoard_ProductSummary_%28UBX-15024894%29.pdf.
- [8] Brian Tee. *AgOpenGPS*. 23. Mai 2019. URL: <https://agopengps.jimdosite.com/>.
- [9] *Data File Header Structure for the dBASE Version 7 Table File*. dBase LLC. URL: http://www.dbase.com/KnowledgeBase/int/db7_file_fmt.htm (besucht am 13.02.2020).
- [10] Espressif. *QGIS Project*. 13. Feb. 2020. URL: <https://docs.qgis.org/3.4/pdf/de/QGIS-3.4-UserGuide-de.pdf> (besucht am 13.02.2020).
- [11] osgEarth - a C++ Geospatial SDK. *Usage of GPIO - Raspberry Pi*. URL: <http://docs.osgearth.org> (besucht am 13.02.2020).

- [12] The MapServer Team. *MapServer Documentation*. 20. Jan. 2020. URL: <http://download.osgeo.org/mapserver/docs/MapServer.pdf> (besucht am 13.02.2020).
- [13] ublox. *NEO-M8P - Data Sheet*. 16. Mai 2017. URL: https://www.u-blox.com/sites/default/files/NEO-M8P_DataSheet_%28UBX-15016656%29.pdf.
- [14] Paul Vixie, Marcela Mašláňová, Colin Dean and Tomáš Mráz. *crond - daemon to execute scheduled commands*. URL: <http://man7.org/linux/man-pages/man8/cron.8.html> (besucht am 17.02.2020).
- [15] *systemd.timer - Timer unit configuration*. URL: <http://man7.org/linux/man-pages/man5/systemd.timer.5.html> (besucht am 17.02.2020).
- [16] *Microsoft - Task Scheduler*. URL: <https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-task-scheduler> (besucht am 17.02.2020).
- [17] T. Ripel J. Hrbáček und J. Krejsa. „Ackermann mobile robot chassis with independent rear wheel drives“. In: Brno University of Technology: J. Hrbáček, T. Ripel und J. Krejsa, 2010.
- [18] Stefan K. Gehrig and Fridtjof J. Stein. *A Trajectory-Based Approach for the Lateral Control of Car Following Systems*. Research Institute Daimler Benz AG, 70546 Stuttgart, Germany.
- [19] Jarrod M. Snider. *Automatic Steering Methods for Autonomous Automobile Path Tracking*. Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania, 2009.
- [20] Andreas Jossen. *Die Bleibatterie*. Techn. Ber. BaSyTec GmbH, 6. Juli 2018. URL: <https://basytec.de/Literatur/Bleibatterie.pdf> (besucht am 19.02.2020).
- [21] Manfred Albach. *Elektrotechnik*. Pearson Studium, 2011. ISBN: 978-3-86894-081-7. URL: <https://www.amazon.com/Elektrotechnik-Manfred-Albach/dp/3868940812?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=3868940812>.
- [22] Espressif Systems. *ESP32 Series Datasheet*. 2019. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [23] BEHA AMPROBE. *37XR-A Datasheet*. 1. Sep. 2007. URL: <http://www.flukepartnerportal.com/efulfillment/11307-eng> (besucht am 29.02.2020).
- [24] Helmut Weber. *ESP32-ADC. ESP32 How to correct the ADC*. 10. Aug. 2019. URL: <https://github.com/MacLeod-D/ESP32-ADC> (besucht am 19.02.2020).
- [25] STMicroelectronics. *ULN2001 - Seven Darlington arrays - Datasheet*. URL: <https://www.st.com/resource/en/datasheet/uln2001.pdf>.
- [26] International Rectifier. *IRF4905PbF - HEXFET Power MOSFET*. 11. Juni 2003. URL: <https://www.infineon.com/dgdl/irf4905pbf.pdf?fileId=5546d462533600a4015355e329b11>.
- [27] Vishay General Semiconductor. *1N5408 - General Purpose Plastic Rectifier*. 1. Aug. 2013. URL: <http://www.vishay.com/docs/88516/1n5400.pdf>.
- [28] Transmotec AB. *DLB Series / 100 - 300mm / 500 - 2500N / potentiometer (POT)*. 2019. URL: <https://www.transmotec.de/Download/Datasheets/DLB-POT.pdf>.

