



KLAUS MOSTEGEL, BSc.

Entwicklung eines Algorithmus zur optimierten Steuerung eines Nivelliersystems

Masterarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium: Maschinenbau

eingereicht an der

Technischen Universität Graz

Betreuer

Dipl.-Ing. Dietmar Neubacher

Institut für Maschinenbau- und Betriebsinformatik
Vorstand: Univ.-Prof. Dipl.-Ing. Dr.techn. Siegfried Vössner

Graz, September 2017

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift

Abstract

Today most caravans and campers are stabilized and leveled using manual driven pillars. As users are today well equipped with smart devices, an Austrian start-up company decided to make use of these gadgets in order to control the stabilization and leveling process. To do so, it is necessary to generate a fully automated stabilization system. The aim of this thesis is to build an abstracted mechanical model of the vehicle in order to determine the involved forces and to develop algorithms to guide the process. Such leveling systems allow a more easy and comfortable use and much more functionalities. Nevertheless a combination of mechanical, electrical and information systems is needed and results in a more complex development process. Such projects tend to be multidisciplinary by definition and the engineering process has to be systematically guided in order to achieve proper results. Such procedure models are heavily used in many different industries, like the principle of Simultaneous Engineering in Automotive or the Scrum concept in Software Engineering. In course of this thesis the System Engineering concept is used and the development process is guided by the V-Model, based on the VDI2206 standards. The results of the mechanical model and a preceding requirements definition has indicated that the involved forces have to be restricted in order to guarantee a safe leveling. Subsequently, the algorithms were designed to avoid wrong loads and undefined conditions. In order to evaluate the performance, a physical prototype was build and tested. Additionally a smartphone application was developed during an interlinked bachelor project and used to control the prototype. Finally the entire prototype was presented to a famous caravan and camper producer and has been very well accepted.

Kurzfassung

Aktuell werden die meisten Wohnmobile und Wohnwägen mit manuell angetriebenen Stützen nivelliert und stabilisiert. Da die Benutzer heute mit Smartphones ausgestattet sind, hat sich ein österreichisches Start-up-Unternehmen entschieden, diese Geräte für die Steuerung des Stabilisierungs- und Nivellierungsprozesses einzusetzen. Dazu ist es notwendig ein vollautomatisches Stabilisierungssystem zu entwickeln. Das Ziel dieser Arbeit ist es, ein abstraktes mechanisches Modell des Fahrzeugs aufzubauen, um die beteiligten Kräfte zu bestimmen und Algorithmen zu entwickeln, die den Prozess steuern. Ein solches Nivellierungssystem ermöglicht eine einfachere und komfortable Bedienung sowie zusätzliche Funktionalitäten. Dafür ist eine Kombination aus mechanischen, elektrischen und Informationssystemen erforderlich, womit ebenfalls eine erhöhte Komplexität im Entwicklungsprozess mit einhergeht. Solche Projekte sind definitionsgemäß interdisziplinär. Der Entwicklungsprozess muss dabei systematisch geführt werden, um optimale Ergebnisse zu erzielen. Vorgehensmodelle dafür werden in verschiedenen Bereichen angewendet, wie das Prinzip des Simultaneous Engineering in der Automobilindustrie oder Scrum in der Softwareentwicklung. Im Rahmen dieser Arbeit wird das Konzept des Systems Engineering angewendet und der Entwicklungsprozess entsprechend der Richtlinie VDI2206 aufgebaut. Die Ergebnisse des mechanischen Modells und die vorhergehende Anforderungsdefinition haben gezeigt, dass die beteiligten Kräfte eingeschränkt werden müssen, um eine sichere Nivellierung zu gewährleisten. Infolgedessen wurde ein Algorithmus entworfen, der falsche Belastungen und undefinierte Zustände vermeidet. Zur Bewertung der Funktionsfähigkeit wurde ein physischer Prototyp erstellt und getestet. Zusätzlich wurde eine Smartphone-Applikation in einem separaten Bachelorprojekt entwickelt, die zur Steuerung des Prototyps verwendet wird. Der gesamte Prototyp wurde schließlich einem berühmten Wohnwagen- und Wohnmobil-Produzenten präsentiert und sehr gut angenommen.

Inhaltsverzeichnis

Abstract	iii
1. Einleitung	1
2. Mechatronische Systeme - Theoretische Grundlagen	3
2.1. Definition Mechatronik	3
2.2. Anforderungen des mechatronischen Systementwurfs	4
2.3. Grundstruktur eines mechatronischen Systems	5
2.4. Domänenspezifischer Entwurf	6
2.4.1. Mechanisches System	6
2.4.2. Elektrisches System	13
2.4.3. Informationssystem	16
2.5. Testen von Systemen	17
3. Systems Engineering	19
3.1. Definition	19
3.1.1. Was ist ein System?	20
3.1.2. Was ist Systems Engineering?	21
3.1.3. V-Modell	23
3.1.4. Vorgehensmodell als Mikrozyklus	25
3.2. Systems Modeling Language	28
3.2.1. Use Case Diagramm	29
3.2.2. State Machine Diagramm	31
4. Mechatronische Nivellierungssysteme	34
4.1. Selbstblancierende Vehikel	34
4.2. Hochpräzise selbstnivellierende Plattform	36

5. Vorstudie	38
5.1. Projektbeschreibung	38
5.2. Anforderungen	39
6. Hauptstudie	45
6.1. Mechanisches Modell	45
6.1.1. Lastabschätzung	45
6.1.2. Dynamisches Modell	54
6.1.3. Sonderfall statisches Modell	69
6.1.4. Einzelstütze	75
6.2. Nivellierungsalgorithmus	82
6.2.1. Algorithmusaufbau	83
6.2.2. Bestandteile des Algorithmus	88
7. Systemintegration	96
7.1. Aufbau und Komponenten	97
7.1.1. Mechanische Komponenten	97
7.1.2. Elektrische Komponenten	97
7.1.3. Informationsverarbeitung	100
7.2. Benutzerschnittstelle	100
7.3. Test und Bewertung von Algorithmen	103
7.4. Messung der tatsächlichen Stromaufnahme	103
7.5. Validierung des mechanischen Modells	104
8. Resümee	111
8.1. Ausblick	113
Literatur	114
A. Datenblätter	117
A.1. Datenblatt Hallsensor	117
B. Programmcode	132

Abbildungsverzeichnis

2.1. Felder der Mechatronik	4
2.2. Grundstruktur eines mechatronischen Systems	6
2.3. Einmassenschwinger	8
2.4. Modellierung eines Fahrzeugs	9
2.5. Ersatzschaltbild Gleichstrommaschine	14
2.6. Kennlinien einer Gleichstrommaschine	15
2.7. Blockdiagramm Mikrocontroller	16
3.1. System	20
3.2. Problemdefinition	21
3.3. Methodik im Kontext der Problemlösung	21
3.4. Konzept des Systems Engineering	23
3.5. Projektphasen	24
3.6. V-Modell	25
3.7. Problemlösungszyklus	27
3.8. Grundelemente Use Case Diagramm	29
3.9. Verbindungen Use Case Diagramm	30
3.10. Elemente State Machine Diagramm	32
4.1. Segway	35
4.2. Segway als inverses Pendel	35
4.3. Freischnitt und Prinzipskizze	37
5.1. Use Case Diagram: Automatische Nivellierung	41
6.1. Skizze Wohnwagen	46
6.2. Skizze Wohnwagen mit Stützen	47

6.3. Skizze Wohnwagen mit Schnittebenen	49
6.4. Qualitative Streckenlast Schnittebene A-A	49
6.5. Qualitative Streckenlast Schnittebene B-B	50
6.6. Qualitative Streckenlast Schnittebene C-C	50
6.7. Qualitative Streckenlast Schnittebene D-D	50
6.8. Skizze Schwerpunktabschätzung	52
6.9. Lage des Schwerpunktes	53
6.10. Stuetzennummerierung	55
6.11. Stützenmodell	55
6.12. Wagenkoordinatensystem	56
6.13. Bewegung des Wohnwagens	67
6.14. Stützkräfte	68
6.15. Kräfte in der x-y-Ebene	68
6.16. Statisches Stützenmodell	70
6.17. Kräfte und Angriffspunkte	73
6.18. Kräfteverläufe beim Nicken	74
6.19. Kräfteverläufe beim Nicken	74
6.20. Geometrie einer Stütze	76
6.21. Freischnitt der Stütze	77
6.22. Anfahrt- und Betriebsmoment Einzelstütze	82
6.23. Zustandsdiagramm Algorithmus	85
6.24. Vergleich Vorhersagemethoden	91
6.25. Zeitbereiche Stütze	94
7.1. Prototyp	96
7.2. AL-KO E.C.S Elektronische Stütze	98
7.3. H-Brücke	99
7.4. Benutzerschnittstelle	102
7.5. Benutzerschnittstelle Messapplikation	104
7.6. Stützenposition Prototyp	106
7.7. Vergleich Kraft RB	108
7.8. Vergleich Kraft LB	109
7.9. Vergleich Kraft RF	109
7.10. Vergleich Kraft LF	110

1. Einleitung

Die zunehmende Digitalisierung bringt Informationstechnik in verschiedenste Anwendungsgebiete, zum Einen der Industrie (Stichwort „Industrie 4.0“), zum Anderen aber auch in alltäglichen Anwendungen. Es werden selbst fahrende Autos entwickelt, der Staubsaugerroboter reinigt die Wohnung und eine Raumklimaanlage sorgt für angenehme Temperaturen, um nur einige wenige Beispiele zu nennen. Mechatronik umgibt uns ständig, ob bewusst oder unbewusst.

Zusätzlich zu einem großen Anwendungsgebiet ist die Mechatronik in einem Spannungsfeld von drei Disziplinen angesiedelt: Dem Maschinenbau, der Elektrotechnik und der Informationstechnik. Durch dieser verschiedenen Disziplinen ist die Entwicklung eines mechatronischen Systems komplizierter, als der Entwicklungsprozess in einer Disziplin für sich. Systems Engineering (Haberfellner u. a., 2012) bietet einen Ansatz, komplizierte Systeme und Projekte systematisch zu betrachten und zu bearbeiten. Es kann für verschiedene Anwendungsgebiete sinnvoll sein, Systems Engineering zu betreiben, so auch bei der Entwicklung von mechatronischen Systemen. Speziell für dieses Anwendungsgebiet wurde die VDI2206 (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004) entwickelt, die sich rein mit der Entwicklung von mechatronischen Systemen auseinandersetzt. In der VDI2206 (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004) sind die wichtigsten Elemente des Systems Engineering für mechatronische Systeme eingearbeitet.

Die Firma Smart Lift GmbH beschäftigt sich mit der Nivellierung und Stabilisierung von Wohnwägen und Wohnmobilen. Sowohl Nivellierung als auch Stabilisierung werden bei einem handelsüblichen Wohnwagen oder Wohnmobil mit mechanischen Stützen, die manuell betrieben werden, realisiert. Einige wenige Modell sind mit elektrischen oder hydraulischen Stützen ausgestattet. Besonders bei der manuellen

Variante ist ein erheblicher körperlicher Aufwand mit der Nivellierung und Stabilisierung verbunden. Zusätzlich muss bei der Nivellierung die Lage des Wohnwagens oder Wohnmobils mit zusätzlichen Hilfsmitteln bestimmt werden. Hier besteht großes Potential diese Vorgänge zu automatisieren und so komfortabler für den Nutzer zu gestalten.

Die Firma Smart Lift GmbH strebt die Entwicklung eines mechatronischen Systems zur automatischen Nivellierung von Wohnwägen und Wohnmobilen an. Dieses System soll einen Satz von elektrisch betriebenen Stützen so Ausfahren, dass eine waagrechte Lage des Fahrzeugs zustande kommt. Neben der Lage ist ein sicherer Stand eine wichtige Eigenschaft, die es zu erreichen gilt. Im Zuge dieser Arbeit wird ein Teil der Entwicklung für dieses System durchgeführt. Am Ende steht ein mechanisches Modell und ein Algorithmus für die Nivellierung.

2. Mechatronische Systeme - Theoretische Grundlagen

In diesem Kapitel werden die Grundlagen für die weiteren Entwicklungsschritte behandelt und ein methodisches Vorgehen eingeführt. Das zu entwickelnde System besteht aus Mechanische, elektrische und Softwarekomponenten. Damit liegt ein mechatronisches System vor. Die Entwicklung erfolgt in allen drei genannten Disziplinen. Zur Lösung einer so vielschichtigen Aufgabe wurden Vorgehensmodelle entwickelt, die einen entsprechenden Gestaltungsprozess beschreiben und vereinfachen. Angewendet wird das Modell des Systems Engineering (Haberfellner u. a., 2012) und die Richtlinie VDI2206 (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004).

2.1. Definition Mechatronik

Der Begriff Mechatronik wurde ursprünglich in Japan geprägt, als eine Zusammensetzung der Worte Mechanik und Elektrotechnik (vgl. Roddeck, 2012). Zu diesem Zeitpunkt waren auch nur die genannten Disziplinen in der Mechatronik involviert. Jedoch durch die fortschreitende Entwicklung in der Informationstechnik, ist heute unter dem Begriff Mechatronik das Zusammenwirken von Maschinenbau, Elektrotechnik und Informatik zu verstehen. Mit diesen interdisziplinären Möglichkeiten ergeben sich vielseitige Lösungsmöglichkeiten für bekannte Probleme und neue Anwendungsmöglichkeiten, die in einer einzelnen Teildisziplin nicht umsetzbar wären. Die Integration der Informatik ermöglicht es zusätzlich Systemschnittstellen

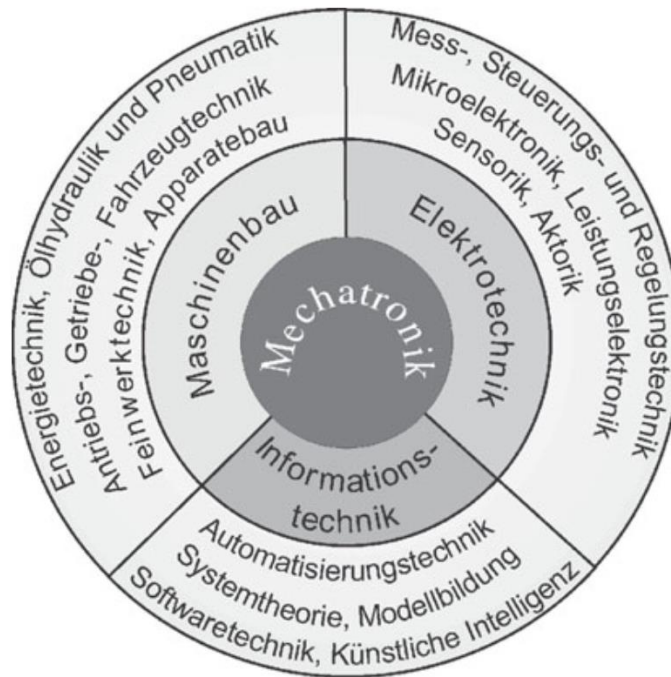


Abbildung 2.1.: Felder der Mechatronik (Roddeck, 2012, S. 4)

zu elektronischen Datenverarbeitungssystemen zu schaffen. Eine einfache Überwachung und Steuerung des Systems von außen ist damit einfach umsetzbar. Das ist einer der Hauptgründe für die immer größer werdende Verbreitung im Vergleich zu konventionellen Systemen.

2.2. Anforderungen des mechatronischen Systementwurfs

Neben den Vorteilen, die ein mechatronisches System bietet, gehen neue Herausforderungen mit einher. Die Entwicklung muss unter Berücksichtigung aller drei erwähnten Disziplinen stattfinden. Dieses interdisziplinäre Vorgehen birgt neben den Chancen auch Risiken, die nicht unterschätzt werden dürfen. Beispielsweise ist es nicht zweckmäßig eine optimale Lösung in jeder Teildisziplin zu suchen. Zu einem Gesamtsystem zusammengesetzt, ergibt das meist nicht die optimale Gesamtlösung

oder sogar kein funktionsfähiges System. Für ein mechatronisches System muss also neben den Teillösungen ein entsprechendes Gesamtkonzept verfolgt werden. Die Gestaltung einer solchen Entwicklung zeigt die VDI2206 (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004). Das darin vorgestellte Vorgehensmodell besteht aus den drei Elementen: Mikrozyklus, Makrozyklus und vordefinierten Prozessbausteinen. Der Mikrozyklus ist auf Basis des Systems Engineering aufgebaut, auf das in Kapitel 3 eingegangen wird. Der Makrozyklus folgt ebenfalls dem Systems Engineering und wird im Unterkapitel 3.1.3 behandelt.

2.3. Grundstruktur eines mechatronischen Systems

Wie bereits eingangs beschrieben, sind in einem mechatronischen System die drei Disziplinen Maschinenbau, Elektrotechnik und Informationstechnik vereint. Diese spiegeln sich ebenfalls in der Grundstruktur eines solchen Systems wider, wie Abbildung 2.2 zeigt. Die mechanischen Komponenten sind durch das Grundsystem repräsentiert, die Aktoren und Sensoren werden den elektronischen Komponenten zugeordnet und die Informationsverarbeitung gehört zum Bereich der Informationstechnik. Weiter sind die verschiedenen Flüsse zwischen den Komponenten dargestellt: Der Informations-, der Energie- und der Stofffluss. Betrachtet man die Abbildung 2.2 im Gesamten, so erkennt man, dass ein mechatronisches System in vielen Punkten einem regelungstechnischen System entspricht. Der Grad der Entsprechung hängt dabei vom jeweiligen Anwendungsgebiet ab. In vielen Fällen wird, wie bei einer Regelung, ein Zielzustand bzw. eine Eingangsgröße vorgegeben, die mit der Regelung erreicht werden soll. Bei einem mechanistischen System sind die regelungstechnischen Aspekte also mit zu betrachten.

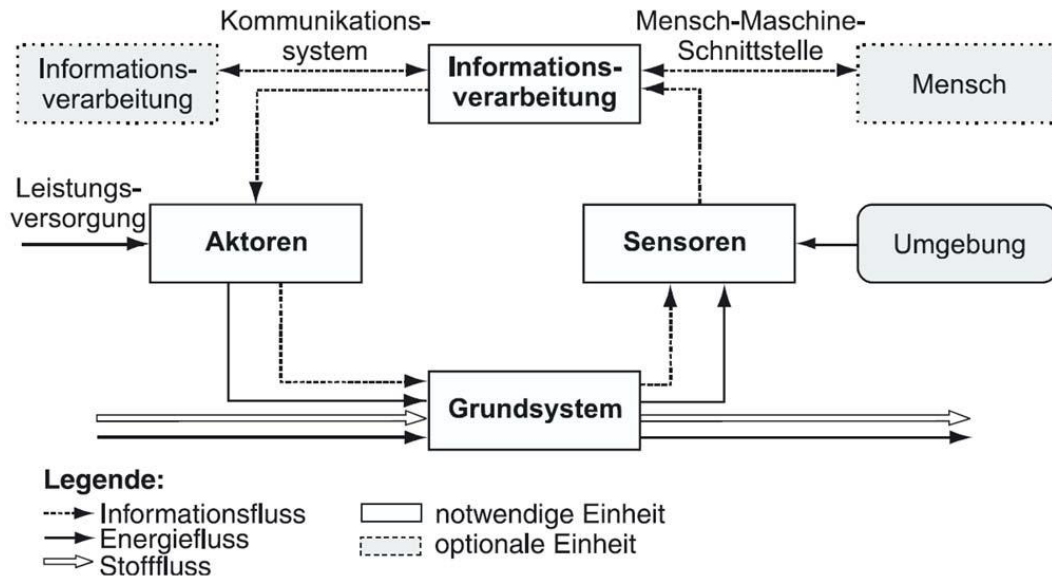


Abbildung 2.2.: Grundstruktur eines mechatronischen Systems (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004, S. 14)

2.4. Domänenspezifischer Entwurf

Wenn ein Gesamtkonzept definiert ist, muss dieses in den Bereichen Maschinenbau, Elektrotechnik und Informatik umgesetzt werden. Die Teillösungen werden aus der Gesamtlösung abgeleitet und gesondert in der jeweiligen Domäne entwickelt, bevor sie schließlich wieder in ein Gesamtsystem integriert werden, gemäß dem Vorgehen in der VDI2206 (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004) (siehe Kapitel 3.1.3). Im Folgenden wird auf die einzelnen Teilsysteme Mechanik, Elektrik und Informatik und die Modelle näher eingegangen.

2.4.1. Mechanisches System

Für die mechanischen Komponenten ist es essentiell die wirkenden Belastungen zu kennen, um diese entsprechend auslegen zu können. Die Ermittlung dieser Belastungen erfolgt mit einem dynamischen Modell. Die theoretischen Grundlagen für dieses Modell werden in diesem Abschnitt behandelt.

Modellbildung

Die Modellbildung baut auf den Grundelementen Feder, Dämpfer und Masse auf. Diese Elemente lassen sich mathematisch beschreiben. Die Eigenschaften sind hier angeführt:

Feder Eine Feder ist für die folgenden Betrachtungen als lineare Umsetzung zwischen Kraft und Weg definiert. Die mathematische Beschreibung folgt der Gleichung

$$F_{Feder} = c_{Feder} * x, \quad (2.1)$$

wobei x den Federweg aus dem unbelasteten Zustand darstellt und c_{Feder} die Federkonstante, die den Umrechnungsfaktor zur Federkraft darstellt. F_{Feder} ist die wirkende Federkraft.

Dämpfer Ein Dämpfer ist ähnlich wie die Feder definiert, an die Stelle des Weges tritt allerdings die Geschwindigkeit. Die beschreibende Gleichung nimmt die Form

$$F_{Dämpfer} = d_{Dämpfer} * \dot{x} \quad (2.2)$$

an. \dot{x} ist die Längenänderung des Dämpfers, $d_{Dämpfer}$ die Dämpferkonstante und $F_{Dämpfer}$ wiederum die wirkende Dämpfungskraft.

Masse Bei einem Massenelement gibt es verschiedene Möglichkeiten ein solches Element zu modellieren. Die gesamte Masse kann in einem Punkt konzentriert und die Mässsenträgheitsmomente vernachlässigt werden. Das ist für dieses Modell nicht zielführend. Eine Masse wird in weitere Folge als Element mit der Masse m und einem Trägheitstensor Θ angesehen.

Aus diesen Elementen kann ein dynamisches Modell erstellt werden. Als Verbindungen werden zusätzlich masselose Elemente verwendet. Sie dienen zur Darstellung von Längenverhältnissen und Anlenkpunkten. Als einfaches Beispiel zur Kombination dieser Elemente ist ein Einmassenschwinger in Abbildung 2.3 dargestellt.