- [29] Pololu. *Pololu G2 High-Power Motor Driver 18v17*. 8. Jan. 2020. URL: <https://www.pololu.com/product/2991>.
- [30] Patrick Schnabel. *PWM - Pulsweitenmodulation*. 8. Jan. 2020. URL: <https://www.elektronik-kompodium.de/sites/kom/0401111.htm>.
- [31] Wikipedia. *Vierquadrantensteller*. 10. Juni 2017. URL: <https://de.wikipedia.org/wiki/Vierquadrantensteller>.
- [32] Vishay General Semiconductor. *1N4148 - Small Signal Fast Switching Diodes*. 6. Juli 2017. URL: <https://www.vishay.com/docs/81857/1n4148.pdf>.
- [33] Open Source Robotics Foundation. *ROSserial*. Hrsg. von Romain Reignier. 1. Okt. 2018. URL: <http://wiki.ros.org/rosserial>.
- [34] Atmel. *ATmega328P*. 2015. URL: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.
- [35] Atmel. *Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V*. 2014. URL: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf.
- [36] Atmel. *ATmega16U4/ATmega32U4*. 2016. URL: http://ww1.microchip.com/downloads/en/devicedoc/atmel-7766-8-bit-avr-atmega16u4-32u4_datasheet.pdf.
- [37] Atmel. *SAM3X / SAM3A Series*. 2015. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf.
- [38] Freescale Semiconductor. *Kinetis KL26 Sub-Family. 48 MHz Cortex-M0+ Based Microcontroller*. 2014. URL: <https://www.pjrc.com/teensy/KL26P64M48SF5.pdf>.
- [39] Freescale Semiconductor. *Kinetis K66 Sub-Family. 80 MHz ARM Cortex-M4F Microcontroller*. 2016. URL: <https://www.pjrc.com/teensy/K66P144M180SF5V2.pdf>.
- [40] STMicroelectronics. *STM32F105xx / STM32F107xx*. 2017. URL: <https://www.st.com/resource/en/datasheet/stm32f105r8.pdf>.
- [41] Espressif Systems. *ESP8266EX Datasheet*. 2019. URL: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
- [42] Microchip Technology Inc. *MCP23017/MCP23S17*. 2016. URL: <http://ww1.microchip.com/downloads/en/devicedoc/20001952c.pdf>.
- [43] EETech Media LLC. *Pull up resistor / Pull down resistor*. 2019. URL: http://www.resistorguide.com/pull-up-resistor_pull-down-resistor/ (besucht am 13.02.2020).
- [44] Frank Brall. *Regelungstechnik*. 13. Sep. 2014. URL: <https://rn-wissen.de/wiki/index.php/Regelungstechnik> (besucht am 24.02.2020).
- [45] TiMOTION. *TA2 series Datasheet*. 16. Jan. 2019. URL: https://www.timotion.com/_upload/files/datasheet_ta2-p_en.pdf.

- [46] Wim Meeussen. *REP 105 - Coordinate Frames for Mobile Platforms*. 27. Okt. 2010. URL: <https://www.ros.org/reps/rep-0105.html> (besucht am 02.03.2020).
- [47] Fritz Kübler GmbH. *Incremental encoders 3700/3720*. 2019. URL: https://www.kuebler.com/pdf?3700-3720_en.pdf.
- [48] Landesamt für Geoinformation und Landentwicklung Baden-Württemberg. *Messverfahren*. URL: <http://www.sapos-bw.de/messverfahren.php> (besucht am 22.02.2020).
- [49] ublox. 8. Feb. 2020. URL: <http://wiki.ros.org/ublox> (besucht am 02.03.2020).
- [50] *Für mehr Wirtschaftlichkeit und Umweltschutz: Niedersachsen stellt SAPOS-Dienste ab sofort kostenlos zur Verfügung*. URL: <https://www.ml.niedersachsen.de/startseite/aktuelles/pressemitteilungen/fur-mehr-wirtschaftlichkeit-und-umweltschutz-niedersachsen-stellt-sapos-dienste-ab-sofort-kostenlos-zur-verfugung-181197.html> (besucht am 02.03.2020).
- [51] Sebastian Thrun. *Probabilistic robotics*. eng. Cambridge, Mass. [u.a., 2006.
- [52] *robot_localization*. 16. Feb. 2020. URL: http://wiki.ros.org/robot_localization (besucht am 02.03.2020).
- [53] Bosch Sensortec. *BNO055: data sheet*. 2016. URL: https://www.bosch-sensortec.com/media/boschsensortec/downloads/smart_sensors_1/bno055/bst-bno055-ds000.pdf.
- [54] Gregor Klancar, Drago Matko, Saso Blazic. *Mobile Robot Control on a Reference Path*.
- [55] ublox. *ZED-F9P. u-blox F9 high precision GNSS module*. 20. Feb. 2020. URL: https://www.u-blox.com/sites/default/files/ZED-F9P_DataSheet_%28UBX-17051259%29.pdf.
- [56] ublox. *C099-F9P Application Board. Easy evaluation of u-blox ZED-F9P with multi-band RTK*. 25. Okt. 2018. URL: https://www.u-blox.com/sites/default/files/C099-F9P-AppBoard_ProductSummary_%28UBX-18022364%29.pdf (besucht am 02.03.2020).
- [57] Bibliographisches Institut GmbH. *Haspel nach Duden*. 1. Mai 2019. URL: <https://www.duden.de/rechtschreibung/Haspel>.
- [58] Andreas Schwarz. *I2C*. 2019. URL: <https://www.mikrocontroller.net/articles/I%C2%B2C> (besucht am 13.02.2020).