Bei der Modellierung von mechanischen Systemen muss ein Optimum zwischen Detailgrad und Vereinfachung gefunden werden. Je nach Anforderungen können verschiedene Modelle sinnvoll sein. Roddeck (2012) beschreibt vier Möglichkeiten, ein Fahrzeug zu modellieren (siehe Abbildung 2.4). Der Unterschied zwischen den

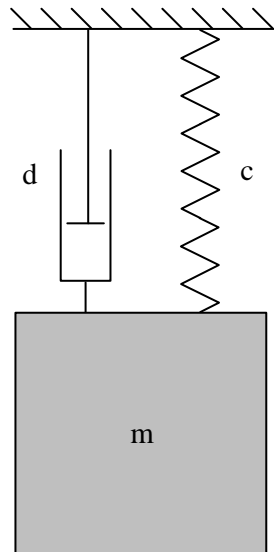


Abbildung 2.3.: Einmassenschwinger

Modellen besteht in den Freiheitsgraden. Während in Möglichkeit a nur ein Freiheitsgrad vorgesehen ist, sind es bei Möglichkeit d fünf Freiheitsgrade. Mit fünf Freiheitsgraden lassen sich die Vertikalbewegung, die Bewegung der beiden Räder, das Nicken des Fahrzeugs und die erste Eigenform darstellen. Diese Berechnung bietet viel Informationen, verlangt aber einen entsprechenden rechnerischen Aufwand bei der Auswertung. In Möglichkeit a ist nur ein Freiheitsgrad vorhanden, der die Vertikalbewegung darstellt. Dieses Modell ist mit einem viel geringeren Aufwand zu berechnen. Sollten nicht alle Informationen für die Berechnung relevant sein, ist es also ratsam, die Freiheitsgrade zu reduzieren.

An diesem Beispiel ist die Bedeutung der anforderungsgerechten Modellbildung gut zu erkennen. Einerseits soll der Berechnungsaufwand sehr niedrig gehalten werden und andererseits muss das Modell alle Abbildungsanforderungen, die daran gestellt werden, erfüllen. Ein Einmassenschwinger ist als Modell für ein Fahrzeug bereits ausreichend.

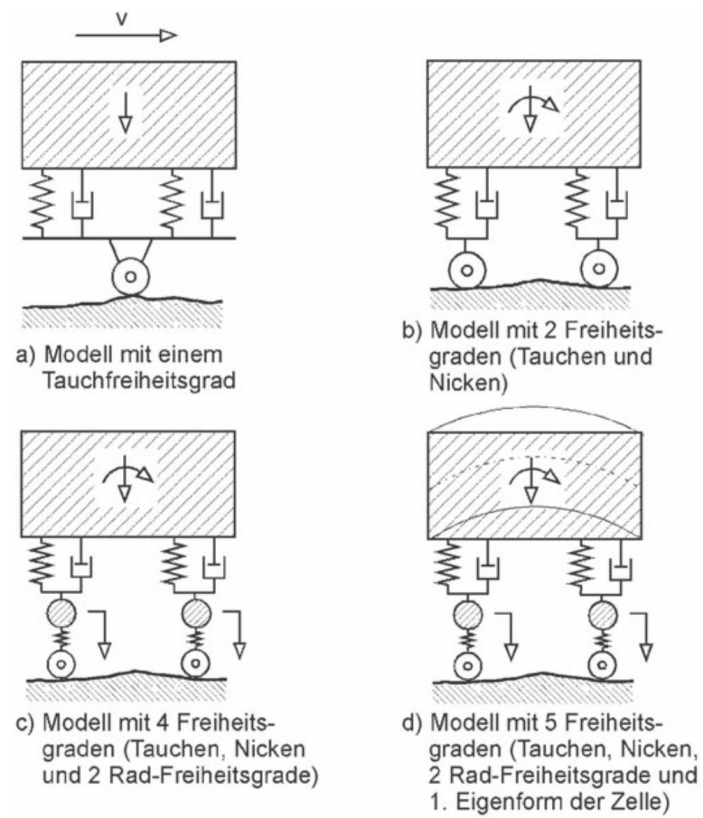


Abbildung 2.4.: Möglichkeiten zur Modellierung eines Fahrzeugs (Roddeck, 2012)

Aufstellen der Bewegungsgleichungen

Das mechanische Modell dient dazu die Bewegungsgleichungen zu formulieren. In den Bewegungsgleichungen werden die mechanischen Zusammenhänge zwischen den Bewegungsgrößen mathematisch festgehalten. Zum Aufstellen der Bewegungsgleichungen gibt es verschiedene gängige Methoden, wie beispielsweise

- Newton-Euler Methode,
- Prinzip von d'Alembert,
- Lagrangesche Gleichung 2.Art,

um nur drei zu nennen. Im Hinblick auf die Anforderungen an das mechanische Modell und die erwarteten Lösungen wird die Newton-Euler Methode angewendet. Bei dieser werden alle wirkenden Kräfte im Zuge der Rechnung direkt mit ermittelt, bei anderen Methoden müssten diese separat errechnet werden. Rill und Schaeffer (2014) führen eine detaillierte Beschreibung der drei Methoden durch. Zur Aufstellung der Bewegungsgleichungen werden bei der Newton-Euler Methode Impuls- und Drallsatz verwendet. Diese sind in Gleichung 2.3 und Gleichung 2.5 dargestellt.

Impulssatz Der Impulssatz beschreibt die Impulsänderung eines Körpers und lässt sich als

$$\frac{d}{dt}\mathbf{p} = \mathbf{F}_a \quad (2.3)$$

formulieren. Die Impulsänderung $\frac{d}{dt}\mathbf{p}$ erfolgt durch die äußeren Kräfte \mathbf{F}_a . Der Impuls selbst ist für einen Körper mit der Masse m als Produkt von Masse und Geschwindigkeit definiert:

$$\mathbf{p} = mv_{0S} \quad (2.4)$$

Dabei beschreibt v_{0S} die Geschwindigkeit des Schwerpunktes S gegenüber dem Inertialsystem 0 .

Drallsatz Analog zum Impulssatz beschreibt der Drallsatz die Dralländerung eines Körpers K . Die Gleichung für den Drallsatz hat, auf den Schwerpunkt des Körpers bezogen, die Gestalt:

$$\frac{d}{dt}\mathbf{L}_S = \mathbf{M}_S \quad (2.5)$$

Die Dralländerung wird durch die wirkenden Momente \mathbf{M}_S um den Schwerpunkt S bewirkt. Der Drall eines Körpers K lässt sich formulieren als:

$$\mathbf{L}_S = \int_K \begin{pmatrix} \omega_x(y^2 + z^2) - \omega_y xy - \omega_z xz \\ -\omega_x xy + \omega_y(x^2 + z^2) - \omega_z yz \\ -\omega_x xz - \omega_y yz + \omega_z(x^2 + y^2) \end{pmatrix} dm \quad (2.6)$$

Mit dem Trägheitstensor

$$\Theta_S = \begin{bmatrix} \int_K (y^2 + z^2) dm & -\int_K xy dm & -\int_K xz dm \\ -\int_K xy dm & \int_K (x^2 + z^2) dm & \int_K yz dm \\ \int_K xz dm & \int_K yz dm & \int_K (x^2 + y^2) dm \end{bmatrix} \quad (2.7)$$

lässt sich der Drall in vereinfachter Form darstellen als:

$$\mathbf{L}_S = \Theta_S \boldsymbol{\omega} \quad (2.8)$$

Dieser Satz von Gleichungen führt zu den Bewegungsgleichungen, indem man die äußeren Kräfte und Momente in geeigneter Weise aufstellt und in den Impuls- und Drallsatz einsetzt. (Magnus und Müller-Slany, 2005)

Lösen der Bewegungsgleichungen

Die Bewegungsgleichung ist im Allgemeinen eine Differentialgleichung 2. Ordnung in Abhängigkeit von der Zeit. Die Bewegungsgrößen des Systems werden durch die Lösung dieser Differentialgleichung erhalten. Bei einfachen Systemen kann eine analytische Lösung gefunden werden. Im Allgemeinen ist aber eine numerische Lösung der Gleichung notwendig. Dazu wird im Folgenden das Programm MATLAB verwendet. MATLAB bietet verschiedene Löser für gewöhnliche Differentialgleichungen. Einem Löser kann die Bewegungsgleichung mit den Anfangsbedingungen übergeben werden, als Ergebnis erhält man diskrete Punkte der Lösung. Die Lösung besteht aus den Bewegungsgrößen des Systems. Je nach Typ der Gleichung können manche Löser falsche Lösungen liefern oder sehr ineffizient arbeiten. Die Auswahl eines geeigneten Löser, kann oft nur durch Versuche ermittelt werden. Eine Auflistung der implementierten Löser ist in Tabelle 2.1 dargestellt. Die Befehle stammen aus der Dokumentation zu MATLAB R2016a der Mathworks, Inc. Verschiedene Quellen,

MATLAB-Befehl	Numerische Methode
<code>ode45</code>	explizites eingebettetes Runge-Kutta-Verfahren nach Dormand und Prince
<code>ode23</code>	explizites eingebettetes Runge-Kutta-Verfahren nach Bogacki und Shampine
<code>ode113</code>	Prädiktor-Korrektor-Verfahren nach Adams-Bashforth-Moulten
<code>ode15s</code>	Mehrschrittverfahren
<code>ode23s</code>	Einschrittverfahren nach Rosenbrock
<code>ode23tb</code>	implizites Runge-Kutta-Verfahren
<code>ode23t</code>	Trapezregel

Tabelle 2.1.: Lösungsverfahren für gewöhnliche Differentialgleichungen aus MATLAB (Shampine und Reichelt, 1997)

darunter Shampine und Reichelt (1997), empfehlen `ode45` als ersten Löser, den man mit wenigen Ausnahmen für ein Problem anwenden sollte. Ausnahmen bilden beispielsweise steife Systeme oder sehr hohe Genauigkeitsanforderungen an die Lösung.

Vereinfachung als statisches Modell

Zur Berechnung der statischen Kräfte kann das dynamische System vereinfacht werden. Für das statische Modell werden alle Geschwindigkeiten und Beschleunigungen als Null definiert. Damit können Gleichgewichtsbedingungen für das statische System formuliert werden zu:

$$\sum F = 0 \quad (2.9)$$

$$\sum M = 0 \quad (2.10)$$

Da die beiden Zeitableitungen per Definition zu Null gebracht wurden, ist es nicht erforderlich eine Differentialgleichung zu lösen. Es muss nur das Gleichungssystem, das durch Einsetzen der Kräfte in Gleichgewichtsbedingungen entsteht, aufgelöst werden. Die Berechnung des Systems erfolgt für die statische Ruhelage. (Magnus und Müller-Slany, 2005)

2.4.2. Elektrisches System

Das elektrische System besteht, wie eingangs erwähnt, aus den Aktoren und Sensoren. Sensoren sorgen für die Aufnahme der physikalischen Größen und deren Umwandlung in ein Messsignal, das mittels der weiteren Datenverarbeitung interpretiert und ausgewertet werden kann. Aktoren bieten die Möglichkeit mit einer entsprechenden Ansteuerung in das System einzugreifen und dessen Zustand zu verändern.

Sensoren

Die zu verwendenden Sensoren sind im Rahmen des Projekts festgelegt. Es wird ein Beschleunigungs- und Gyrosensor zur Lagebestimmung verwendet und Hallsensoren zur Messung der Stromaufnahme der Aktoren.

Lagesensor Als Lagesensor kommt eine Kombination aus Gyrometer und Beschleunigungssensor zum Einsatz. Die Sensortypen entsprechen Standardtypen, die im Smartphonebereich verbaut werden. Für Details zur Funktionsweise sei auf einschlägige Fachliteratur verwiesen.

Hallsensor Die Strommessung erfolgt mit Hallsensoren, die mit dem namensgebenden Halleffekt eine Strommessung ermöglichen. Vereinfacht formuliert erzeugt der Hallsensor, wenn er mit Strom durchflossen wird, eine messbare Spannung, die Hall-Spannung. Die erzeugte Spannung wächst mit dem angelegten Strom. Eine genauere Beschreibung kann der entsprechenden Fachliteratur entnommen werden. (Tränkler und Fischerauer, 2014) (Bumiller und Tkotz, 2016)

Aktoren

Als Aktor können verschiedene Wirkprinzipien genutzt werden. Die Rahmenbedingungen, darunter die Aktoren, sind jedoch vom Projektpartner vorgegeben, der **Gleichstrommaschinen** verbauen möchte. Im Detail handelt es sich hierbei um Gleichstromnebenschlussmaschinen. Das Ersatzschaltbild einer solchen Maschi-

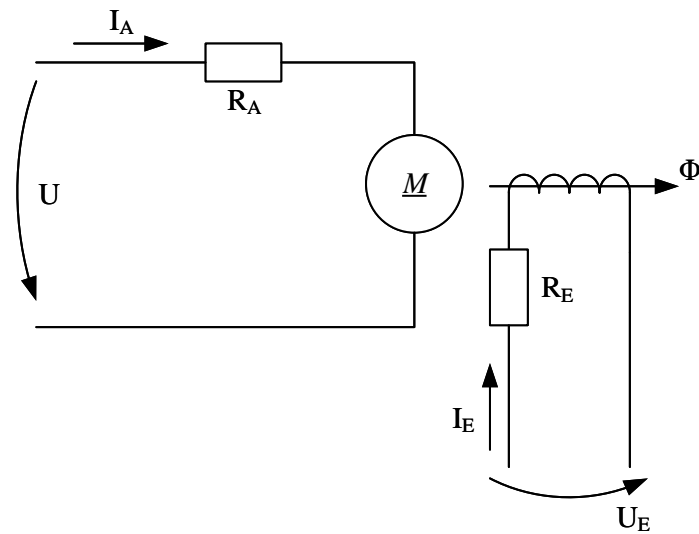


Abbildung 2.5.: Ersatzschaltbild Gleichstrommaschine (Busch, 2011)

ne ist angelehnt an Busch (2011) in Abbildung 2.5 dargestellt. Bezeichnungen mit dem Index A beziehen sich dabei auf den Anker und mit dem Index E auf die Erregerwicklung. Wichtige Information über das Motorverhalten im Einsatz bieten die Kennlinien, die in Abbildung 2.6 qualitativ dargestellt sind (entnommen aus Busch, 2011). Bei der näheren Betrachtung der Kennlinien erkennt man, dass die Drehzahl nur in geringem Maße vom Belastungsmoment und Ankerstrom abhängt, wohingegen zwischen dem Moment und dem Ankerstrom ein direkt proportionaler Zusammenhang besteht. Diese Zusammenhänge lassen sich nach Busch (2011) in mathematischer Form ausdrücken, als:

$$M = c\Phi I_A \quad (2.11)$$

$$n = \frac{U}{2\pi c\Phi} - \frac{R_A}{2\pi(c\Phi)^2} M = n_0 - \Delta n \quad (2.12)$$

Die in der Gleichung 2.11 und 2.12 verwendeten Symbol sind in Tabelle 2.2 aufgeschlüsselt und erklärt. Laut Busch (2011) bewegt sich die Änderung der Nenndrehzahl von der Leerlaufdrehzahl bei Nennmoment bei einer 1 kW Maschine in einem Bereich von 6...8%. Man kann hier also von einer sehr geringen Abweichung sprechen.

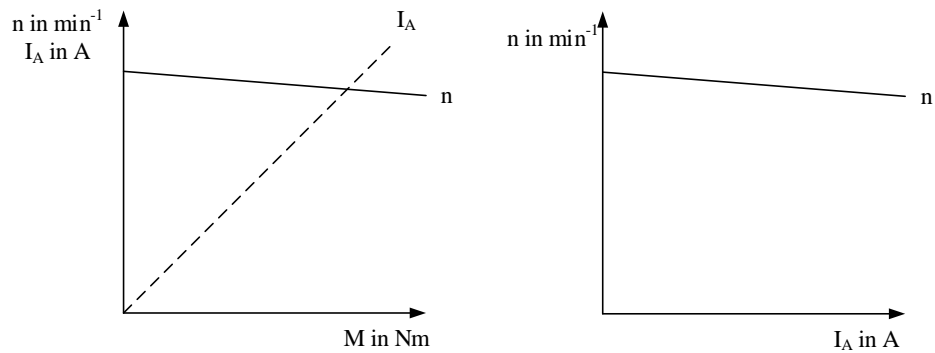


Abbildung 2.6.: Kennlinien einer Gleichstrommaschine (Busch, 2011)

Symbol	Einheit	Bedeutung
M	Nm	Drehmoment
n	U/s	Drehzahl
n_0	U/s	Leerlaufdrehzahl
Δn	U/s	Drehzahlabfall
c	-	Maschinenkonstante
Φ	Vs	Fluss der Hauptpole
I_A	A	Ankerstrom
U	V	Spannung
R_A	Ω	Ankerwiderstand

Tabelle 2.2.: Berechnung der Gleichstrommaschine - Symbole

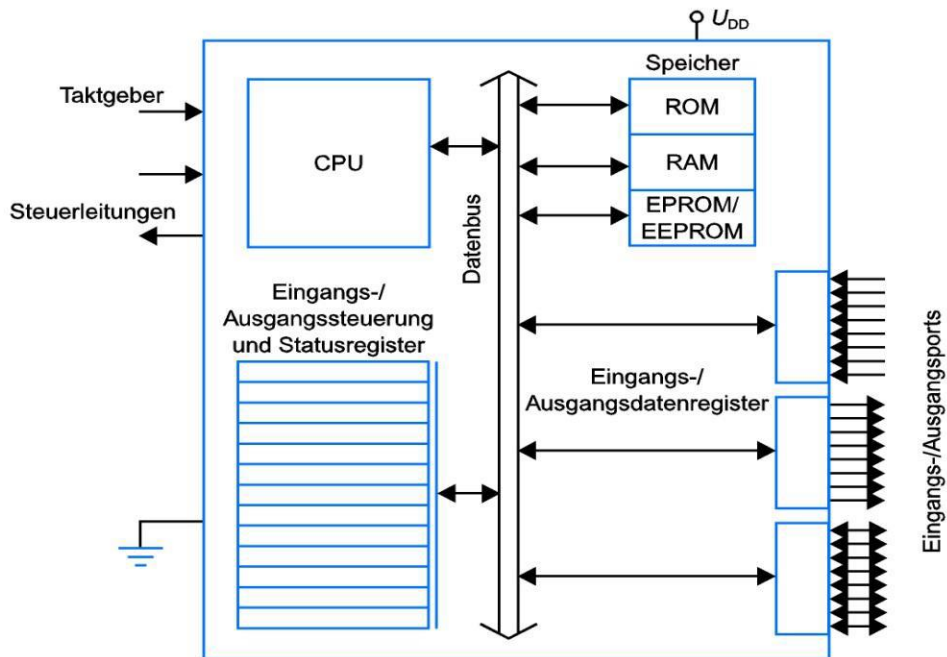


Abbildung 2.7.: Blockdiagramm Mikrocontroller (Bolton, 2004)

2.4.3. Informationssystem

Im Informationssystem werden die Daten aus den Sensoren verarbeitet und entsprechende Steuersignale für die Aktoren errechnet. Die Benutzerschnittstelle wird ebenfalls mit dem Informationssystem realisiert. Für diese vielfältigen Aufgabenbereiche ist ein Embedded Controller eine gute Basis. Er liefert, neben einer Mikrocontrollerarchitektur, die wichtigsten Peripheriebausteine, wie Analog-Digital-Wandlung und eine Pulsweitenmodulation zur Ansteuerung von Motoren.

Mikrocontroller

Ein Mikrocontroller stellt für sich ein lauffähiges System dar. Neben einem Mikroprozessor besitzt dieses ein Betriebssystem und die wichtigsten Peripheriebausteine. Der Aufbau eines solchen Mikrocontrollers ist in Abbildung 2.7 dargestellt. Dieses System bildet die Grundlage für einen Embedded Controller. Ein solcher Controller ist für spezielle Aufgaben ausgerüstet, beispielsweise als Grundbaustein für allgemeine mechatronische Systeme. Für ein mechatronisches System werden in der Regel

Aktoren und Sensoren verbaut. Ein darauf ausgelegter Embedded Controller besitzt die entsprechenden Schnittstellen, wie beispielsweise die vorher bereits genannte Analog-Digital-Wandlung oder eine Pulsweitenmodulation.

Echtzeitfähigkeit Neben den Schnittstellen ist die Echtzeitfähigkeit eine zentrale Eigenschaft für die Informationsverarbeitung bei einem mechatronischen System. Bei einer Echtzeitanforderung an das System müssen Berechnungsergebnisse innerhalb einer definierten Zeitspanne zur Verfügung stehen. Wird die definierte Zeitspanne für die Berechnung überschritten, verlieren die Ergebnisse rapide an Wert oder sie sind sofort wertlos. Im Betrieb des Mikrocontrollers muss sichergestellt sein, dass dieser die Berechnungen in Echtzeit durchführt, da ansonsten elektrische und mechanische Komponenten beschädigt werden können. (Roddeck, 2012)

2.5. Testen von Systemen

Das Testen von Systemen ist ein wichtiger Teil des Entwicklungsprozesses. Hier wird sichergestellt, dass ein zuverlässiger und sicherer Betrieb des Systems gewährleistet ist. Durch immer komplexer werdende System und immer engere Zeitvorgaben wird es immer schwieriger ausreichende Tests zur Sicherstellung aller Funktionalitäten durchzuführen. In diesem Kontext kommen neue Methoden zur Sicherstellung von Sicherheit und Zuverlässigkeit bereits im Entwicklungsprozess zum Einsatz.

Leveson (2011) beschreibt einen systematischen Ansatz, mit dem ein solches Vorgehen möglich ist. Drei wesentliche Punkte werden hier aufgegriffen:

- STAMP (Systems-Theoretic Accident Model and Process)
- STPA (Systems-Theoretic Process Analysis)
- Safety-Guided Design

Die Grundelemente von **STAMP** beinhalten die Fokussierung auf Einschränkungen, auf eine Hierarchie in der Kontrollstruktur und auf Prozessmodelle zur Generierung von Kontrollaktionen. Zur Vermeidung von Fehlverhalten des Systems werden Einschränkungen definiert, die verhindern, dass das System in einen gefährlichen

Zustand kommt. Diese Einschränkungen werden von einem Kontrollsystem überwacht, das hierarchisch über dem Prozess steht. Bei mehreren Ebenen kontrolliert die hierarchisch höhere Ebene die direkt darunter liegende.

STPA baut auf STAMP auf und dient zur Identifizierung von Schwachstellen in einem System, die zu einem gefährlichen Verhalten führen können. Zentraler Punkt ist die Identifikation von unangemessenen Steuerungsaktionen des Systems. Sind solche gefunden, wird ihr Auftreten untersucht und so Präventionsmaßnahmen festgelegt. Besonders bei multipel kontrollierten Systemen besteht die Gefahr einer unangemessenen Steuerungsaktion.

Safety-Guided Design verdeutlicht, dass eine Behandlung von Gefahren in einer möglichst frühen Entwicklungsphase Vorteile in Kosten und Sicherheit mit sich bringt. Kann eine Gefahr bereits durch Abwandlung des Konzeptes ausgeschlossen werden, kann auf die Kontrolle dieser Gefahrensituation naturgemäß verzichtet werden. Die Abfolge der Maßnahmen gestaltet sich im Wesentlichen wie folgt:

1. Gefahren eliminieren
2. Gefahren reduzieren
3. Gefahren kontrollieren
4. Schaden minimieren

Je früher Maßnahmen in dieser Auflistung gesetzt werden, desto geringer sind die Kosten und desto besser ist die Effektivität.

3. Systems Engineering

Zunehmende Anforderungen und Erwartungen an Produkte und Lösungen von Kundenseite führen zu einem komplexeren Produkt und dadurch zu einem komplexeren Entwicklungsprozess. Zur Beherrschung dieser, immer komplexer werdenden, Problemstellungen können verschiedene Lösungsansätze angewandt werden, wie beispielsweise Scrum oder die VDI-Richtlinie 2221. In den folgenden Kapiteln wird das Systems Engineering angewandt. Systems Engineering bietet Denkansätze und Vorgehensmodelle, die dem Lösungsprozess zu einem methodischen Aufbau verhelfen und ihn dadurch vereinfachen.

In den nächsten Kapiteln wird genauer auf das Systems Engineering eingegangen und Grundbegriffe daraus eingeführt. Aus dem Systems Engineering lassen sich ein Mikrozyklus zur Problemlösung sowie ein Makrozyklus zum Projektmanagement ableiten. Als grafisches Hilfsmittel wird zusätzlich die Systems Modeling Language (SysML) eingeführt.

3.1. Definition

Zum Verständnis von Systems Engineering müssen bestimmte Begriffe definiert und erklärt werden, die in weiterer Folge, bei der Vorstellung Vorgehensmodelle, verwendet werden. Allem voran steht der Begriff eines Systems und die Definition des Systems Engineering. Aufbauend auf diese Begriffe werden Vorgehensmodelle vorgestellt, die eine systematische Problemlösung ermöglichen.

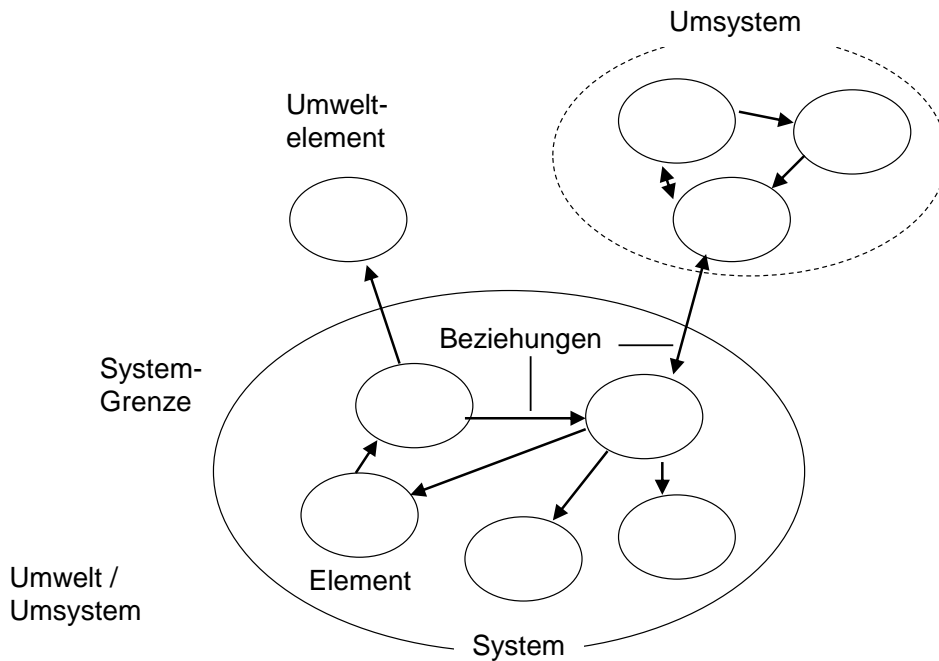


Abbildung 3.1.: Darstellung eines Systems (Haberfellner u. a., 2012, S. 34)

3.1.1. Was ist ein System?

Der Begriff eines Systems wird im Systems Engineering zur Beschreibung von komplexen Zusammenhängen verwendet, die so modellhaft beschrieben werden können, ohne sie zu sehr zu vereinfachen. In Abbildung 3.1 ist die Struktur eines Systems für das Systems Engineering grafisch dargestellt. Ein System besteht aus **Elementen**, die in einer **Beziehung** zueinander stehen. Die einzelnen Elemente können in sich wieder als System betrachtet werden und besitzen Eigenschaften und Funktionen. Die Beziehungen unter diesem Element sind als sehr allgemein zu betrachten und können zum Beispiel den Material-, Informations- oder Energiefluss beschreiben. Wesentlich für die Beschreibung eines Systems ist im Weiteren eine **Systemgrenze**, die das betrachtete System von der Umgebung abgrenzt. (Haberfellner u. a., 2012)

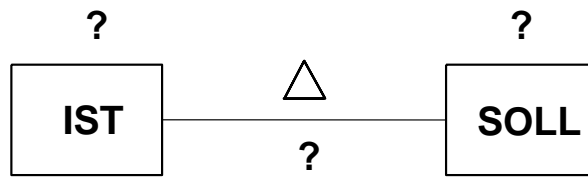


Abbildung 3.2.: Problem als Differenz zwischen IST und SOLL (Haberfellner u. a., 2012, S. 27)

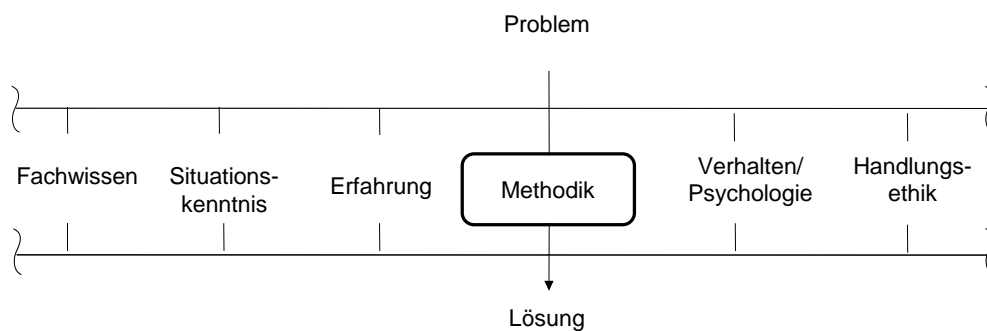


Abbildung 3.3.: Methodik im Kontext der Problemlösung (Haberfellner u. a., 2012, S. 28)

3.1.2. Was ist Systems Engineering?

Systems Engineering definiert Vorgehensmodelle und Denkansätze die als Hilfsmittel zur Problemlösung fungieren. Ein Problem ist nach Haberfellner u. a. (2012) die Differenz zwischen einer Vorstellung und dem Vorhandenem, zwischen einem SOLL und dem IST und ist in Abbildung 3.2 zur Verdeutlichung dargestellt. Einer akkuraten Beschreibung des Problems geht also die Feststellung des IST sowie die Festlegung des SOLL voran. Systems Engineering liefert Methoden, die eine Problemlösung anstreben und unterstützen. Methodik allein führt einem Problem aber noch keine Lösung zu. Dass andere Faktoren ebenfalls eine Rolle spielen, sei in Abbildung 3.3 verdeutlicht. Beispielsweise wird eine Problemlösung durch fehlendes Fachwissen und Erfahrung in den fraglichen Bereichen erheblich erschwert oder sogar nahezu unmöglich gemacht.

Einen guten Überblick über die Inhalte des Systems Engineering liefert die Abbil-

dung 3.4. Die Anordnung der Themen ist darin bewusst gewählt. Die SE-Philosophie bildet die grundsätzlichen Denkansätze die zu einer Lösung führen. Diese Ansätze werden im Problemlösungsprozess eingesetzt und führen so zu einem methodischen Vorgehen auf dem Weg vom Problem zur Lösung. Techniken der Systemgestaltung und des Projektmanagements unterstützen dabei den Problemlösungsprozess. In der SE-Philosophie wird zwischen dem Systemdenken und dem Vorgehensmodell unterschieden. Das Systemdenken stellt Zusammenhänge aus der realen Welt möglichst einfach und verständlich dar. Ein kurzer Auszug wurde bereits in Kapitel 3.1.1 behandelt. Das Vorgehensmodell beschreibt den systematischen Ablauf einer Problemlösung. Dieser Ablauf findet sich in den Projektphasen, die im nächsten Absatz beschrieben werden, wieder. Das Vorgehensmodell beinhaltet neben den Projektphasen den Problemlösungszyklus das Prinzip des „Denkens in Varianten“ und das Vorgehen „Vom Groben zum Detail“. Im Problemlösungsprozess werden die vorgestellten Vorgehensmodelle und das Systemdenken im Zuge der Systemgestaltung, konkret in der Konzeptgestaltung, an einem realen Problem zur Anwendung gebracht. Bei der Architekturgestaltung handelt es sich um die Gestaltung des jeweiligen Lösungsprinzips. Das Projektmanagement übernimmt die Planung, Überwachung und Steuerung des Problemlösungsprozesses. Mit den Techniken zur Systemgestaltung sind die einzelnen Methoden gemeint, die im Laufe eines Problemlösungsprozesses Anwendung finden können, wie beispielsweise die stufenweise Variantenbildung und die Nutzwertanalyse. Analoges gilt für die Techniken des Projektmanagements.

Die Gliederung eines Projekts erfolgt beim Systems Engineering nach einer Reihe von Projektphasen, die in Abbildung 3.5 dargestellt sind. Die Phasen sind nach dem Prinzip „Vom Groben ins Detail“ aufgebaut. In der Vorstudie werden die Rahmenbedingungen für ein Projekt abgesteckt und in den weiteren Phasen Schritt für Schritt konkretisiert, bis es schließlich zur Systemeinführung kommt. In den jeweiligen Projektphasen kommt der Problemlösungszyklus, der ebenfalls in der Abbildung 3.5 dargestellt ist, zum Einsatz. Auf den Problemlösungszyklus wird gesondert in Kapitel 3.1.4 eingegangen. Die Projektphasen finden sich im Aufbau dieser Arbeit wieder. Ein gängiger Makrozyklus bei der Entwicklung eines mechatronischen Systems ist das V-Modell, das im nächsten Kapitel beschrieben wird.

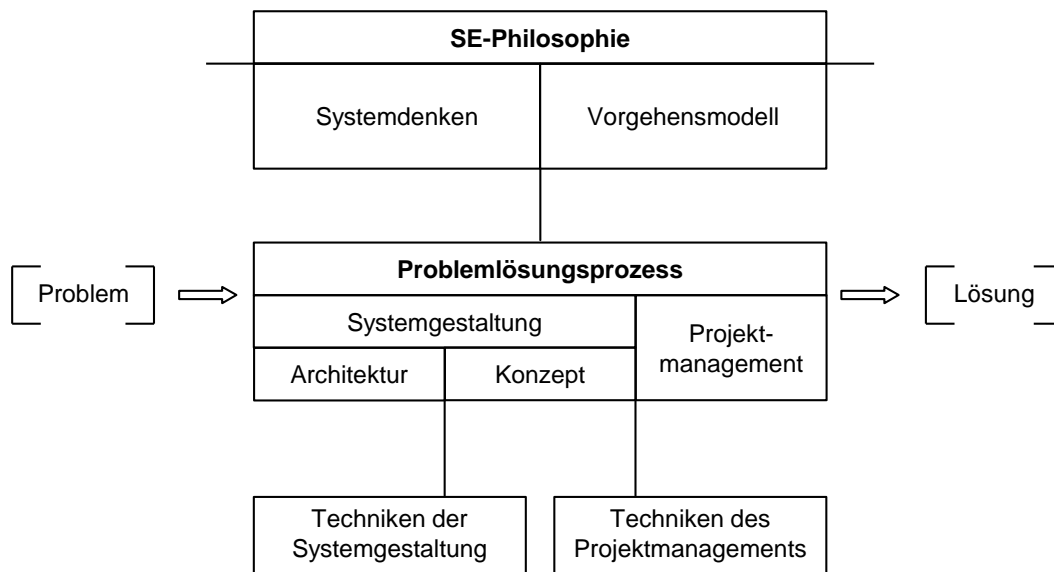


Abbildung 3.4.: Konzept des Systems Engineering (Haberfellner u. a., 2012, S. 28)

3.1.3. V-Modell

Als Makrozyklus kommt das in der VDI2206 (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004) und von Haberfellner u. a. (2012) beschriebene V-Modell zur Anwendung. Der Name stammt von der grafischen Darstellung des Modells entsprechend der Abbildung 3.6. Im Systems Engineering entspricht das V-Modell einem Vorgehensmodell, wie es in der SE-Philosophie zu finden ist. Im V-Modell werden der Top-Down- und der Bottom-Up-Ansatz miteinander kombiniert.

Beim Vorgehen nach dem V-Modell werden als Ausgangspunkt die **Anforderungen** an das Gesamtsystem, die das endgültige Produkt erfüllen muss. Mit diesen **Anforderungen** kann ein Systementwurf angefertigt werden. Der Entwurf stellt eine domänenübergreifende Lösung des Problems dar. Daraus werden die Teillösungsprinzipien abgeleitet, die für den domänenspezifischen Entwurf notwendig sind. Im **domänenspezifischen Entwurf** werden die einzelnen Teillösungen in der jeweiligen Domäne konkretisiert, ausgelegt und berechnet. Die Konkretisierung erfolgt dabei meist getrennt für eine Domäne. Die damit erhaltenen Ergebnisse werden schließlich im Zuge der **Systemintegration** zu einem **Produkt** zusammengeführt. Eine ständige **Eigenschaftsabsicherung** stellt sicher, dass die zu Beginn getroffenen Anforderungen

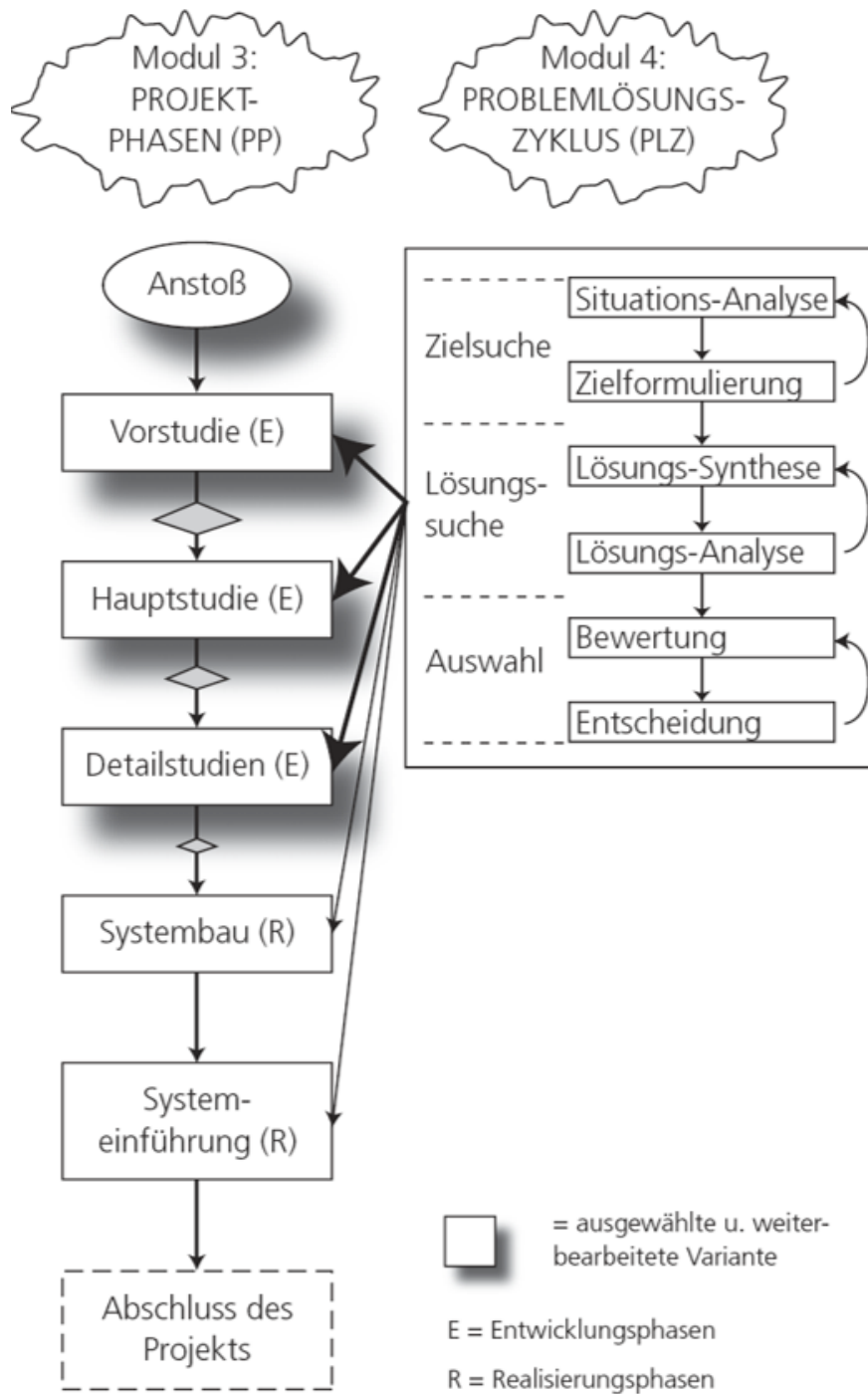


Abbildung 3.5.: Projektphasen (Haberfellner u. a., 2012, S. 28)

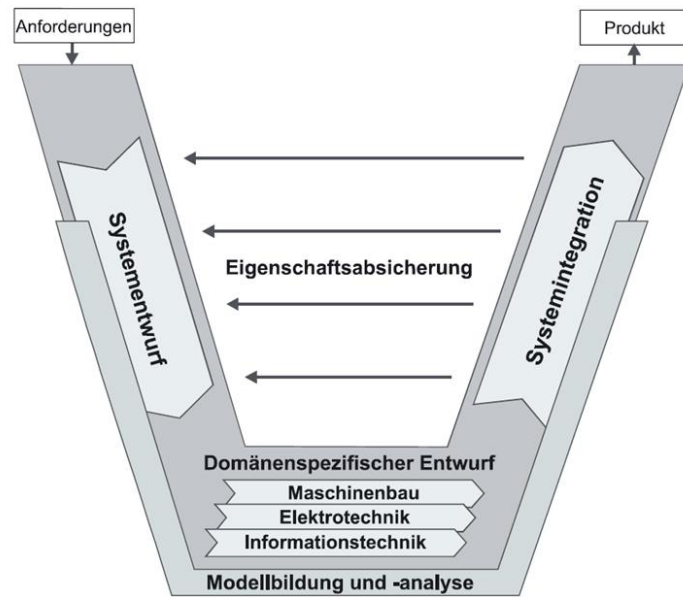


Abbildung 3.6.: V-Modell (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004, S. 29)

an das Produkt erfüllt werden. Bei der Eigenschaftensicherung ist also eine fortlaufende Kontrolle des Lösungskonzepts mit dem Systementwurf und insbesondere den Anforderungen an das System notwendig. Die Entwicklung wird durch geeignete Modellbildung und -analyse unterstützt. In der Regel wird das V-Modell mehrere Male durchlaufen, um so einen immer höheren Reifegrad zu erreichen. Beispielsweise entsteht beim ersten Durchlauf ein Labormuster, das nach mehreren Durchläufen zur Serienreife gebracht wird. Beim Durchlaufen des V-Modells müssen immer wieder Probleme gelöst werden, um im Projekt voranschreiten zu können. Eine Möglichkeit den auftretenden Problemen zu begegnen ist der Problemlösungszyklus, der im nächsten Kapitel beschrieben wird.

3.1.4. Vorgehensmodell als Mikrozyklus

Der **Problemlösungszyklus**, der von Haberfellner u. a. (2012) beschrieben wird und auf den auch die VDI2206 (VDI-Fachbereich Produktentwicklung und Mechatronik, 2004) verweist, bildet den Mikrozyklus zur Lösung von Problemen in jeder Projektphase. Der Problemlösungszyklus ist in der SE-Philosophie als Vorgehensmodell enthalten. Die Schwerpunkte des Zyklus bestehen in

- Zielsuche,
- Lösungssuche und
- Auswahl.

Man beginnt mit der Konkretisierung des Ziels, sucht geeignete Lösungen und wählt die zweckmäßigste Lösung für das vorliegende Problem aus. Diese Vorgehensweise ist von Haberfellner u. a. (2012) in Abbildung 3.7 verdeutlicht. Die einzelnen Punkte des Problemlösungszyklus werden im Folgenden aufgeschlüsselt und erläutert.

Anstoß Der Anstoß bildet den Einstieg in den Problemlösungszyklus. In dieser Phase wird über den Start des Zyklus entschieden. Fällt diese Entscheidung positiv aus, werden die nächsten Schritte in Gang gesetzt. Einen Anstoß kann eine Entscheidung in einer höheren Planungsebene darstellen, wie das Festlegen auf ein Lösungsprinzip, das in der nächsten tieferen Detailstufe umgesetzt werden muss.

Situationsanalyse Zu Beginn ist es sinnvoll sich ein klares Bild über die aktuelle Lage zu verschaffen und das Problem im richtigen Kontext zum Gesamtprojekt und anderen Einflüssen zu sehen. Die Situationsanalyse liefert die quantitativen und qualitativen Informationen für die Zielformulierung und legt Randbedingungen für etwaige Ziele fest. Mit dem geschärften Blick auf das Problem können im nächsten Schritt die Ziele für die Problemlösung definiert werden.

Zielformulierung Die in der Situationsanalyse gewonnenen Informationen werden in der Zielformulierung in konkrete Ziele umgesetzt. Je nach Projektphase und zu lösendem Problem können diese mehr oder weniger präzise sein. Mit den Zielen wird die erwünschte Wirkung einer Lösung beschrieben. Nicht immer können alle Ziele vollständig erreicht werden. Eine Priorisierung der Ziele erleichtert die Unterscheidung, welche erreicht werden müssen und welche Ziele Wunschziele sind. Die Prioritäten sind in absteigender Reihenfolge:

- Mussziele
- Sollziele
- Wunschziele

Synthese von Lösungen Anhand der Ziele könne bei der Synthese von Lösungen verschiedene Lösungskonzepte erarbeitet werden. Die Ausarbeitung der Lösungen sollte in diesem Schritt soweit gehen, dass eine Gegenüberstellung der

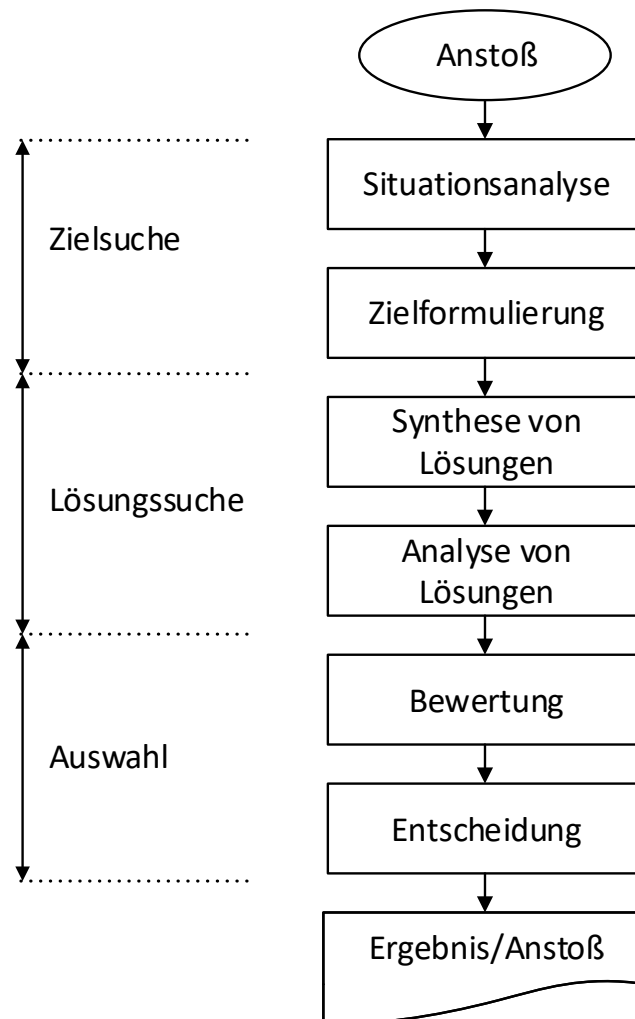


Abbildung 3.7.: Problemlösungszyklus (Haberfellner u. a., 2012, S. 74)

Varianten möglich ist.