Glossar

ADC

Ein elektrische Baustein, welcher gemessenen, analogen Spannungsbereich in einen diskreten Wertebereich (digital) übersetzt. 42–44

Berechnungsaufgabe

Die geplante Abfahrt einer Berechnungssequenz zu einem definierten Startzeitpunkt. Siehe Definition 4. 5, 6, 17, 21, 41, 42, 82, 99

Berechnungsfläche

Eine Fläche, die beregnet werden soll. 5, 10, 11, 13, 15, 17, 21, 24, 42, 64, 79, 81, 84, 88

Berechnungsmaschine

Ein Berechnungssystem zum Beregnen von Flächen. Die Beregnung erfolgt durch das Ziehen einer Beregnungsvorrichtung entlang eines Schlauches. 3–17, 19, 21–26, 28–33, 36, 37, 39–42, 44, 46, 47, 50, 56, 58–61, 63, 65–75, 77–79, 81–86, 88–92

Beregnungspfad

Eine Liste von Koordinaten, welche der Reihe nach im Rahmen einer Beregnung abgefahren werden. Siehe Definition 1. 2, 5–8, 12–14, 18–26, 32, 35, 59, 60, 66, 70–74, 77, 79, 80, 85, 86, 90, 91, 99

Beregnungsplan

Eine zeitlich geordnete Liste von Beregnungsaufgaben. Siehe Definition 5. 1, 5–8, 14, 15, 17, 21, 22, 26, 82, 86

Beregnungssequenz

Eine Liste von Beregnungspfaden mit der Eigenschaft, dass aufeinander folgende Elemente, sowie das letzte und erste Element der Liste *zusammenhängend* sind. Siehe Definition 3. 1, 5–8, 14, 17, 20–24, 26, 32, 70, 79, 82, 86, 88, 90, 91, 99

Beregnungsvorrichtung

Eine Vorrichtung zum Zerstäuben von Wasser. 3, 11, 13, 14, 22, 99, 101

Controller

Ein programmierbarer, elektrischer Baustein mit Schnittstellen zur Anbindung von Hardwarekomponenten. 30, 31, 39–55, 58, 61, 62, 67, 69

DAC

Ein elektrischer Baustein, welcher aus einen gegebenen, diskreten Wert (digital) eine analoge Spannung erzeugt. 43, 44

dBASE-Datei

Eine Datei nach [3], welche im vorliegenden Fall Fahrgassen einer Bewirtschaftungsfläche spezifiziert. 18

DGPS

Ein Verfahren, beidem zwei GNSS-Empfänger genutzt werden, um atmosphärische Störungen zu korrigieren. 30, 63, 64, 102

EKF

Ein Sensorfusionsfilter, der auf dem Kalman Filter basiert und die Fusion von nicht-linearen Sensorwerten ermöglicht. 59, 66

GIS

Ein Informationssystem zur Darstellung und Verarbeitung von Geoinformationstechnischen Daten. 18

GND

Abkürzung für Ground (English) und bezeichnet die elektrische Masse, beziehungsweise den elektrischen Minuspol einer Spannungsquelle. 42, 44, 46, 47, 53

GNSS

Ein globales Navigationssatellitensystem ist ein System zur Bestimmung der Position auf der Erde unter Verwendung von Satelliten. 16, 28–30, 60, 63–65, 85, 100, 102

GPS

Ein vom US-Verteidigungsministerium entwickeltes und betriebenes globales Navigationssatellitensystem. 18

Haspel

Eine Vorrichtung zum Aufwickeln von langgestreckten Materialien wie Schläuchen, Garnen und Seilen [57]. 3, 4, 7, 12, 13, 28, 29, 61, 63

Hodometer

Ein oder mehrere Sensoren, welche durch Messen von mechanischen Bewegungen eine zurückgelegte Strecke erfassen. 61

I2C

Ein nach [58] synchroner, serieller Zweidraht-Bus, welcher die Kommunikation zwischen mindestens einem Master und bis zu 127 Slaves über kurze Datenleitungen ermöglicht. 31, 51, 53, 67

IMU

Ein System aus Beschleunigungs- und Drehratensensoren zur Erfassung von Trägheitsmomenten. 16, 31, 66, 67