Analyse von Lösungen Bei der Analyse von Lösungen gilt es, die erarbeiteten Lösungsmöglichkeiten zu prüfen und ihre Tauglichkeit sicher zu stellen. Neben einer Vorselektierung von untauglichen Lösungen sollen taugliche Lösungsentwürfe weiter konkretisiert werden.

Bewertung von Lösungen Bei der Lösungsbewertung werden nur Lösungen berücksichtigt, die alle Mussziele erfüllen. Wenn keine Lösung die offensichtlich bessere ist, können verschiedene Methoden angewendet werden, die eine Entscheidungsbasis bilden. Mögliche Methoden und Techniken, die zur Anwendung kommen können sind:

- Argumenten-Bilanz
- Nutzwertanalyse
- Kosten-Nutzen-Rechnung
- Kosten/Wirksamkeits-Analyse
- Wirtschaftlichkeitsrechnung
- Real-Optionen

Entscheidung Auf Basis der Bewertungsergebnisse wird eine umzusetzende Lösungsvariante festgelegt

Ergebnis Am Ende des Problemlösungszyklus kann eine Lösung für das Problem stehen oder keine Lösung gefunden worden sein. Eine gefundene Lösung kann der Anstoß für weitere Zyklen oder andere Projektphasen sein. Ist keine Lösung gefunden worden, müssen die Konsequenzen für das Gesamtprojekt gezogen werden (Projektabbruch, anderes Lösungsprinzip, ...).

3.2. Systems Modeling Language

Mit der Systems Modeling Language (SysML) können Strukturen und Verhalten von komplexen Systemen klar dargestellt werden. Sie bietet die verschiedenen Diagrammtypen die ein Problem aus verschiedenen Sichten darstellen. Das Problem kann so gut grafisch aufbereitet und komplexe Sachverhalte einfacher dargestellt werden. Die Systems Modeling Language ist eine Erweiterung der Unified Modeling Language (UML), Version 2. SysML ist eine standardisierte Modellierungssprache,

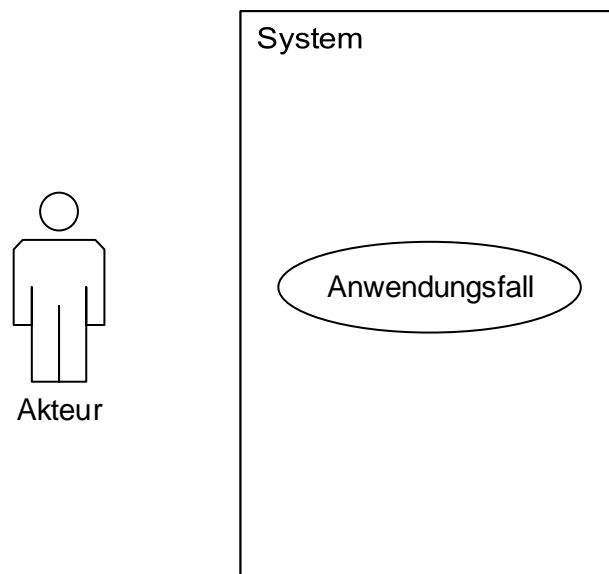


Abbildung 3.8.: Grundelemente Use Case Diagramm (Friedenthal, Moore und Steiner, 2012)

die im Systems Engineering verwendet wird. Die wichtigsten Eigenschaften einer solchen Modellierungssprache sind einerseits, dass sie alle Abbildungsanforderungen erfüllen kann und dabei eine robuste Sprache darstellt. Beide Eigenschaften sind in SysML enthalten. (Friedenthal, Moore und Steiner, 2012)

3.2.1. Use Case Diagramm

Das Use Case Diagramm, im deutschen als Anwendungsfalldiagramm bezeichnet, bildet Nutzer und deren Beziehung zu einem System in verschiedenen Anwendungsfällen ab. Durch diese Darstellung gelingt es, die Anwendungsfälle für ein System im Auge zu behalten und bei der Entwicklung in Form von Anforderungen an das zu entwickelnde System entsprechend zu berücksichtigen.

Die Grundelemente eines Use Case Diagramms sind, wie auch in Abbildung 3.8 dargestellt: die **Akteure**, das **System** und die **Anwendungsfälle**. Diese Elemente können in unterschiedlichen Verbindungen zueinander stehen. Die Verbindungen sind in Abbildung 3.9 abgebildet und werden nachfolgend beschrieben:

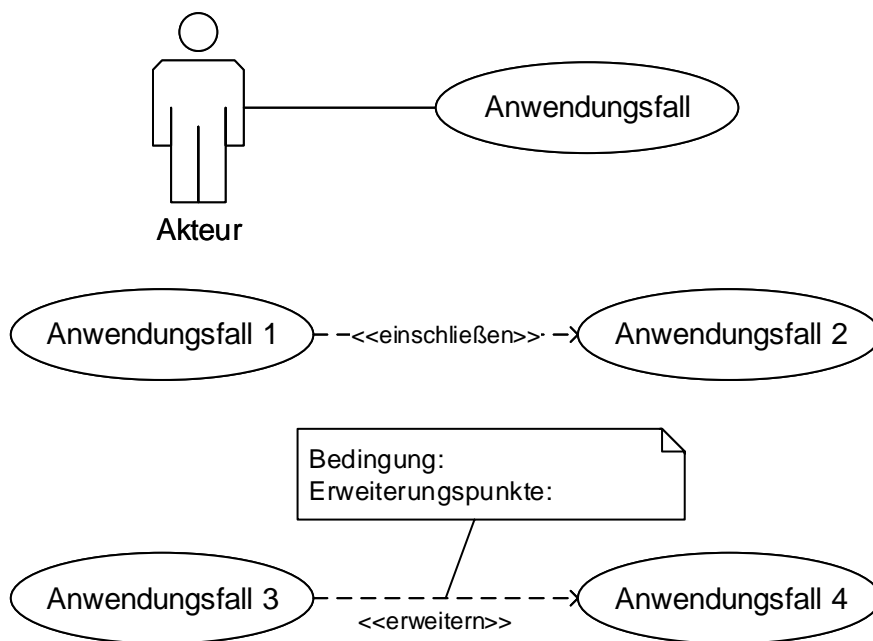


Abbildung 3.9.: Verbindungen Use Case Diagramm (Friedenthal, Moore und Steiner, 2012)

Assoziation dargestellt mittels einer durchgezogenen Linie zwischen einem Akteur und einem Anwendungsfall. Diese Linie markiert, dass der Anwendungsfall für den verbundenen Akteur von Relevanz ist, er ihn anwendet und damit Informationen austauscht.

Einschließen Assoziation wird symbolisiert durch eine gestrichelte Linie mit dem Wort „einschließen“ in spitzen Klammern und verbindet zwei Anwendungsfälle. Bei einem eingeschlossenen Anwendungsfall (wie der Anwendungsfall 2 in Abbildung 3.9) wird dessen Funktionalität in den einschließenden Anwendungsfall integriert. Das heißt, dass bei Ausführung des Basisanwendungsfalls der eingeschlossene Anwendungsfall im Laufe der Ausführung ebenfalls ausgeführt wird.

Erweitern Assoziation ist dargestellt durch eine gestrichelte Linie mit dem Wort „erweitern“ in spitzen Klammern und verbindet zwei Anwendungsfälle. Der Anwendungsfall, von dem die Verbindung ausgeht (in Abbildung 3.9 der Anwendungsfall 3), wird nur ausgeführt, wenn im eingehenden Anwendungsfall (in Abbildung 3.9 Anwendungsfall 4) bestimmte Bedingungen erfüllt wurden.

In einer Notiz zur Verbindung ist die Bedingung angegeben, sowie der Punkt, an dem die Erweiterung auf den zusätzlichen Anwendungsfall stattfindet. Dieser Punkt wird ebenfalls beim Anwendungsfall (in der Abbildung 3.9 der Anwendungsfall 4) angegeben.

Mit den beschriebenen Elementen lassen sich die wichtigsten Zusammenhänge für die Verwendung von Systemen einfach darstellen. Das Diagramm und die enthaltenen Elemente sind dabei bewusst einfach gehalten, um den Fokus auf die wesentlichen Informationen zu lenken. (Friedenthal, Moore und Steiner, 2012)

3.2.2. State Machine Diagramm

Das State Machine Diagramm, zu Deutsch Zustandsdiagramm, eignet sich gut, um die verschiedenen Zustände eines Objekts während der Laufzeit darzustellen. Die Darstellung dient als Basis für die Programmierung und verschafft dem Programmierer damit einen Überblick über die Zustände, die erreicht werden können und wie diese erreicht werden.

Einige ausgewählte Elemente für die Erstellung von Zustandsdiagrammen sind in Abbildung 3.10 dargestellt. Diese Elemente werden für die weiteren Diagramme verwendet und sind für diese ausreichend. Es sei darauf hingewiesen, dass diese Diagrammelemente nicht vollständig aufgelistet sind und die weiteren Elemente in der Fachliteratur nachgelesen werden können. Beispielsweise beschreiben Friedenthal, Moore und Steiner (2012) diese sehr detailliert.

Bei allen in Abbildung 3.10 dargestellten Symbolen handelt es sich um Zustände. Allerdings sind nur das erste und das dritte Symbol ein tatsächlicher Zustand, in dem sich das System dauerhaft befinden kann. Die weiteren Zustände werden als sogenannte Pseudozustände bezeichnet. Diese Zustände kann das System nur übergangsweise und für kurze Zeit einnehmen. Ein anschauliches Beispiel dafür bietet der Entscheidungsknoten, das 4. Symbol in der Liste. Dabei handelt es sich in der Regel um eine Bedingung, die entweder erfüllt sein kann oder nicht. In beiden Fällen geht das System in den nächsten Zustand über.

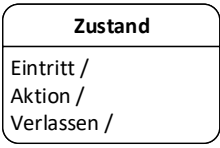

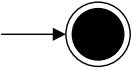

	Zustand
	Startzustand
	Endzustand
	Entscheidung

Abbildung 3.10.: Ausgewählte Elemente State Machine Diagramm (Friedenthal, Moore und Steiner, 2012)

Die Verbindungen zwischen den Zuständen werden als Pfeile dargestellt, wie sie in der Abbildung 3.10 bei Start- und Endzustand angedeutet sind. Übergänge zwischen den Zuständen erfolgen, wenn ein bestimmtes Ereignis eintritt und die Bedingungen für den Übergang erfüllt sind. Solche Ereignisse und Bedingungen werden in der Regel mit dem Übergangspfeil angegeben. Die Bedingung wird nicht angegeben, wenn für den Übergang keine Einschränkung notwendig ist. Das Ereignis wird nicht angegeben, wenn es dem Ende des vorherigen Zustandes entspricht.

4. Mechatronische Nivellierungssysteme

Nivellierungssysteme sind in verschiedenen Anwendungen und Ausführungen anzutreffen. Beispielsweise gewinnen kompakte Einpersonen-Transportsysteme zusehends an Popularität. Konkret sind das die als Segway® und die, umgangssprachlich, als „Hoverboard“ bezeichneten Vehikel. Sowohl Segways als auch Hoverboards besitzen nur eine Achse mit zwei Rädern. Konstruktiv ist also ein Kippen um diese Achse möglich, wird aber durch ein entsprechendes mechatronisches System verhindert. Im Folgenden werden zwei Systeme näher beleuchtet.

4.1. Selbstblancierende Vehikel

Bei einem selbstbalancierendem Vehikel wird im Wesentlichen ein inverses Pendel in der instabilen Gleichgewichtslage gehalten. Eine beispielhafte Auslegung eines solchen Systems wird anhand des Beispiels Segway von Roddeck (2012) durchgeführt. Das Segway, sowie das zugehörige mechanische Modell sind in Abbildung 4.1 und 4.2 dargestellt.

Die Anfertigung eines mechanischen Modells ist essentiell für den Reglerentwurf und die Auslegung des mechatronischen Systems. Besonders in Fällen, wie diesem Beispiel, in dem Versuche mit einem Prototyp zur Reglerauslegung sich als schwierig bis nicht durchführbar gestalten, ist die Modellbildung von entscheidender Bedeutung. In unserem Beispiel ist die mechanische Modellbildung in der Abbildung 4.2 bereits erfolgt, da die wesentlichen Annahmen mit dieser Skizze getroffen wurden. Somit gilt es die mathematischen Gleichungen aufzustellen, mit denen das System beschrieben



Abbildung 4.1.: Segway (Roddeck, 2012, S. 440)

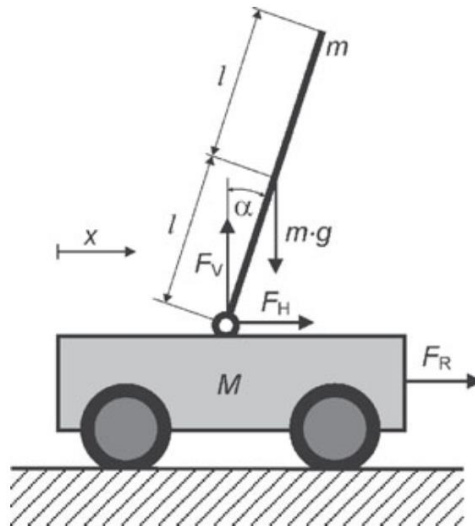


Abbildung 4.2.: Modell Segway als inverses Pendel (Roddeck, 2012, S. 440)

wird. Roddeck (2012) verwendet zu diesem Zweck die Newton-Euler-Methode, wie sie in Kapitel 2.4.1 beschrieben wurde. Damit folgen die Bewegungsgleichungen für das Pendel:

$$m \cdot l \cdot \ddot{\alpha} + (m + M)\dot{x} = F_R \quad (4.1)$$

$$(J + M \cdot l^2)\ddot{\alpha} + m \cdot l \cdot \dot{x} - m \cdot g \cdot l \cdot \alpha = 0 \quad (4.2)$$

Wobei J das Massenträgheitsmoment des Pendels darstellt. Mit diesen Gleichungen kann ein Regler entworfen werden, der in der Lage ist, das Segway in aufrechter Position zu halten. Davon ausgehend, wird die erforderliche Aktorik und Sensorik sowie ein passender Mikrochip gewählt. Diese Einzelkomponenten können in weiterer Folge zu einem Gesamtsystem integriert werden.

4.2. Hochpräzise selbstnivellierende Plattform

Ein ähnliches Problem, wie das Nivellieren eines Wohnmobils, wird von Geranmehr, Vafae und Sadeqi (2014) beschrieben. Seine Arbeit beschäftigt sich mit einer hochpräzisen, selbstnivellierenden Plattform. Diese Plattform ist mit vier elektromechanischen Stützen ausgestattet, die eine Nivellierung der Plattform durchführen. Ein Freischnitt mit einer Prinzipskizze ist in Abbildung 4.3 dargestellt.

Die Lageregelung erfolgt bei dieser Plattform über eine Fuzzylogik. Eine Fuzzylogik aus Bedingungen aufgebaut, wie beispielsweise, dass sich gegenüber des höchsten Zylinders der Plattform, der niedrigste Zylinder befinden muss. Daraus können Informationen für die Lageregelung abgeleitet werden, wie zum Beispiel welcher Zylinder am schnellsten ausgefahren werden muss. Der Grundaufbau der Regelung besteht aus zwei Schleifen. Die Bewegung der Aktoren wird synchron gesteuert und so die Plattform in eine waagrechte Lage gebracht. Das Ziel ist im Wesentlichen ident mit dem der Wohnwagennivellierung. Jedoch bestehen vollkommen andere Anforderungen an diese hochpräzise selbstnivellierende Plattform. Vor allem bei der Geschwindigkeit und Präzision mit der die Nivellierung stattfindet, bewegt man sich in einem vollkommen anderen Bereich, als bei der Nivellierung eines Wohnwagens.

4. Mechatronische Nivellierungssysteme

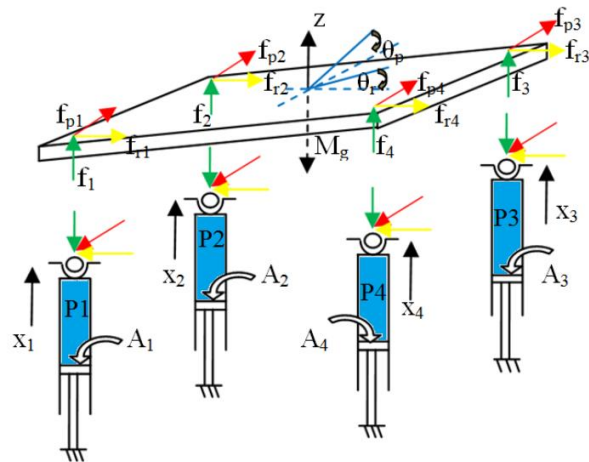


Abbildung 4.3.: Freischnitt und Prinzipskizze einer selbst nivellierenden Plattform (Geranmehr, Vafae und Sadeqi, 2014)

Bei der Plattform wird die Endposition in einer Zeit von circa 5 Sekunden erreicht, während beim Wohnwagen eine Nivellierung in circa zwei Minuten abgeschlossen ist. Die Genauigkeitsanforderungen liegen beim Wohnwagen im Bereich von einem halben Grad, während bei dieser Plattform eine Genauigkeit eine Abweichung von maximal zwei Winkelminuten erreicht wird. All das bei einer Masse der Plattform von 120 Tonnen.

5. Vorstudie

Am Beginn eines Projektes steht, entsprechend den Projektphasen aus dem Systems Engineering, die Vorstudie. Hier gilt es den Projektumfang zu definieren, das Problem zu identifizieren und Lösungsanforderungen mit den Stakeholdern zu vereinbaren. Eine klare Definition der Erwartungen an die Lösung räumt im Vorfeld bereits Missverständnisse bei Projektergebnissen aus und ermöglicht so ein gezieltes Arbeiten an der gewünschten Lösung. In der Vorstudie können bereits grobe Lösungsprinzipien betrachtet und eine Vorauswahl für die erfolgversprechendsten Varianten getroffen werden.

5.1. Projektbeschreibung

Als Projektinitiator tritt die Firma Smart Lift auf, die ihr erstes Produkt, den Caravan Smart Lift, entwickelt. In Kooperation mit der TU Graz sollen in verschiedenen Teilgebieten Fortschritte erzielt werden. Das betrifft einen Algorithmus zur Nivellierung von Wohnwägen und Wohnmobilen sowie eine Applikation für Smartphones, die den Nivelliervorgang steuert. In den weiteren Kapiteln steht der Algorithmus und dessen Entwicklung im Vordergrund.

Die Firma Smart Lift hat bereits in eigenständiger Vorarbeit einen Modellaufbau angefertigt, der die potenziell zu verbauenden Stützen beinhaltet und über einen Mikrocontroller gesteuert wird, der zu Testzwecken über einen einfachen Algorithmus verfügt. Der Prototyp kann über eine Bluetooth-Schnittstelle gesteuert werden. Für die weiteren Schritte im Projekt sollen die Hardwarekomponenten beibehalten werden. Maßgabe ist es einen effizienten und zuverlässigen Algorithmus zu entwickeln, der die Steuerung der Stützen übernehmen soll. Des Weiteren sollen in Zukunft die

aktuellen Stützen durch eine Eigenentwicklung der Firma Smart Lift ersetzt werden. Das bedeutet, der Algorithmus muss unabhängig von der konkreten Stütze sein. Die Eigenentwicklung einer Stütze ist jedoch nicht Umfang der Arbeit.

Für den Algorithmus müssen verschiedene Randbedingungen und Anforderungen eingehalten werden, die im Abschnitt 5.2 genauer erläutert werden. Im Wesentlichen muss eine sichere Nivellierung gegeben sein, die die Integrität des Wohnwagens oder des Wohnmobils nicht zerstört. Zusätzlich sind von Firmenseite einige Funktionen vorgegeben, die von der Steuerung beherrscht werden sollen. Diese treten später in der Smartphoneapplikation als Steuerbefehle auf oder dienen der Sicherheit des Algorithmus.

Der Caravan Smart Lift soll in zwei Ausführungen entwickelt werden, die sich in der Stützenanzahl unterscheiden. Für Wohnwägen wird primär eine Variante mit sechs Stützen angestrebt, da die Hauptlast über die Achse gehoben werden muss und die äußeren Stützen für Stabilität sorgen. Bei einem Wohnmobil sind vier Stützen ausreichend, da dort die Last an den äußeren Punkten gehoben werden kann.

Für eine Einschätzung der tatsächlich vorliegenden Kräfte und für weitere Auslegungen ist es notwendig ein mechanisches Modell anzufertigen. Dieses Modell dient in weiterer Folge als Informationsquelle für den Algorithmus und dessen Entwicklung.

5.2. Anforderungen

Die Anforderungen an die Lösung müssen klar definiert werden. Gemeinsam mit der Firma Smart Lift wurden, bei Projektstart, die Anforderungen ausgearbeitet und schriftlich festgehalten. Zusammenhängende der Anforderungen werden zur besseren Übersicht in Kategorien eingeteilt. Je nach Priorität der Anforderungen werden diese in Muss-, Soll- und Wunschanforderungen unterteilt. Eine Mussanforderung ist in jedem Fall zu erfüllen, sollte diese nicht erfüllt werden, ist mit schweren Mängeln in Funktionalität und Sicherheit des Endproduktes zu rechnen. Eine Sollanforderung stellt einen erstrebenswerten Mehrwert, bei Erreichung, für das Endprodukt dar. Es sollten also möglichst viele Sollanforderungen erreicht werden. Die Wunschanforderungen stellen im Wesentlichen eine abgeschwächte Variante der Sollanforderungen

dar. Sie bieten ebenfalls einen Mehrwert. Eine Sollenforderung sollte aber, gegenüber einer Wunschanforderung, priorisiert werden.

Generell gibt es drei Anwendungsfälle, für die der Algorithmus geeignet sein soll:

- Wohnwagen
- Wohnmobile (<5,2t)
- Nachrüstung für Wohnwägen

Als Anwender werden folgende Zielgruppen definiert:

Zielgruppe 1: 60-jähriger, mittelständischer Camper, der nicht sehr technologieaffin ist und eine einfache und zuverlässige Handhabung wünscht. (Primäre Zielgruppe)

Zielgruppe 2: Zukünftige Camper, die heute 35 Jahre alt sind und eine Technologieaffinität besitzen. Neben einer zuverlässigen Bedienung ist diese Gruppe an Zusatzinformationen über den Nivelliervorgang interessiert. Die Einstellung von einfachen Parametern kann an sie übertragen werden.

Eine Verdeutlichung der Anwendungen, denen das System gerecht werden muss, ist in Abbildung 5.1 als Anwendungsfalldiagramm dargestellt. Die beiden angeführten Zielgruppen sind eingezeichnet, sowie ein Servicetechniker, der das System wartet und mögliche Fehler behebt. Die Zielgruppe 2 nutzt einen erweiterten Umfang im Vergleich zur ersten Zielgruppe. Insbesondere können Parameter in einem erweiterten Modus angepasst werden. Die Grundfunktionalitäten sind für die Zielgruppe 2 ebenso relevant.

Für die Formulierung der Anforderungen werden diese in drei Teilbereiche unterteilt:

- Funktionale Anforderungen
- Strukturanforderungen
- Anforderungen an das mechanische Modell

Die Relevanz der Anforderungen wird in Muss-, Soll- und Wunschbedingungen eingeteilt. In der folgenden Auflistung sind alle für die Steuerung relevanten Anforderungen zusammengefasst.

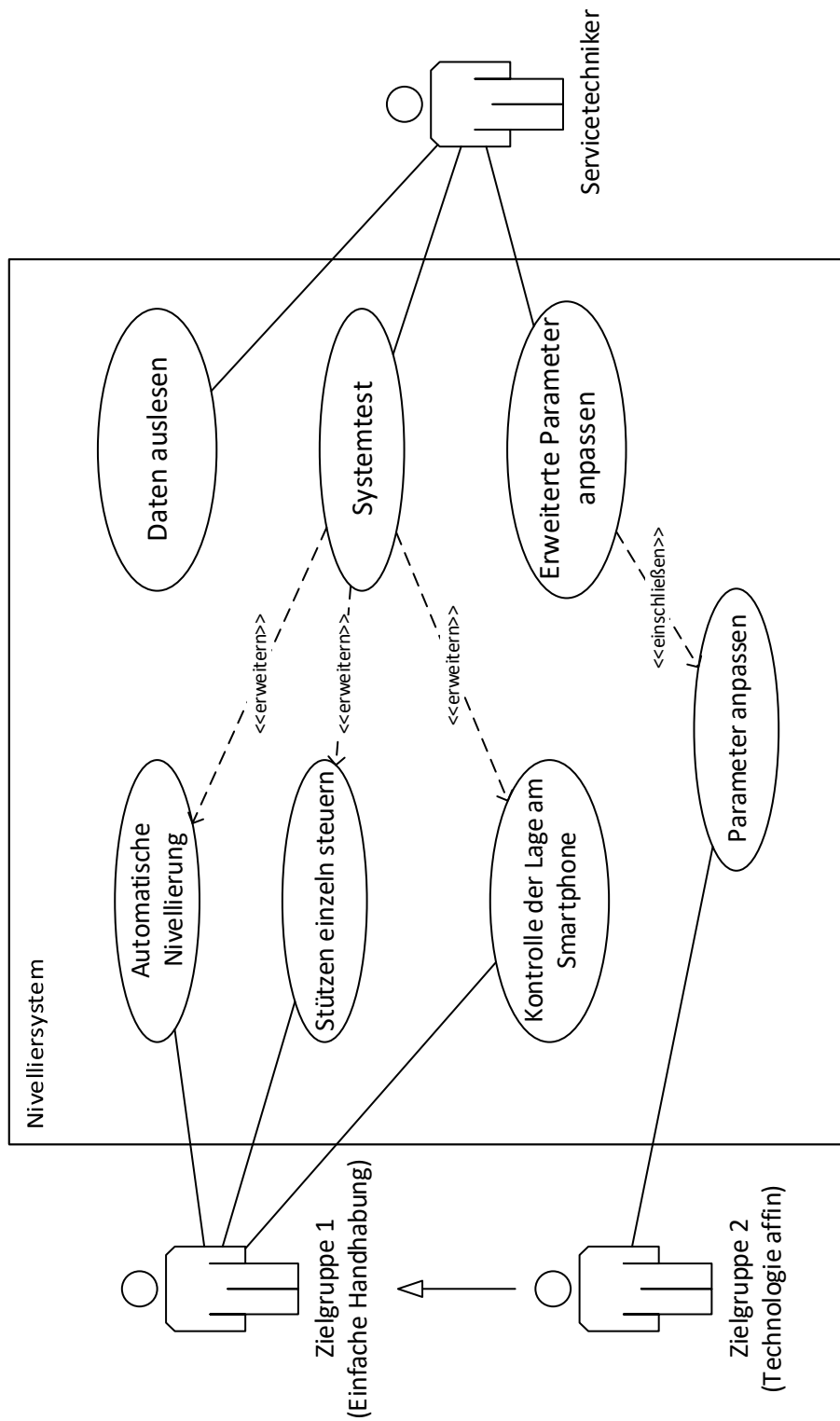


Abbildung 5.1.: Use Case Diagram: Automatische Nivellierung

1. Funktionale Anforderungen

a) Muss

Neigungswinkel $\pm 0,5^\circ$ Im nivellierten Zustand soll die Abweichung maximal $\pm 0,5^\circ$ betragen.

Stützenanzahl im Algorithmus variabel Die Anzahl der Stützen muss im Algorithmus der Vier- und Sechsstützenvariante angepasst werden können.

Stützenposition parametrisierbar Die Position der einzelnen Stützen am Chassis soll in der Steuerung an einen Wohnwagen oder ein Wohnmobil angepasst werden können.

Nivellierung bis 15% Neigung Die Nivellierung muss bis zu einem Neigungswinkel des Untergrunds von 15% problemlos möglich sein.

Automatisches Nivellieren Kein Nutzereingriff während eines Nivelliervorgangs nötig.

Stützen automatisch einfahren Stützen müssen vollständig und automatisch einfahren.

Stützen manuell einfahrbar Bei einem Versagen des mechatronischen Systems müssen die Stützen manuell eingefahren werden können.

Stützen einzeln verfahrbar Die Stützen müssen einzeln gesteuert werden können.

Verschiedene Untergründe Die Nivellierung muss auf verschiedenen Untergründen (Asphalt, Sand, Schotter, Gras, ...) zum erwünschten Ergebnis führen.

b) Soll

Einmalige Kalibrierung Für das System soll eine einmalige Kalibrierung vor der Inbetriebnahme ausreichend sein.

Nivellierten Zustand erhalten Bei einem veränderten Zustand, außerhalb der Toleranzen für die Nivellierung, soll im Betrieb mit kontinuierlicher Stromversorgung automatisch nachnivelliert werden. Bei Batteriebetrieb ist eine Warnung an den Nutzer auszugeben.

Nivellierverhalten anpassbar Verschiedene Verhaltensmuster bei der Nivellierung sollen an Kundenwünsche angepasst werden können.

Informationen zum Systemzustand Im Betrieb sollen Informationen zum

aktuellen Systemzustand dem Nutzer zugänglich gemacht werden.

Erstellung von Kraftverläufen Bei einem Nivelliervorgang soll für jede Stütze ein Kraftverlauf erstellt werden können.

Schnittstelle zum Bussystem Die Steuerung soll in das existierende Bussystem des Wohnwagens oder Wohnmobils eingebunden werden können.

Überwachung der Batterieladung Bei geringer Batterieladung soll der Nutzer informiert werden.

2. Anforderungen an die Systemstruktur

a) Muss

Integrität des Chassis Das Chassis darf nicht plastisch verformt werden.

Seitenkräfte aufnehmbar Mögliche Seitenkräfte die aus Windkräften oder ähnlichem resultieren müssen aufgenommen werden können.

Gekapselte Systeme Das System soll möglichst modular aufgebaut sein.

Systemsicherheit Das Nivellierungssystem muss gegen unerwünschte Eingriffe von außen geschützt werden.

Witterungsbeständigkeit Das System muss bei Temperaturen von $-40\text{ }^{\circ}\text{C}$ bis $70\text{ }^{\circ}\text{C}$ einsatzbereit sein.

Maximale Ausfahrtslängen Zur Vermeidung zu weitem Ausfahrens der Stützen muss die maximale Ausfahrtslänge beschränkt werden.

b) Soll

Minimale Sensorik Die Anzahl der Sensoren soll so gering als möglich gehalten werden.

Minimale Bauteilanzahl Das System soll aus möglichst wenigen Bauteilen bestehen. Dadurch wird ein Einbauen und das Nachrüsten erleichtert.

Leichtbauweise Da das System in Fahrzeugen verbaut werden soll, ist eine möglichst leichte Bauart notwendig.

3. Anforderungen an das mechanische Modell

a) Muss

Alle relevanten Belastungen abbilden Die Belastungen durch Eigengewicht, Beladung und Seitenwind sollen abgebildet werden.

Für alle Varianten Alle Produktvarianten sollen beinhaltet sein.

Stützkräfte ermittelbar Die auftretenden Stützkräfte sollen mit dem mechanischen Modell ermittelt werden können.

Stützenpositionen parametrisierbar Die Position der modellierten Stützen soll an die tatsächlichen Gegebenheiten angepasst werden können.

b) **Soll**

Dynamische Kräfte Dynamische Kräfte sollen abgebildet werden können.

Optionale Zusatzkräfte Mögliche Zusatzkräfte durch Stöße oder ähnliches sollen bei Bedarf berücksichtigt werden können.

6. Hauptstudie

In der Hauptstudie wird das mechanische Modell entwickelt, aus dem die Bewegungsgleichungen für das System hervorgehen. Die Lösung dieser Gleichungen liefert die Bewegung des Systems. Damit lassen sich die wirkenden Stützkkräfte ermitteln. Auf Basis dieser Daten wird ein geeigneter Algorithmus konzipiert und programmiert. Die Anforderungen aus dem vorangegangenen Kapitel bilden die Rahmenbedingungen für den Algorithmus.

6.1. Mechanisches Modell

Das mechanische Modell wird als dynamisches Modell konzipiert. Neben statischen Kräften, wie Eigengewicht und Beladung, treten auch dynamisch Kräfte, wie Wind und Stöße von Personen, auf. Dennoch liegt das Augenmerk in der ersten Berechnung auf der statischen bzw. quasistatischen Belastung, die im Normalbetrieb auftritt. Für die Auslegung neuer Stützen und anderer Bauteile soll es aber möglich sein, die dynamischen Belastungen mit abbilden zu können.

6.1.1. Lastabschätzung

Zur groben Orientierung wird eine qualitative Abschätzung der auftretenden Lasten bei einem Wohnwagen durchgeführt. Der Wohnwagen ist schemenhaft in Abbildung 6.1 dargestellt. Von dieser Darstellung wird in den nächsten Betrachtungen ausgegangen.

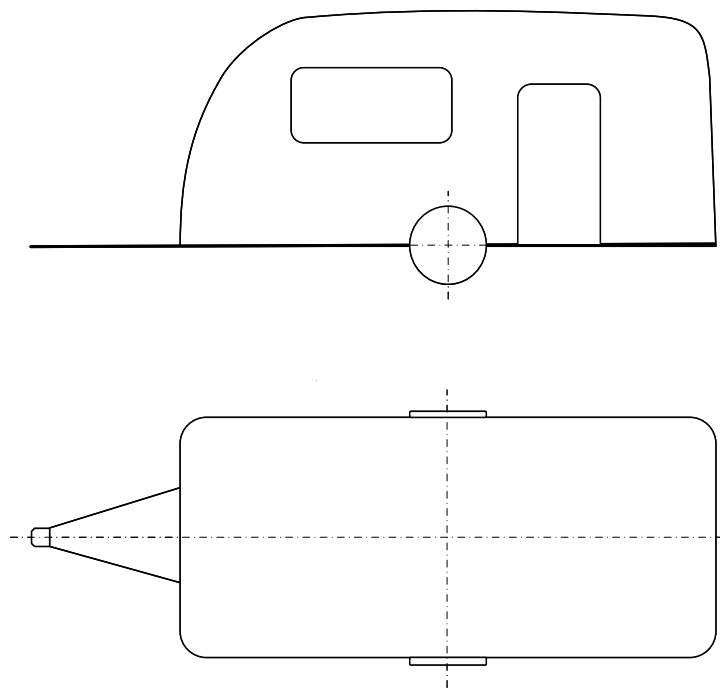


Abbildung 6.1.: Skizze Wohnwagen

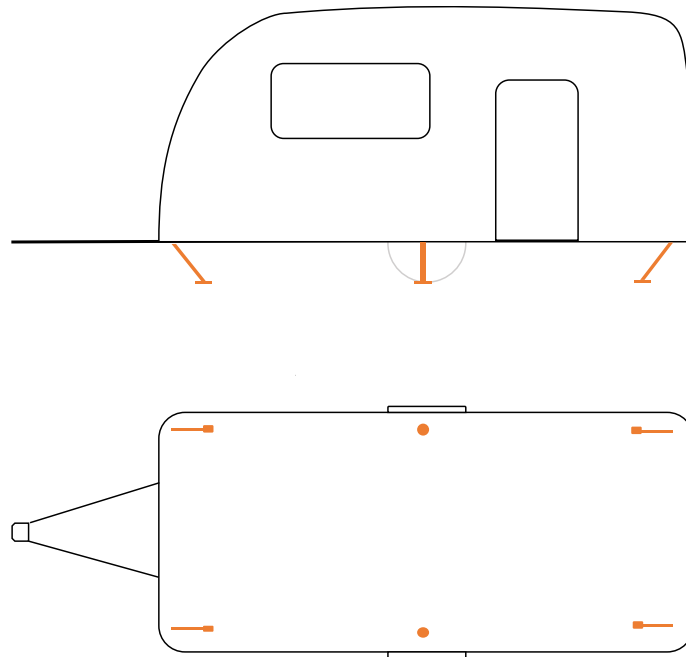


Abbildung 6.2.: Skizze Wohnwagen mit Stützen

Ein Wohnwagen soll mit sechs Stützen zur Nivellierung ausgestattet werden, wie in den Anforderungen im Kapitel 5.2 bereits beschrieben. Zwei Stützen befinden sich an der Achse des Wohnwagens, die restlichen vier werden an den Ecken angebracht. Die beiden Stützen an der Achse tragen die Hauptlast, da bei einem Wohnwagen die mögliche Stützlast an den Eckpunkten sehr beschränkt ist. Die Positionen der Stützen sind in Abbildung 6.2 qualitativ dargestellt.

Zur Veranschaulichung der Belastung werden charakteristische Schnittebenen definiert, die in Abbildung 6.3 dargestellt sind. Die in Abbildung 6.4, 6.5, 6.6 und 6.7 dargestellten Streckenlasten sind rein qualitativ und basieren auf der Abschätzung einer typischen Lastverteilung in Wohnwägen. Möbel sind an den Außenwänden angebracht und schwere Elektrogeräte in Nähe zur Achse des Wohnwagens. Der mittlere Bereich des Wohnwagens ist meist frei für Bewegungsflächen. Die Streckenlasten in den verschiedenen Schnitten werden hier noch einmal aufgeschlüsselt:

Schnittebene A-A Mittiger Schnitt quer zur Fahrtrichtung. Die Möblierung ist meist an der Wand angeordnet und der mittlere Bereich ist als Bewegungsfläche frei,

wodurch sich ein qualitativer Verlauf entsprechend Abbildung ergibt.

Schnittebene B-B Schnitt am Rand quer zur Fahrtrichtung. Da hier die Schnittebene am Rand liegt, kann von einer gleichmäßigeren Verteilung ausgegangen werden: Hier ist das Maximum geringer, da das Chassis in größerer Entfernung zur Achse nicht so stark belastet werden kann.

Schnittebene C-C Mittiger Schnitt in Fahrtrichtung. Hier ergibt sich ein ähnliches Bild wie in Schnittebene A-A.

Schnittebene D-D Schnitt am Rand in Fahrtrichtung. In dieser Ebene ist das Maximum im Achsenbereich, da hier schwere Objekte aufgestellt werden können.

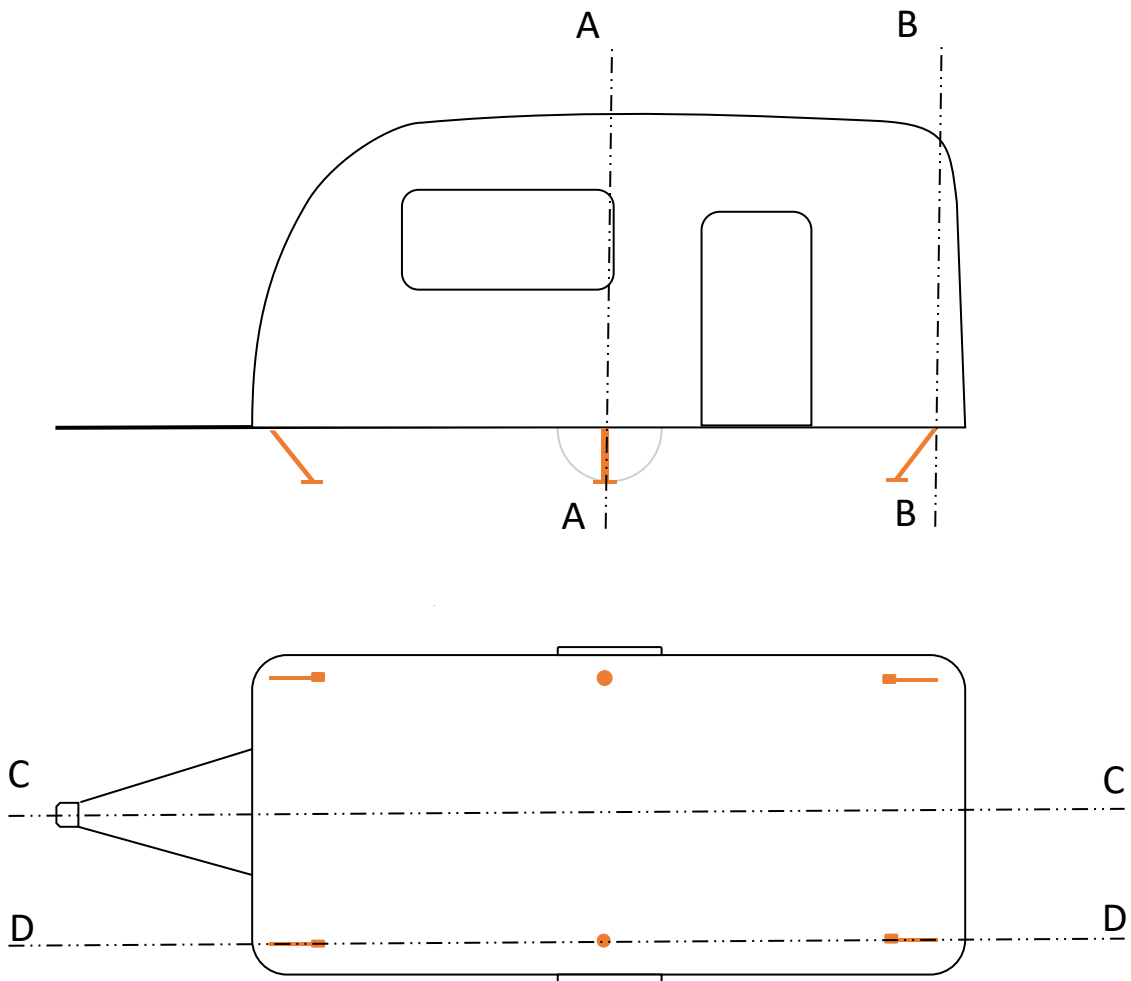


Abbildung 6.3.: Skizze Wohnwagen mit Schnittebenen

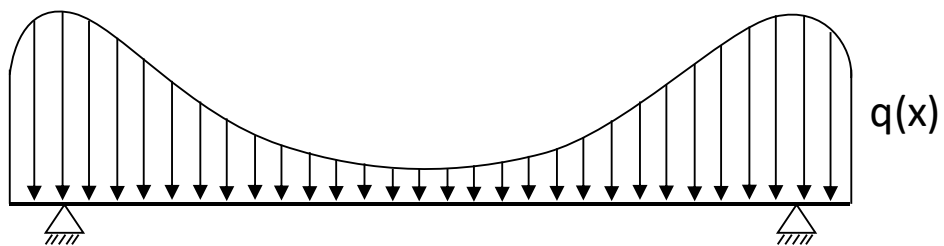


Abbildung 6.4.: Qualitative Streckenlast Schnittebene A-A

6. Hauptstudie

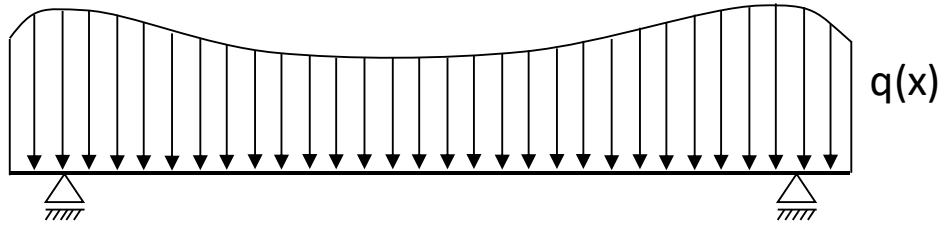


Abbildung 6.5.: Qualitative Streckenlast Schnittebene B-B

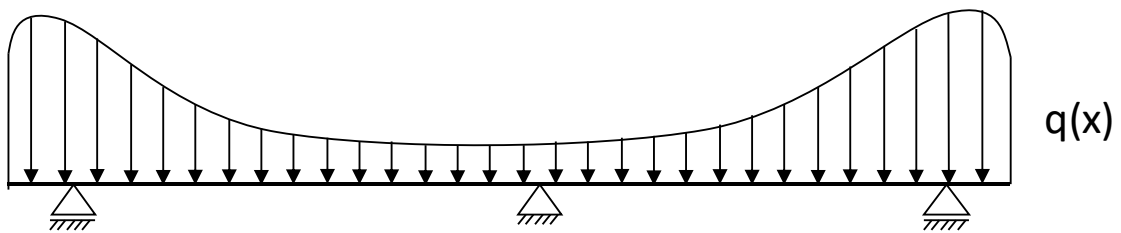


Abbildung 6.6.: Qualitative Streckenlast Schnittebene C-C

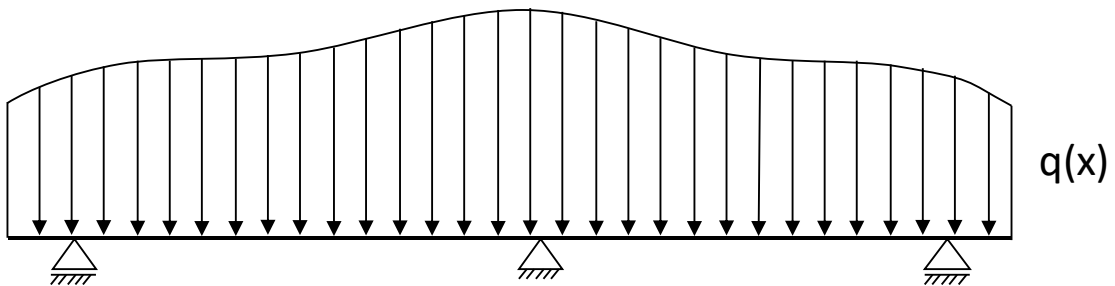


Abbildung 6.7.: Qualitative Streckenlast Schnittebene D-D

Abschätzung der Schwerpunktlage

Da es sich bei einem Wohnwagen um ein Fahrzeug handelt, unterliegt dieser bestimmten Randbedingungen, die für eine Abschätzung der Schwerpunktlage herangezogen werden können. Quer zur Fahrtrichtung kann von einer Schwerpunktlage im Bereich der Symmetrieebene ausgegangen werden. Je nach Beladung und Ausstattung kann die Lage etwas abweichen, für eine erste Annahme ist sie aber ausreichend genau.