IP

Ein Netzwerkprotokoll. 30

Linearantrieb

Ein Zylinder mit einem innenliegenden Koblen, welcher mit einer Motor-Getriebe-Kombination elektrisch ein- und ausgefahren werden kann. 2, 29, 38, 39, 44, 46, 47, 53–58, 61, 72, 85

LTE

Der Standard für den Mobilfunk in dritter Generation. 31, 34, 69

LuT

Eine Tabelle, die zu einer Eingabe innerhalb eines definierten Bereichs immer eine Ausgabe liefert. 43, 44

NTRIP

Ein Protokoll zur Übertragung von RTCM-Nachrichten über das Internet-Protokoll. 28, 30, 33, 35, 64, 65

Open Street Map

Ein Projekt zur Schaffung einer freizugänglichen Weltkarte. 18

Potentiometer

Ein einstellbarer, ohmscher Widerstand. Je nach Bauart führt eine Dreh- oder Linearbewegung zur Veränderung des Widerstands innerhalb eines festen Bereichs. 16, 44, 47, 54, 58, 61

Rechtecksignal

Ein periodischer Spannungsverlauf über die Zeit, welcher näherungsweise nur zwei Werte annimmt. 48–50, 52, 58, 62, 102

Regenwagen

Eine mobile Beregnungsvorrichtung. 3, 4

REP

Eine Ansammlung von Konventionen, die eine einheitliche Entwicklung von ROS Komponenten ermöglichen soll. 36, 60, 66

ROS

Eine Sammlung aus Software-Bibliotheken und Werkzeugen, welche dabei Helfen, Applikationen und Systeme der Robotik zu entwickeln. 36, 51, 66, 102

RTCM

Organisation zur Schaffung von Grundlagen und Normungen im Bereich der maritimen Telekommunikation. 30, 33, 35, 64, 65, 101

RTK

Ein Verfahren, basierend auf DGPS, bei dem die Trägersignale der GNSS zur Bestimmung der Position verwendet werden. 29–31, 33–35, 59, 64–66, 69, 70, 102

SAPOS

Ein Anbieter von RTK-Korrektursignalen. 64

Shape-Datei

Eine Datei nach [3], welche im vorliegenden Fall Fahrgassen einer Bewirtschaftungsfläche spezifiziert. 8, 18–21, 35, 64, 69, 70, 81, 84, 88

Spannungsteiler

Eine Schaltung aus in Reihe geschalteten Widerständen um eine Spannung um einen definierten Faktor zu reduzieren. 42

Tastgrad

Gibt nach [30] das Verhältnis von Impulsdauer zu Periodendauer eines Rechtecksignals an. 48, 55, 57, 58

UART

Eine asynchrone, serielle Schnittstelle mit einer Wortlänge von fünf bis neun Bit, welche auch eine synchrone Datenübertragung unterstützt. 30, 31, 34

USB

Ein serielles Bussystem, über welches Geräte miteinander verbunden werden können. 30, 31, 34

UTM

Ein Koordinatensystem, welches die Erdoberfläche in mehrere Zonen aufteilt und diese als kartesisches Koordinatensystem abbildet. 66

WGS84

World Geodetic System 1984 ist ein geodätisches Referenzsystem zur Positionsbeschreibung auf der Erde. 18, 19, 23, 59, 60, 66, 88

Die Darstellung der als Prüfungsleistung zu bewertenden Beiträge der einzelnen Bearbeiter erfolgt über die Zuordnung von Kapitel bzw. Abschnitt zum Autor in Tabellenform. Die Autoren werden gemäß ihren vollständigen Namen, Marvin Banse (MB) und Florian Schmalriede (FS), abgekürzt.

Kapitel/ Abschnitt	Autor	Kapitel/ Abschnitt	Autor
1	FS	5.2.1	MB
1.1	FS	5.2.2	MB
1.2	FS	5.3	MB
1.3	FS	5.3.1	MB
1.4	FS	5.3.2	MB
1.5	FS	5.3.3	MB
2	FS	5.4	FS
2.1	FS	5.4.1	MB
2.2	FS	5.4.2	FS
2.3	MB	5.4.3	FS
3	FS	5.4.4	MB
3.1	FS	5.5	FS
3.2	FS	5.6	FS
3.3	FS	5.6.1	FS
3.4	FS	5.6.2	FS
3.5	FS	5.6.3	FS
4	FS	5.6.4	FS
4.1	MB	5.6.5	FS
4.1.1	MB	5.7	MB
4.1.2	MB	6	MB
4.2	MB	7	MB
4.2.1	MB	8	MB
4.2.2	MB	A	MB
5	MB	B	FS
5.1	MB	C	MB
5.1.1	MB	C.1	MB
5.1.2	FS	C.2	MB
5.2	MB	C.3	MB

Hiermit versichere ich an Eides statt, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

Marvin Banse

Florian Schmalriede