In Fahrtrichtung beschränkt die maximale Stützlast des Zugfahrzeuges die Lage des Schwerpunktes, als Abstand von der Achse. Typische Stützlasten bewegen sich im Bereich von 50 – 100 kg. Mit der Gewichtskraft und dem Abstand der Anhängerkupplung zur Achse des Wohnwagens lässt sich, mit einem einfachen Momentengleichgewicht (Gleichung 6.1) um die Wohnwagenachse, der Abstand des Schwerpunkts von der Achse bestimmen. Die beschriebenen Verhältnisse sind in Abbildung 6.8 illustriert. Basierend auf dieser Darstellung kann die Gleichung 6.2 als Abstandsformel für den Schwerpunkt formulieren werden. Die dabei verwendeten Symbole sind in Tabelle 6.1 erläutert.

$$\sum M^{(A)} = 0 \quad (6.1)$$

$$S \cdot a - G \cdot x_S = 0$$

$$x_S = \frac{S \cdot a}{G} \quad (6.2)$$

Symbol	Einheit	Bedeutung
S	N	Stützkraft
G	N	Gewichtskraft
A	N	Achskraft
$M^{(A)}$	Nm	Drehmoment um A
a	m	Abstand zwischen Achse und Anhängerkupplung
x_S	m	Schwerpunktsabstand von der Achse

Tabelle 6.1.: Schwerpunktabschätzung - Symbole

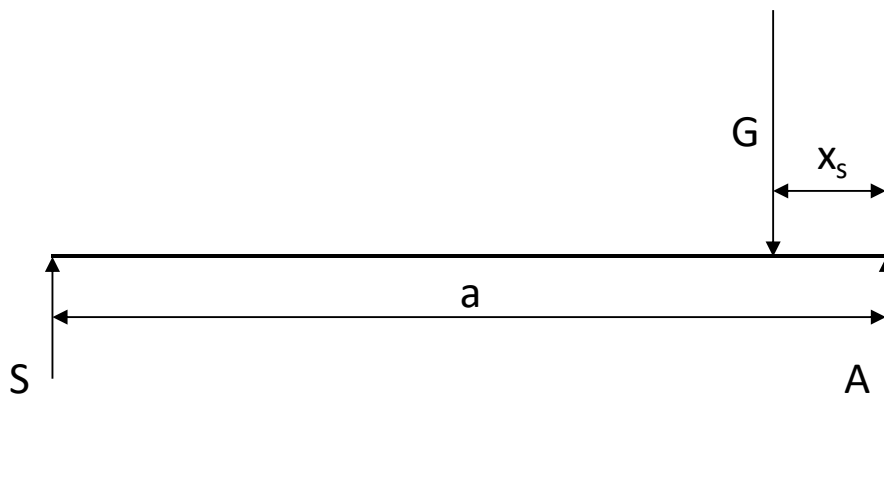


Abbildung 6.8.: Kräfte und Abstände zur Schwerpunktabstimmung

Die Höhe des Schwerpunktes lässt sich nicht so einfach bestimmen. Sie ist sehr stark von Aufbau und Beladung des Wohnwagens abhängig und muss so für jeden Wohnwagen individuell abgeschätzt werden. Die ungefähre Lage des Schwerpunktes ist in Abbildung 6.9 dargestellt.

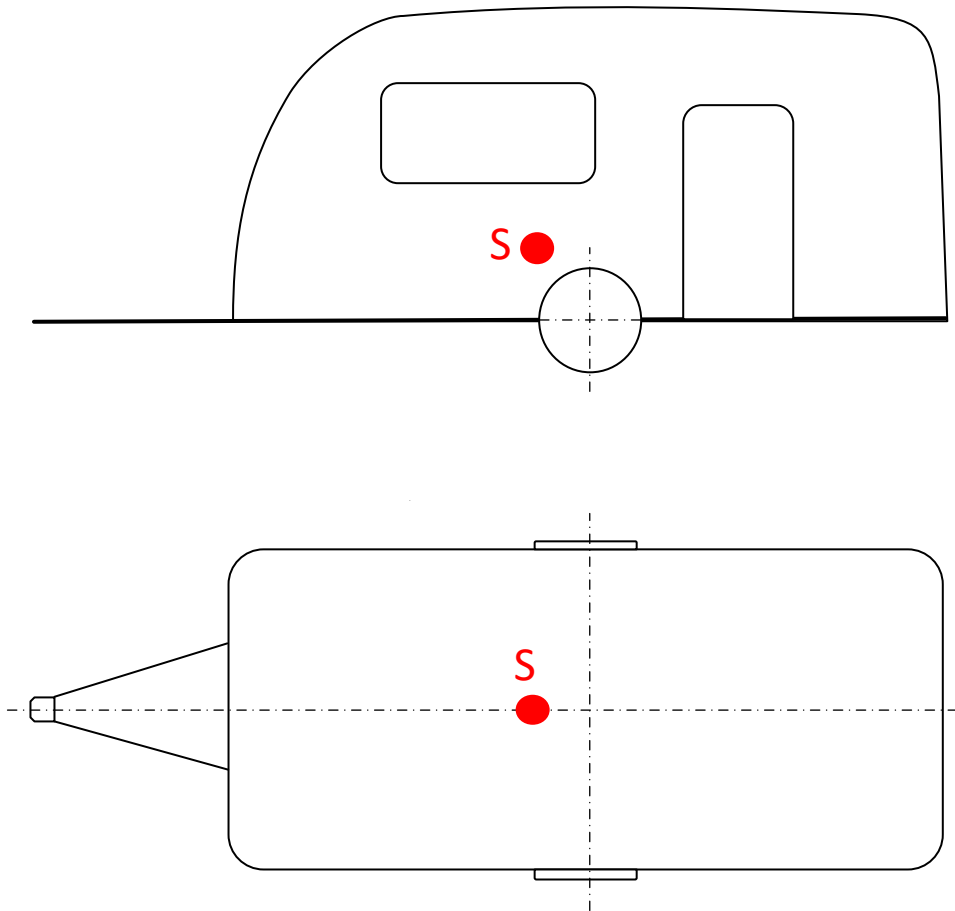


Abbildung 6.9.: Lage des Schwerpunktes

6.1.2. Dynamisches Modell

Zur Berechnung der auftretenden Kräfte wird ein dynamisches Modell angefertigt. Als Grundsatz für die Modellerstellung gilt, dass das Modell möglichst einfach gehalten werden soll. Wenn alle Anforderungen erfüllt sind, ist das vollkommen ausreichend. Die getroffenen Vereinfachungen sind im nächsten Abschnitt behandelt. Weiters gilt es die Bewegungsgleichungen aufzustellen und einer Lösung zuzuführen. Zur Identifizierung der einzelnen Stützen werden diese mit Nummern versehen. Die Nummerierung ist in Abbildung 6.10 zu sehen.

Modellbildung und Abstraktion

Als erste Vereinfachung wird der Wohnwagen als ein Objekt mit einem Schwerpunkt und einem Massenträgheitstensor angesehen. Die ungefähre Lage des Schwerpunktes wurde bereits im Kapitel 6.1.1 beschrieben. Der Massenträgheitstensor kann für eine erste Annäherung von einem Quader, mit ähnlichen Abmessungen wie die des Wohnwagens, abgeleitet werden. Mehr dazu im Abschnitt „Annahmen“.

Modell der Stützen Die Stützen werden als Feder-Dämpfer-Elemente modelliert. Vorrangig ist für das dynamische Modell die Nachgiebigkeit an den Stützen maßgebend und weniger, wie sich der Wohnwagen als solcher verformt. Vor diesem Hintergrund wird die Steifigkeit und die Materialdämpfung des Wohnwagens den Stützen hinzugerechnet. Die Stützen bilden also, zusätzlich zu ihrer eigenen Steifigkeit und Dämpfung, die des Wohnwagens am jeweiligen Montagepunkt der Stützen ab. Die Elastizität beseitigt das Problem der Überbestimmtheit bei einer Stützenanzahl größer als drei. Das Modell der Stützen ist in Abbildung 6.11 dargestellt. Die Federelemente sind als lineare Federn und die Dämpfer als viskose Dämpfer modelliert.

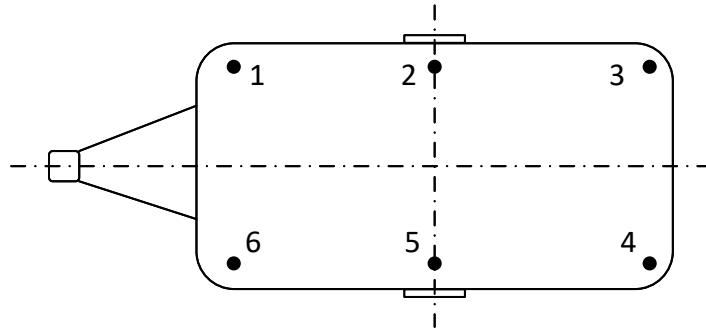


Abbildung 6.10.: Stuetzennummerierung

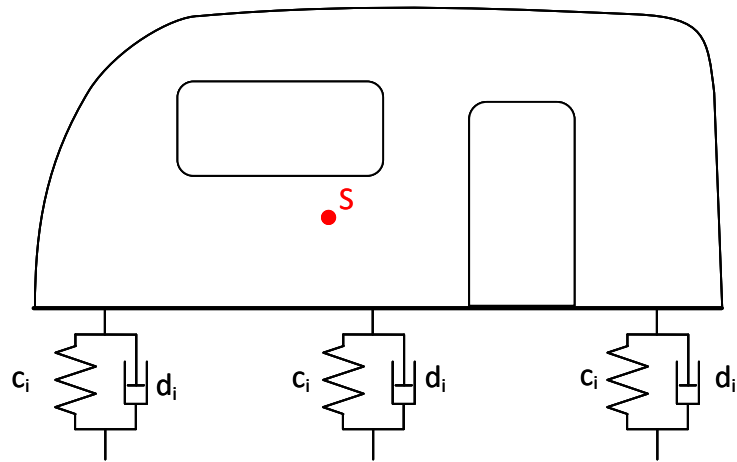


Abbildung 6.11.: Stuetzenmodell

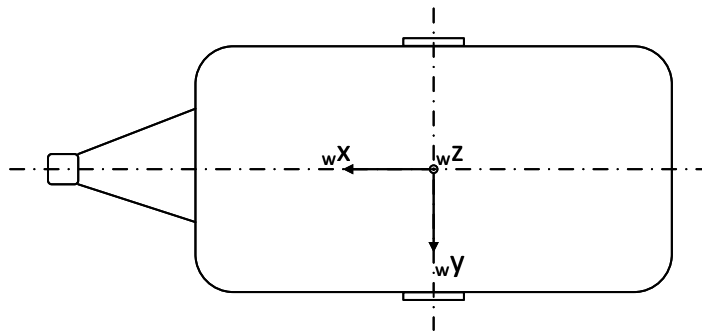


Abbildung 6.12.: Wagenkoordinatensystem

Mathematisches Modell

Aus dem Modell für den Wohnwagen werden nun mathematische Zusammenhänge formuliert, die auf die Bewegungsgleichungen für das System führen. Die Kräfte, Momente und Abstände werden im dreidimensionalen Raum als Vektoren angegeben. Damit dies möglich ist, müssen Koordinatensysteme definiert werden.

Koordinatensysteme Es werden drei verschiedene Koordinatensysteme festgelegt. Diese verdeutlichen die Lagen von Gewichtskraft, Boden und Wohnwagen. Das Inertialkoordinatensystem I , das Bodenkoordinatensystem B und das Wagenkoordinatensystem W . Im Inertialsystem ist die z -Achse immer in lotrechter Richtung. Bei einem perfekt nivellierten Wohnwagen ist das Inertialkoordinatensystem um den Bodenabstand in z -Richtung zum Wagenkoordinatensystem verschoben, ansonsten besitzen die korrespondierenden Achsen die gleiche Orientierung in den beiden Koordinatensystemen. Die Lage des Wagenkoordinatensystems ist in Abbildung 6.12 dargestellt. Die y -Achse liegt dabei in der Wagenachse und die x -Achse in der Symmetrieebene in Fahrtrichtung (Wagenlängsachse). Bei einem horizontalen Boden entspricht das Bodenkoordinatensystem dem Inertialkoordinatensystem. Die x - y -Ebene stimmt mit dem Boden überein. Bei einer Steigung beschreibt das Bodenkoordinatensystem die Lage des Bodens zum Inertialsystem.

Zur Umrechnung zwischen den Koordinatensystemen werden Drehungsmatrizen

definiert. Für die Transformation vom Inertialsystem in das Wagenkoordinatensystem ergibt sich folgende Matrix:

$${}^{IW}\mathbf{A} = \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ \sin \alpha \sin \beta & \cos \alpha & \cos \beta \sin \alpha \\ \cos \alpha \sin \beta & -\sin \alpha & \cos \alpha \cos \beta \end{pmatrix} \quad (6.3)$$

Dabei beschreibt β die Drehung um die y-Koordinate des Inertialsystems und α die Drehung um die, aus der ersten Drehung entstandene, x-Achse. Die beiden Winkel sind Freiheitsgrade des Systems und damit zeitabhängig.

Die Matrix für die Umrechnung zwischen Inertialsystem und Bodenkoordinatensystem ist analog zur vorher gezeigten Matrix. Der Boden verändert sich in der Regel nicht zum Inertialsystem, womit die Winkel in einer solchen Drehmatrix als konstant anzusehen sind. Mit diesem Koordinatensystem lassen sich die Stützenlängen darstellen. Die Drehmatrix ist nachfolgend dargestellt. Konstante Winkel sind mit einem Index 0 versehen.

$${}^{IB}\mathbf{A} = \begin{pmatrix} \cos \beta_0 & 0 & -\sin \beta_0 \\ \sin \alpha_0 \sin \beta_0 & \cos \alpha_0 & \cos \beta_0 \sin \alpha_0 \\ \cos \alpha_0 \sin \beta_0 & -\sin \alpha_0 & \cos \alpha_0 \cos \beta_0 \end{pmatrix} \quad (6.4)$$

Bei einer Transformation vom Wagenkoordinatensystem in das Inertialsystem oder vom Bodenkoordinatensystem ins Inertialsystem kann die transponierte Matrix, der vorher gezeigten Matrizen, verwendet werden:

$${}^{WI}\mathbf{A} = {}^{IW}\mathbf{A}^T \quad (6.5)$$

$${}^{BI}\mathbf{A} = {}^{IB}\mathbf{A}^T \quad (6.6)$$

Abstände Mit den Koordinatensystemen können die Abstände für den Schwerpunkt und die Stützen definiert werden. Inklusive der Höhe des Wohnwagens, die als dritter Freiheitsgrad zeitabhängig ist. Es ist zweckmäßig die Vektoren in jenem Koordinatensystem aufzustellen, in dem sie am einfachsten zu formulieren sind und anschließend eine Koordinatentransformation in das gewünschte Koordinatensystem

durchzuführen. Das geschieht mit einer einfachen Matrizenmultiplikation. Als Index vor dem Vektor wird das aktuelle Koordinatensystem angeführt, in dem er dargestellt ist.

Abstand zum Boden \mathbf{r}_{AM} im Bodenkoordinatensystem

$${}^B\mathbf{r}_{AM} = \begin{pmatrix} 0 \\ 0 \\ h_{Wagen} \end{pmatrix} \quad (6.7)$$

Umrechnung ins Inertialsystem

$${}^I\mathbf{r}_{AM} = {}^{IB}\mathbf{A} {}^B\mathbf{r}_{AM} \quad (6.8)$$

Schwerpunkt im Wagenkoordinatensystem

$${}^W\mathbf{r}_S = \begin{pmatrix} x_S \\ y_S \\ z_S \end{pmatrix} \quad (6.9)$$

Umrechnung ins Inertialsystem

$${}^I\mathbf{r}_S = {}^{IW}\mathbf{A} {}^W\mathbf{r}_S \quad (6.10)$$

Stützenabstände in Wagenkoordinaten

$${}^W\mathbf{r}_{St_i} = \begin{pmatrix} x_{St_i} \\ y_{St_i} \\ 0 \end{pmatrix} \quad i = 1 \dots 6 \quad (6.11)$$

Umrechnung ins Inertialsystem

$${}^I\mathbf{r}_{St_i} = {}^{IW}\mathbf{A} {}^W\mathbf{r}_{St_i} + {}^I\mathbf{r}_{AM} \quad i = 1 \dots 6 \quad (6.12)$$

Umrechnung in Wagenkoordinaten

$${}^B\mathbf{r}_{St_i} = {}^{BI}\mathbf{A} {}^I\mathbf{r}_{St_i} \quad i = 1 \dots 6 \quad (6.13)$$

Neben den Abständen für die bereits angeführten Kräfte werden Abstände für Zusatzkräfte eingeführt, an denen Wind-, Stoß- und ähnliche Kräfte angreifen können. In jeder Koordinatenrichtung im Wagenkoordinatensystem wird dafür eine mögliche Position eingeführt. Da man Kräfte entlang ihrer Wirkungslinie verschieben kann, ist die Komponente des Abstandsvektors in Krafrichtung immer mit 0 angegeben.

$${}^W\mathbf{r}_{ZZ} = \begin{pmatrix} Wx_{zz} \\ Wy_{zz} \\ 0 \end{pmatrix} \quad (6.14)$$

$${}^W\mathbf{r}_{ZY} = \begin{pmatrix} Wx_{zy} \\ 0 \\ Wz_{zy} \end{pmatrix} \quad (6.15)$$

$${}^W\mathbf{r}_{ZX} = \begin{pmatrix} 0 \\ Wx_{zx} \\ Wz_{zx} \end{pmatrix} \quad (6.16)$$

Umrechnung ins Inertialsystem

$${}^I\mathbf{r}_{ZZ} = {}^{IW}\mathbf{A} \cdot {}^W\mathbf{r}_{ZZ} \quad (6.17)$$

$${}^I\mathbf{r}_{ZY} = {}^{IW}\mathbf{A} \cdot {}^W\mathbf{r}_{ZY} \quad (6.18)$$

$${}^I\mathbf{r}_{ZX} = {}^{IW}\mathbf{A} \cdot {}^W\mathbf{r}_{ZX} \quad (6.19)$$

Für die Feder-Dämpfer-Elemente muss die Längenänderung der Stützen bekannt sein, sowie die Geschwindigkeit mit der die Änderung passiert. Diese Änderung ist in Gleichung 6.20 allgemein für alle Stützen formuliert.

Längenänderung der Stützen

$$\Delta h_i = \frac{h_{Wagen_0} - h_{di} - B r_{St_{i,z}}}{\cos(\alpha - \alpha_0) \cdot \cos(\beta - \beta_0)} \quad i = 1 \dots 6 \quad (6.20)$$

Die Formelzeichen für diese Gleichung sind in Tabelle 6.2 angeführt.

Die Geschwindigkeit der Längenänderung der einzelnen Stützen i kann durch eine einfache Zeitableitung erhalten werden:

Symbol	Einheit	Bedeutung
Δh_i	m	Abweichung von der Ausgangslänge der Stütze i
h_{Wagen_0}	m	Höhe des Wohnwagens im unbelasteten Zustand
h_{di}	m	Längenabweichung der Stütze i von h_{Wagen_0}
$B^rSt_{i,z}$	m	Normalabstand der Stütze i vom Wagen zum Boden
α	rad	Aktueller Neigungswinkel des Wohnwagens, siehe Gleichung 6.3
β	rad	Aktueller Neigungswinkel des Wohnwagens, siehe Gleichung 6.3
α_0	rad	Aktueller Neigungswinkel des Bodens, siehe Gleichung 6.4
β_0	rad	Aktueller Neigungswinkel des Bodens, siehe Gleichung 6.4

Tabelle 6.2.: Längenänderung der Stützen - Symbole

$$v_i = \frac{d\Delta h_i}{dt} \quad i = 1 \dots 6 \quad (6.21)$$

Kräfte Für den Schwerpunktsatz müssen die angreifenden Kräfte aufgestellt und zur Aufsummierung in einem Koordinatensystem dargestellt werden. Die einfachste aufzustellende Kraft ist die Gewichtskraft, die im Inertialsystem folgenden Gestalt hat:

$${}_I\mathbf{F}_G = \begin{pmatrix} 0 \\ 0 \\ -m_W \cdot g \end{pmatrix} \quad (6.22)$$

Die nächsten wesentlichen Kräfte sind die der jeweiligen Stütze. Die Kräfte in x- und y-Richtung des Wagenkoordinatensystems werden, aus Gründen der Bestimmtheit des Systems, zu drei Kräften zusammengefasst. Es wird angenommen, dass die beiden mittleren Stützen als Festlager wirken. Daher wirken zwei Kräfte in x-Richtung auf die mittleren Stützen und in y-Richtung wirkt nur eine Kraft auf eine der beiden mittleren Stützen. In dieser Richtung wird die tieferliegende Stütze als Festlager

angesehen. Welche das ist, lässt sich allgemein nicht festlegen, deshalb wird eine eigene Achskraft definiert, die diese Kraft darstellt.

Stützkkräfte in Wagenkoordinaten

$${}^W\mathbf{F}_{St_i} = \begin{pmatrix} 0 \\ 0 \\ c_i \cdot \Delta h_i + d_i \cdot v_i \end{pmatrix} \quad i = 1, 3, 4, 6 \quad (6.23)$$

$${}^W\mathbf{F}_{St_i} = \begin{pmatrix} {}^W F_{St_i,x} \\ 0 \\ c_i \cdot \Delta h_i + d_i \cdot v_i \end{pmatrix} \quad i = 2, 5 \quad (6.24)$$

$${}^W\mathbf{F}_{Achse} = \begin{pmatrix} 0 \\ {}^W F_{Achse,y} \\ 0 \end{pmatrix} \quad (6.25)$$

In den Formeln sind nun auch die Dämpfungskonstanten d_i und die Federkonstanten c_i für die einzelnen Stützen enthalten. Die Kräfte ${}^W F_{St_{2,x}}$, ${}^W F_{St_{5,x}}$ und ${}^W F_{Achse,y}$ sind die vorher beschriebenen Kräfte in der x-y-Ebene des Wagenkoordinatensystems.

Umrechnung ins Inertialsystem

$${}^I\mathbf{F}_{St_i} = {}^{IW}\mathbf{A} \cdot {}^W\mathbf{F}_{St_i} \quad i = 1 \dots 6 \quad (6.26)$$

$${}^I\mathbf{F}_{Achse} = {}^{IW}\mathbf{A} \cdot {}^W\mathbf{F}_{Achse} \quad (6.27)$$

Wie bereits unter den Abständen beschrieben, werden Zusatzkräfte in allen drei Koordinatenrichtungen eingeführt:

$${}^W\mathbf{F}_{ZZ} = \begin{pmatrix} 0 \\ 0 \\ {}^W F_{ZZz} \end{pmatrix} \quad (6.28)$$

$${}^W\mathbf{F}_{ZY} = \begin{pmatrix} 0 \\ {}^WF_{Zyy} \\ 0 \end{pmatrix} \quad (6.29)$$

$${}^W\mathbf{F}_{ZX} = \begin{pmatrix} {}^WF_{Zxx} \\ 0 \\ 0 \end{pmatrix} \quad (6.30)$$

Umrechnung ins Inertialsystem

$${}_I\mathbf{F}_{ZZ} = {}^{IW}\mathbf{A} {}^W\mathbf{F}_{ZZ} \quad (6.31)$$

$${}_I\mathbf{F}_{ZY} = {}^{IW}\mathbf{A} {}^W\mathbf{F}_{ZY} \quad (6.32)$$

$${}_I\mathbf{F}_{ZX} = {}^{IW}\mathbf{A} {}^W\mathbf{F}_{ZX} \quad (6.33)$$

Schwerpunktsatz Für die Aufstellung des Schwerpunktsatzes muss, neben den zuvor aufgestellten Kräften, die Schwerpunktbeschleunigung ermittelt werden. Das lässt sich mathematisch mit der zweiten Ableitung des Schwerpunktabstandes nach der Zeit berechnen.

$${}_I\mathbf{a}_W = \frac{d^2}{dt^2} {}_I\mathbf{r}_S \quad (6.34)$$

Mit der Beschleunigung kann der Schwerpunktsatz formuliert werden zu:

$$m_W \cdot {}_I\mathbf{a}_W = \sum_{i=1}^6 {}_I\mathbf{F}_{St_i} + {}_I\mathbf{F}_{Achse} + {}_I\mathbf{F}_G + {}_I\mathbf{F}_{ZZ} + {}_I\mathbf{F}_{ZY} + {}_I\mathbf{F}_{ZX} \quad (6.35)$$

Drallsatz Für den Drallsatz müssen die Winkelgeschwindigkeiten mit der die Drehungen vonstatten gehen, ermittelt werden. Diese Berechnung erfolgt über die sogenannte Tilde-Matrix (Rill und Schaeffer, 2014, S.12ff), entsprechend folgender Formel:

$${}_I^{IW}\tilde{\boldsymbol{\omega}} = {}^{IW}\mathbf{A}^T {}^{IW}\dot{\mathbf{A}} \quad (6.36)$$

In der Tilde-Matrix ergeben sich damit die Winkelgeschwindigkeiten in der Form:

$${}^I W \tilde{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (6.37)$$

Womit sich der Vektor der Winkelgeschwindigkeiten formulieren lässt:

$${}^I W \boldsymbol{\omega} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (6.38)$$

Der Vektor kann wieder in Wagenkoordinaten umgerechnet werden:

$${}^I W \boldsymbol{\omega} = {}^W I \mathbf{A} {}^I W \boldsymbol{\omega} \quad (6.39)$$

Als Trägheitstensor wird ein allgemeiner Tensor definiert:

$$\Theta_{WM} = \begin{pmatrix} \Theta_{WMxx} & \Theta_{WMxy} & \Theta_{WMxz} \\ \Theta_{WMyx} & \Theta_{WMyy} & \Theta_{WMyz} \\ \Theta_{WMzx} & \Theta_{WMzy} & \Theta_{WMzz} \end{pmatrix} \quad (6.40)$$

Damit lässt sich der Drallsatz formulieren als

$$\begin{aligned} \frac{d}{dt} \Theta_{WM} {}^I W \boldsymbol{\omega} + {}^I W \boldsymbol{\omega} \times (\Theta_{WM} {}^I W \boldsymbol{\omega}) &= \sum_{i=1}^6 {}^W \mathbf{r}_{St_i} \times {}^W \mathbf{F}_{St_i} + \\ &+ {}^W \mathbf{r}_{ZZ} \times {}^W \mathbf{F}_{ZZ} + {}^W \mathbf{r}_{ZY} \times {}^W \mathbf{F}_{ZY} + {}^W \mathbf{r}_{ZX} \times {}^W \mathbf{F}_{ZX} + {}^W \mathbf{r}_S \times {}^W \mathbf{F}_G \end{aligned} \quad (6.41)$$

Bestimmen der Bewegungsgleichungen Mit dem Drall- und Schwerpunktsatz lassen sich die unbekanntenen Kräfte ${}^W F_{St_{2,x}}$, ${}^W F_{St_{5,x}}$ und ${}^W F_{Achse,y}$ bestimmen. Diese hängen somit nur mehr von den drei Freiheitsgraden α , β und h_{Wagen} ab. In Summe ergeben Drall- und Schwerpunktsatz ein Differenzialgleichungssystem, das die Bewegung des Wohnwagens in den drei Freiheitsgraden beschreibt, die Bewegungsgleichungen.

Annahmen

Zur Lösung der Bewegungsgleichungen wird ein numerischer Ansatz gewählt und in MATLAB umgesetzt. Für diese numerische Lösung sind vorweg einige Werte zu bestimmen, die in der Herleitung der Bewegungsgleichungen nicht von Relevanz waren.

Gewichtskraft Für die Gewichtskraft muss neben der Erdbeschleunigung eine Wagenmasse angegeben werden. Die Wagenmasse ist im Allgemeinen zwar bekannt, aber von Wohnwagen zu Wohnwagen unterschiedlich. Deshalb wird die Masse für die erste Berechnung so abgeschätzt, dass sie eine typische Wohnwagenmasse widerspiegelt. Selbiges gilt für die Koordinaten des Schwerpunkts.

Werte:

$$g = 9.81 \frac{\text{m}}{\text{s}^2}$$
$$m_W = 1500 \text{ kg}$$

Abschätzung des Schwerpunkts in Wagenkoordinaten:

$$x_S = 0.2 \text{ m}$$
$$y_S = 0 \text{ m}$$
$$z_S = 0.5 \text{ m}$$

Steifigkeit und Dämpfung der Stützen Die Werte für die Steifigkeits- und Dämpfungskonstanten der einzelnen Stützen hängen vom Aufbau des jeweiligen Wohnwagens ab und erfordern einiges an Aufwand bei der Bestimmung. Für den Moment wird versucht möglichst realistische Werte zu finden, die die Verhältnisse im Wohnwagen widerspiegeln. Die gewählten Feder- und Dämpfungskonstanten sind in Tabelle 6.3 angegeben.

Stütze i	Federkonstante c_i	Dämpfungskonstante d_i
1	50 000 $\frac{\text{N}}{\text{m}}$	10 000 $\frac{\text{Ns}}{\text{m}}$
2	300 000 $\frac{\text{N}}{\text{m}}$	10 000 $\frac{\text{Ns}}{\text{m}}$
3	20 000 $\frac{\text{N}}{\text{m}}$	10 000 $\frac{\text{Ns}}{\text{m}}$
4	20 000 $\frac{\text{N}}{\text{m}}$	10 000 $\frac{\text{Ns}}{\text{m}}$
5	300 000 $\frac{\text{N}}{\text{m}}$	10 000 $\frac{\text{Ns}}{\text{m}}$
6	50 000 $\frac{\text{N}}{\text{m}}$	10 000 $\frac{\text{Ns}}{\text{m}}$

Tabelle 6.3.: Feder- und Dämpfungskonstanten

Massenträgheitstensor Die Massenträgheit wird für die ersten Berechnungen ebenfalls abgeschätzt. Die Werte werden in kg m^2 angegeben:

$$\Theta_{WM} = \begin{pmatrix} 5000 & 0 & 0 \\ 0 & 6000 & 0 \\ 0 & 0 & 6500 \end{pmatrix}$$

Lage der Stützen Die Lage der Stützen hängt von den Montagepunkten am Wohnwagen ab. Daher auch hier die Wahl von möglichst allgemein gültigen Werten (in m):

$$\begin{aligned} w\mathbf{r}_{St_1} &= \begin{pmatrix} 2,5 \\ 1 \\ 0 \end{pmatrix} & w\mathbf{r}_{St_2} &= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\ w\mathbf{r}_{St_3} &= \begin{pmatrix} -2,5 \\ 1 \\ 0 \end{pmatrix} & w\mathbf{r}_{St_4} &= \begin{pmatrix} -2,5 \\ -1 \\ 0 \end{pmatrix} \\ w\mathbf{r}_{St_5} &= \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} & w\mathbf{r}_{St_6} &= \begin{pmatrix} 2,5 \\ -1 \\ 0 \end{pmatrix} \end{aligned}$$

Zusatzkräfte Die Zusatzkräfte werden zu Beginn auf null gesetzt. Bei späteren Untersuchungen können über diese Kräfte Störungen eingebracht werden.

Kräfte:

$${}^w\mathbf{F}_{ZZ} = {}^w\mathbf{F}_{ZY} = {}^w\mathbf{F}_{ZX} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Angriffspunkte:

$${}^w\mathbf{r}_{ZZ} = {}^w\mathbf{r}_{ZZ} = {}^w\mathbf{r}_{ZZ} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Anfangsbedingungen Als Anfangsbedingungen werden folgende Werte festgelegt:

$$h_{Wagen_0} = 0.3 \text{ m}$$

$$\alpha_0 = 5^\circ$$

$$\beta_0 = 5^\circ$$

Lösung der Bewegungsgleichungen

Die Lösung der Bewegungsgleichungen erfolgt mit MATLAB. Darin sind verschiedene Löser für gewöhnliche Differentialgleichungen implementiert, wie sie bereits im Kapitel 2.4.1 behandelt wurden. Für das vorliegende Problem kann der Löser „ode45“ verwendet werden. Die daraus resultierenden Lösungen für die Bewegungsgrößen und Stützkkräfte sind in den Abbildungen 6.13, 6.14 und 6.15 grafisch dargestellt.

Die Bewegung des Wohnwagens kann an den Bewegungsgrößen α , β und h_{Wagen} abgelesen werden. Diese Größen sind in Abbildung 6.13 dargestellt. Aus diesen Diagrammen lässt sich leicht erkennen, dass eine Bewegung nur am Anfang der

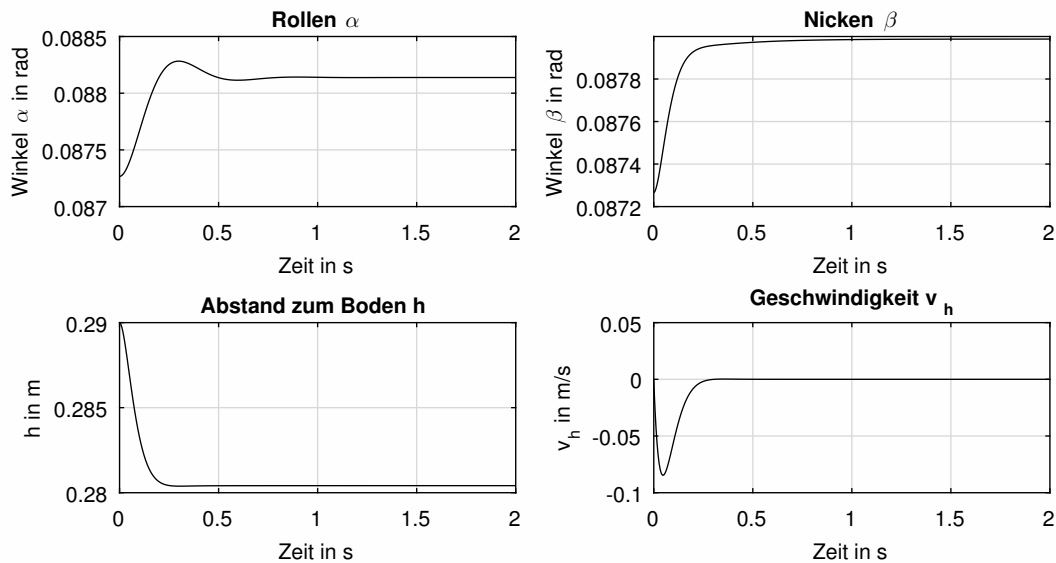


Abbildung 6.13.: Bewegung des Wohnwagens bei den getroffenen Annahmen

Simulation stattfindet, solange bis sich das System auf die statische Ruhelage eingeschwungen hat. Zur Verdeutlichung dieses Umstandes ist die Geschwindigkeit der Wagenhöhe v_h exemplarisch für alle Geschwindigkeiten dargestellt. Auch dort erkennt man, dass eine Bewegung nur bis zum eingeschwungenen Zustand stattfindet.

Sobald der eingeschwungene Zustand erreicht ist, ändern sich die Stützkkräfte nicht mehr. Der Einschwingvorgang ist nicht von Interesse, da die Stützen zu Beginn der Simulation nicht belastet sind. Die tatsächlich auftretenden Stützenkräfte sind jene ab einer Zeit von > 1 s. Die in Abbildung 6.14 dargestellten Stützenkräfte sind jene Kräfte, die in Richtung der z-Achse des Wagenkoordinatensystems wirken.

Die Kräfte in der x-y-Ebene des Wagenkoordinatensystems sind in Abbildung 6.15 dargestellt. Bei einem negativen Vorzeichen ist die Orientierung der Kraft entgegen der jeweiligen Achse. Für die Gesamtkraft der Stützen 2 und 5 müssen die Kräfte aus Abbildung 6.14 und Abbildung 6.15 vektoriell addiert werden.

Somit ist die Lösung für die getroffenen Annahmen gefunden. Die Annahmen können für verschiedene Szenarien variiert werden, die von Interesse für die Kräfte auf den Wohnwagen, beziehungsweise seine Stützen sein könnten. Beispielsweise wurde die

6. Hauptstudie

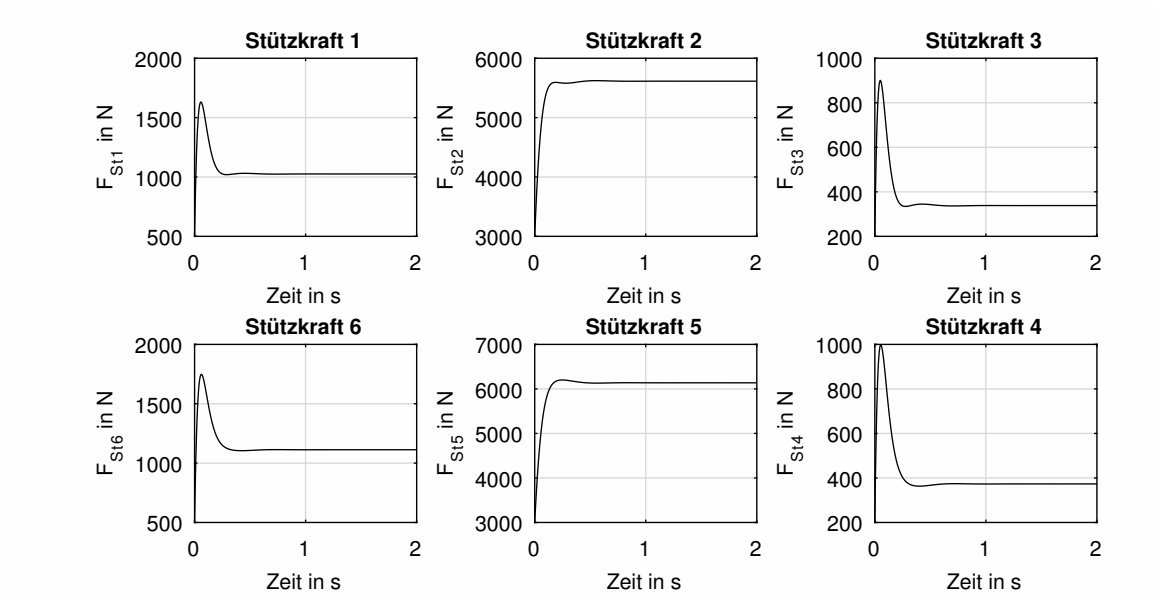


Abbildung 6.14.: Auftretende Stützkräfte normal zum Wohnwagenboden

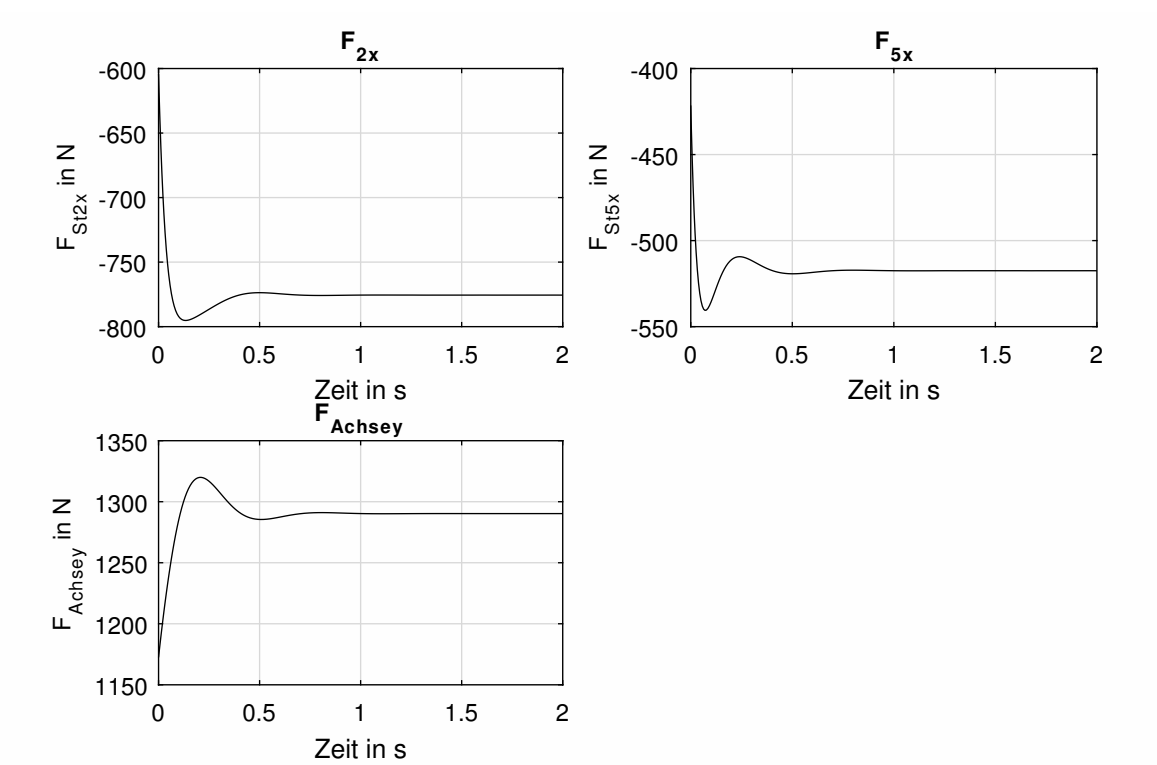


Abbildung 6.15.: Kräfte in x-y-Ebene des Wagenkoordinatensystems

Möglichkeit Windkräfte mit einzubeziehen noch nicht ausgeschöpft. Vorstellbar wäre auch, dass sich ein Mensch gegen den Wohnwagen lehnt. All das kann mit einer einfachen Veränderung der Parameter abgebildet werden und ist vom aufgestellten mechanischen Modell darstellbar.

Das für sechs Stützen aufgestellte Modell kann mit wenigen Anpassungen auf vier Stützen geändert werden. Im Wesentlichen kann die Steifigkeit und die Dämpfung der jeweiligen Stützen auf Null gesetzt werden. Eine Lösung müsste für die Seitenführungskräfte gefunden werden, die ohne Stützen nicht mehr an der mittleren Achse angreifen können. Diese Kräfte müssen auf die vier verbleibenden Stützen aufgeteilt werden. Mit diesen Änderungen kann die Belastung bei einem Wohnmobil mit vier Stützen ebenfalls betrachtet werden.

Einen speziellen Fall stellt ein Wohnwagen dar, der nur vier Stützen besitzt und damit nicht aus der Achse gehoben werden kann. In diesem Fall fungieren die Räder als Stützen 2 und 5. Mit der geeigneten Steifigkeit und Dämpfung für die Räder kann hier die Variante mit vier Stützen analog zur Variante mit sechs Stützen berechnet werden.

6.1.3. Sonderfall statisches Modell

Bereits bei der Auswertung des dynamischen Modells wurde klar, dass in vielen Fällen eine statische Betrachtung des Problems ausreichend ist. Der Vorteil des statischen Modells besteht in einem wesentlich geringerem Rechenaufwand. Die Ergebnisse müssen nicht für ein gesamtes Zeitintervall ermittelt werden, sondern nur für die statische Ruhelage. Der Nachteil liegt darin, dass keine dynamischen Phänomene abgebildet werden können. Beispielsweise kann eine Windkraft nur mit einer statischen Belastung modelliert werden, wodurch die eine zeitliche Veränderung der Windkraft nicht berücksichtigt wird.

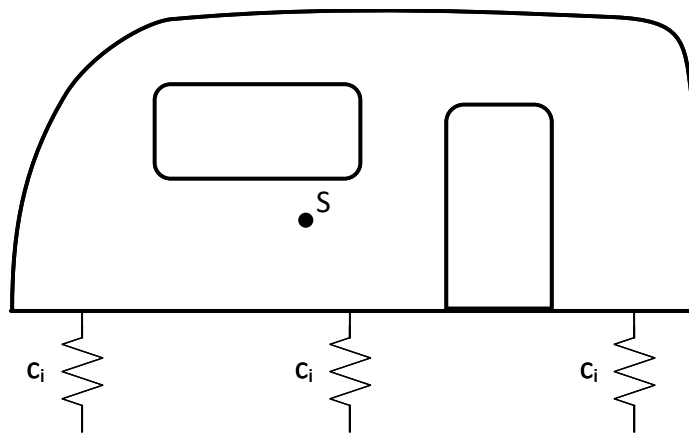


Abbildung 6.16.: Modell der Stützen für die statische Betrachtung

Vereinfachungen

Für den statischen Ansatz verändert sich das Modell des Wohnwagens, beziehungsweise jenes der Stützen. Die Feder-Dämpfer-Elemente können als reine Federelemente ausgeführt werden (siehe Abbildung 6.16), da die Dämpfung nur von der Längenänderung der Stützen abhängig ist und bei der statischen Variante keine Längenänderung auftritt. Statische Unbestimmtheit ist aber weiterhin vorhanden und wird durch die nachgiebigen Stützen ausgeglichen.

Berechnung

Bei der Berechnung interessieren vor allem die unbekanntenen Stützekräfte. Diese Kräfte lassen sich mit den statischen Gleichgewichtsbedingungen für die äußeren Kräfte (Gleichung 6.42) und die äußeren Momente (Gleichung 6.43) um einen Bezugspunkt Q ermitteln. Für die weiteren Berechnungen wird Q im Koordinatenursprung des Inertialsystems angenommen.

$$\sum_j \mathbf{F}_j^a = 0 \quad (6.42)$$

$$\sum_k \mathbf{M}_k^{(Q)} = 0 \quad (6.43)$$

Die Koordinatensysteme und Abstände, wie sie im Kapitel 6.1.2 Mathematisches Modell eingeführt wurden, können für die statische Berechnung beibehalten werden, mit dem Unterschied, dass es keine Zeitabhängigkeiten mehr vorhanden sind.

Die Kräfte für das System sind ebenfalls analog zu Kapitel 6.1.2. Eine Änderung ergibt sich nur bei den Stützenkräften, da diese, wie bereits beschrieben, keine Dämpfung mehr besitzen.

Stützkräfte in Wagenkoordinaten

$${}^w\mathbf{F}_{St_i} = \begin{pmatrix} 0 \\ 0 \\ c_i \cdot \Delta h_i \end{pmatrix} \quad i = 1, 3, 4, 6 \quad (6.44)$$

$${}^w\mathbf{F}_{St_i} = \begin{pmatrix} {}^wF_{St_i,x} \\ 0 \\ c_i \cdot \Delta h_i \end{pmatrix} \quad i = 2, 5 \quad (6.45)$$

$${}^w\mathbf{F}_{Achse} = \begin{pmatrix} 0 \\ {}^wF_{Achse,y} \\ 0 \end{pmatrix} \quad (6.46)$$

Mit den Kräften und Momenten können die Gleichgewichtsbedingungen für Kräfte und Momente aufgestellt werden. Als Bezugspunkt für das Momentengleichgewicht wird der Ursprung des Interalsystems gewählt.

Kräftegleichgewicht:

$$0 = \sum_{i=1}^6 {}^I\mathbf{F}_{St_i} + {}^I\mathbf{F}_{Achse} + {}^I\mathbf{F}_G + {}^I\mathbf{F}_{ZZ} + {}^I\mathbf{F}_{ZY} + {}^I\mathbf{F}_{ZX} \quad (6.47)$$

Momentengleichgewicht:

$$0 = \sum_{i=1}^6 {}^I\mathbf{r}_{St_i} \times {}^I\mathbf{F}_{St_i} + \\ + {}^I\mathbf{r}_{ZZ} \times {}^I\mathbf{F}_{ZZ} + {}^I\mathbf{r}_{ZY} \times {}^I\mathbf{F}_{ZY} + {}^I\mathbf{r}_{ZX} \times {}^I\mathbf{F}_{ZX} + {}^I\mathbf{r}_S \times {}^I\mathbf{F}_G \quad (6.48)$$

Kraft	Betrag in N
F_{St_1}	1025,5
F_{St_2}	5667,4
F_{St_3}	338,3
F_{St_4}	373,2
F_{St_5}	6159,0
F_{St_6}	1112,7
F_{Achse}	1290,4

Tabelle 6.4.: Beträge der Kräfte gerundet

Diese Gleichungen stellen im Vergleich zum dynamischen Modell keine Differentialgleichungen mehr dar. Es handelt sich um ein Gleichungssystem mit mehreren Variablen, das es zu lösen gilt. Dies stellt eine deutlich einfachere Aufgabe dar, als die Differentialgleichungen aus dem dynamischen Modell.

Ergebnisse

Microsoft Excel® (im Weiteren vereinfacht als Excel bezeichnet) kann, aufgrund der einfacheren Gleichungen, zur Lösung des Gleichungssystems herangezogen werden. Die Ergebnisse lassen sich dort anschaulich darstellen und es wird keine MATLAB-Lizenz benötigt um Berechnungen durchführen zu können.

In Tabelle 6.4 ist das Resultat der Berechnung dargestellt. Die Kräfte sind dabei als Betrag der Gesamtkraft angegeben. Die x-Komponenten im Wagenkoordinatensystem der Kräfte F_{St_2} und F_{St_5} sind in den jeweiligen Beträgen bereits berücksichtigt. Die Kraft in y-Richtung F_{Achse} wurde dagegen noch nicht in eine der beiden Kräfte eingerechnet und ist deshalb gesondert angegeben.

Mit Excel können die angreifenden Kräfte an ihren Angriffspunkten stilisiert dargestellt werden. In Abbildung 6.17 sind die Kräfte auf den als Fläche perspektivisch dargestellten Wohnwagen eingezeichnet. So ist auf einen Blick ersichtlich, welche Stützen hoch belastet sind und welche nur für die Stabilisierung des Wohnwagens sorgen.

6. Hauptstudie

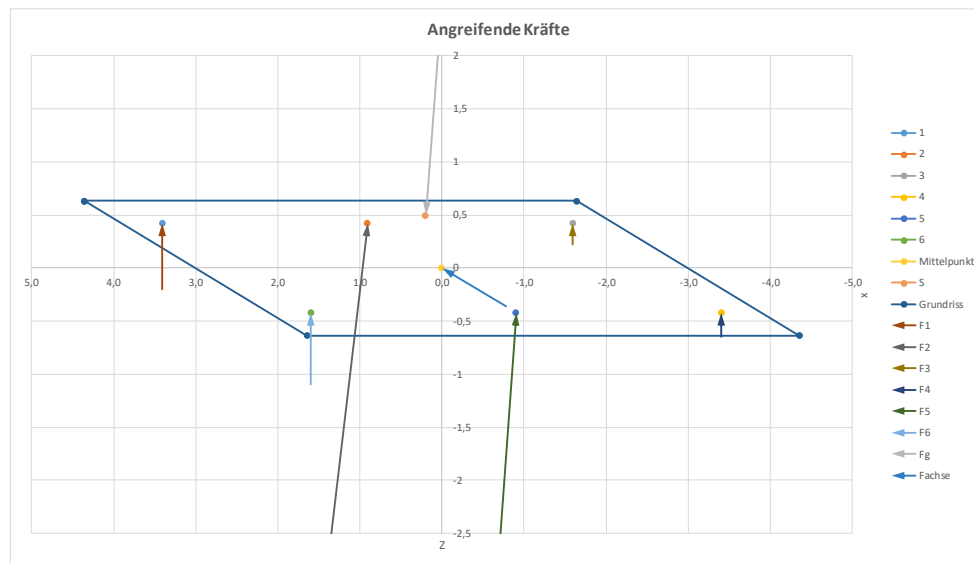


Abbildung 6.17.: Kräfte und Angriffspunkte

Das statische Modell bietet zusätzlich die Möglichkeit, die Verläufe der statischen Kräfte für eine Nick- oder Rollbewegung zu ermitteln. Dafür wird einer der beiden Winkel festgehalten und für den anderen die Kräfte an diskreten Punkten ermittelt. In den Abbildungen 6.18 und 6.19 sind solche Verläufe dargestellt. Beim Rollen ist der Nickwinkel auf 0° festgelegt und bei Nicken entsprechend der Rollwinkel auf 0° . Die Kräfte werden in Fünfgradschritten des variablen Winkels ermittelt. Die Diagramme zeigen gut, wie sich die Kraftsituation je nach Winkellage des Wohnwagens verändert. Beim Nicken und Rollen tragen die mittleren Stützen die Hauptlast. Bei der Nickbewegung kann die Verlagerung der Belastung von den hinteren auf die vorderen Stützen betrachtet werden. Durch die Lage des Schwerpunktes wird beim negativen Nickwinkel die maximale Belastung der mittleren Stützen erreicht. Beim Rollen erkennt man die Verlagerung der Kräfte von rechts nach links. Bei einem negativen Rollwinkel erreichen die Kräfte eins bis drei ihr Maximum, während bei einem positiven Rollwinkel die Kräfte vier bis sechs ihr Maximum erreichen.

Das statische Modell erleichtert also die Auswertung, wenn dynamische Kräfte und das dynamische Bewegungsverhalten für die aktuelle Analyse nicht von Relevanz sind. Viele grundsätzliche Betrachtungen zur Kraftsituation können so einfacher

6. Hauptstudie

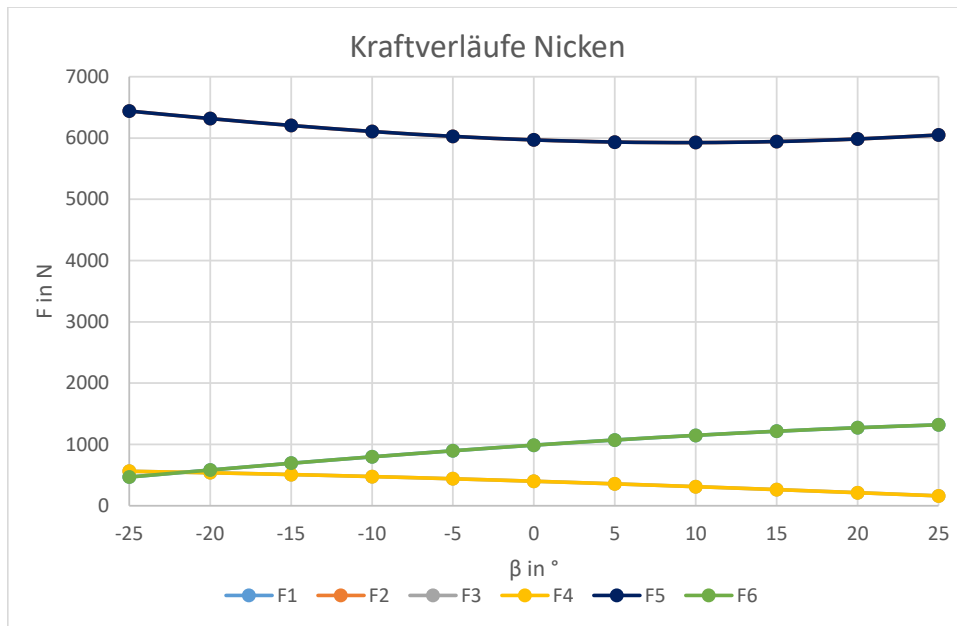


Abbildung 6.18.: Kräfteverläufe bei einer reinen Nickbewegung

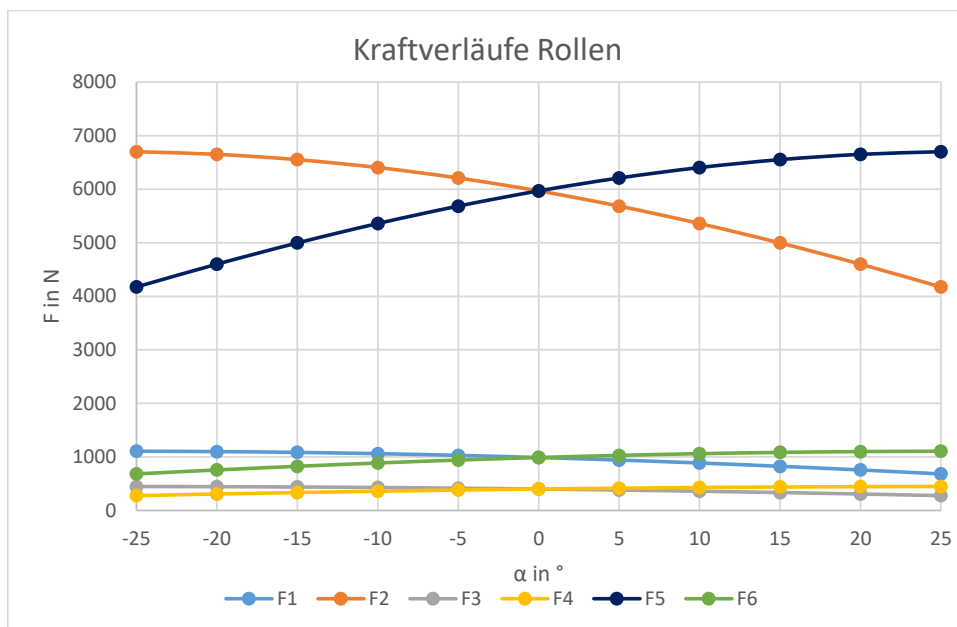


Abbildung 6.19.: Kräfteverläufe bei einer reinen Rollbewegung

und mit gängigerer Software durchgeführt werden. Ein weiterer Vorteil ist, dass die Massenträgheitsmomente hier keine Rolle spielen und nicht ermittelt werden müssen. Die wesentliche Größe für den Einfluss der Masse ist die Lage des Schwerpunkts, die leichter abgeschätzt werden kann, wie in Kapitel 6.1.1 gezeigt wurde. Eine Materialdämpfung spielt ebenfalls keine Rolle und muss somit nicht ermittelt werden. Nur die Nachgiebigkeit an den Stützstellen ist für dieses Modell ausschlaggebend. Nach Möglichkeit sollte also das statische Modell für Berechnungen herangezogen werden und das dynamische dann verwendet werden, wenn die dynamischen Zusammenhänge von Interesse sind.

6.1.4. Einzelstütze

Die Belastung der elektrischen Komponenten kann über den Aufbau der einzelnen Stütze errechnet werden. Die Kräfte, die von den Stützen getragen werden, sind durch das vorangegangene mechanische Modell bekannt. Allerdings müssen die Elektromotoren ein Drehmoment für den Antrieb der Bewegungsschrauben der Stützen liefern. Dieses Moment kann aus der Stützenbelastung errechnet werden, da es von der Geometrie der verbauten Stützen abhängt. Für die Berechnung werden die, am Prototyp verbauten, Stützen verwendet. Genaueres zu den Stützen ist in Kapitel 7 beschrieben ist. Ein dynamisches Modell ist bei der Stütze nicht erforderlich, da die Bewegung sehr langsam, quasi statisch, erfolgt.

Geometrie der Stütze

In Abbildung 6.20 ist die Geometrie der Stütze mit steifen Elementen modelliert. Die wesentlichen Längen sind abgebildet und mit einem Buchstaben als Symbol für das entsprechende Längenmaß versehen. Das Lager B ist ein Festlager, während das Lager A als Loslager dargestellt ist. Lager A ist aber nur insofern ein Loslager, als dass es durch die Bewegungsschraube, die hier nicht dargestellt ist, verschoben werden kann. Es wirkt also eine Kraft von der Bewegungsschraube auf das Lager A, sodass es in Position gehalten wird. Diese Kraft ist durch $F_{Spindel}$ dargestellt. Am Auflagepunkt zum Boden besteht ein Reibkontakt, der später im Kapitel aufgelöst

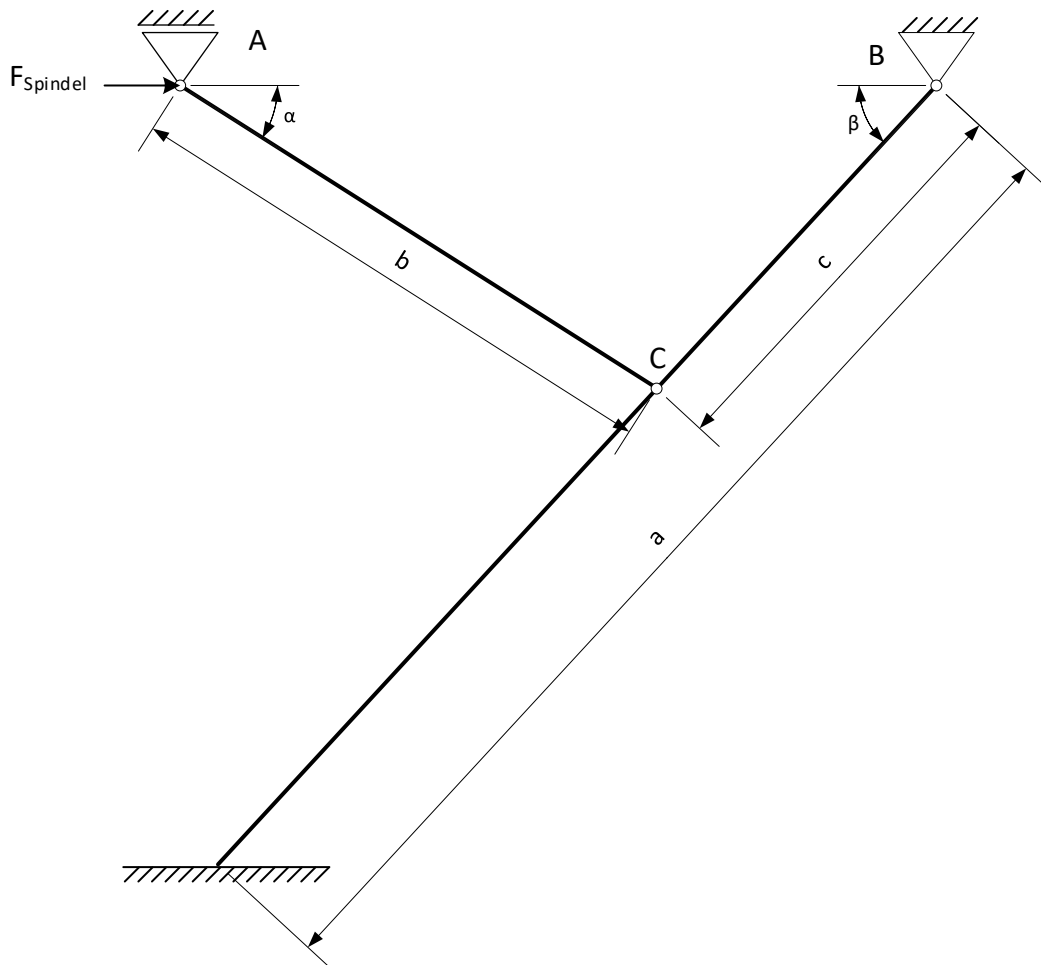


Abbildung 6.20.: Geometrie einer einzelnen Stütze

wird. Die eingezeichneten Winkel bestimmen bei gegebenen Längen, wie weit die Stütze ausgefahren ist.

Freischnitt

Für das statische Kräfte- und Momentengleichgewicht ist es notwendig einen Freischnitt der Stützelemente anzufertigen. Ein solcher Freischnitt wurde in Abbildung 6.21 durchgeführt. Die Reibungskräfte sind in der Gegenrichtung zur Ausfahrbewegung eingezeichnet, da die Hauptbelastung beim Ausfahren der Stütze auftritt.

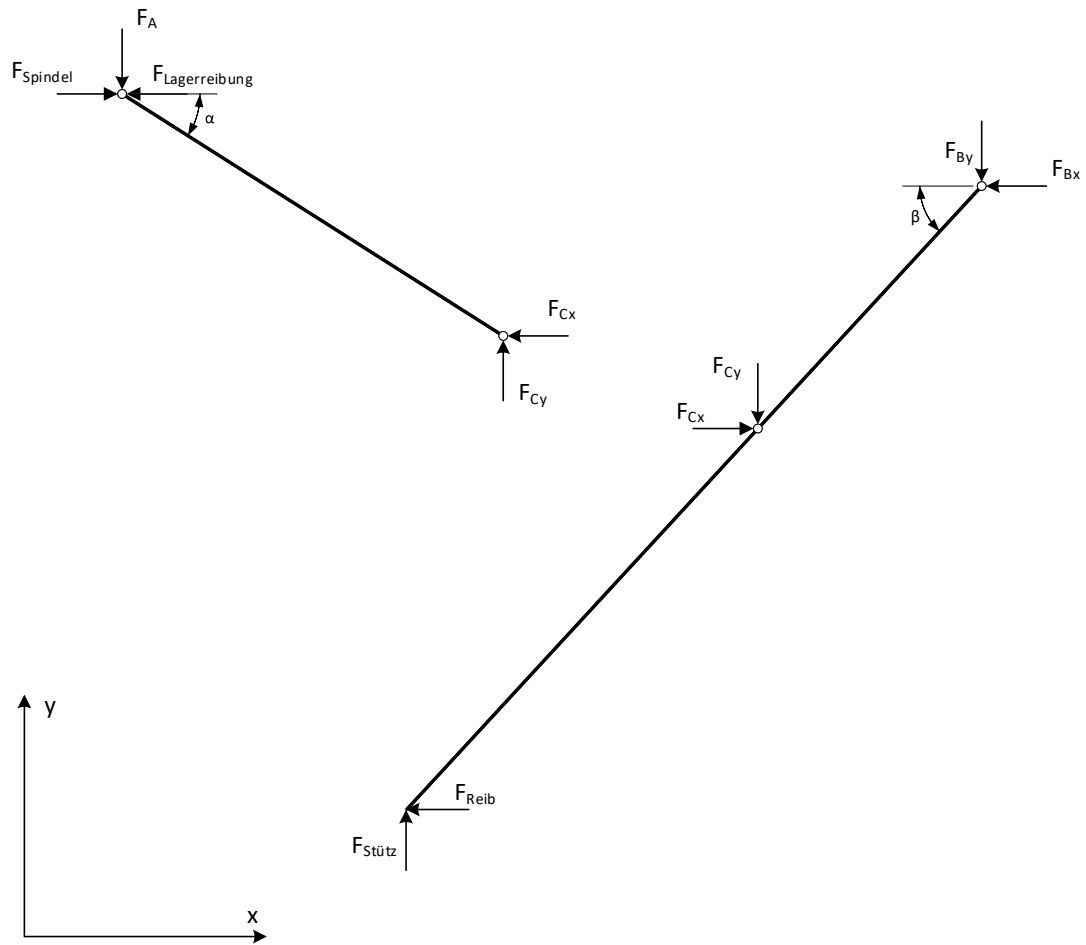


Abbildung 6.21.: Freischnitt einer einzelnen Stütze

Kräfte und Momente

Mit den Abmessungen und den Kräften aus dem Freischnitt lässt sich die Kraft $F_{Spindel}$ berechnen. Sie ist für das Ausfahren der Stütze verantwortlich. Findet keine Bewegung statt, hält die Spindelkraft die Stütze in Position. Für die Berechnung werden die Stützelemente A-C und B-C-Boden gesondert betrachtet. Die Verbindung zwischen den Elementen wird durch die Kraft im Lagerpunkt C hergestellt. Da die Berechnung zweidimensional stattfindet, können die Kräfte auf x- und y-Komponenten aufgeteilt werden. Diese Aufteilung wird durch einen entsprechenden Index gekennzeichnet.

Stützelement A-C Im Element A-C greifen die Kräfte nur an den äußeren Lagerpunkten A und C an. Diese Tatsache müssen die Richtungen der resultierenden Kräfte in den Lagerpunkten die gleiche sein, die Richtung des Stabes. Durch diese einfache Überlegung führt zu zwei Gleichungen für die Kräfte am Element A-C:

$$F_{Spindel} - F_{Lagerreibung} = F_{Cx} \quad (6.49)$$

$$F_A = F_{Cy} \quad (6.50)$$

Stützelement B-C-Boden Hier kommen die statischen Gleichgewichtsbedingungen zum Einsatz, wie bereits in den vorherigen Kapiteln angewendet. Für zweidimensionale Probleme formuliert:

$$\sum F_x = 0 \quad (6.51)$$

$$\sum F_y = 0 \quad (6.52)$$

$$\sum M^{(B)} = 0 \quad (6.53)$$

In die Gleichgewichtsbedingungen werden die Kräfte aus dem Freischnitt eingesetzt. Daraus folgt das Gleichungssystem:

$$0 = F_{Cx} - F_{Reib} - F_{Bx} \quad (6.54)$$

$$0 = F_{Stuetz} - F_{By} - F_{Cy} \quad (6.55)$$

$$0 = F_{Cx} \cdot c \cdot \sin \beta + F_{Cy} \cdot c \cdot \cos \beta - F_{Stuetz} \cdot a \cdot \cos \beta - F_{Reib} \cdot a \cdot \sin \beta \quad (6.56)$$

Reibungskräfte Die Reibungskräfte ergeben sich dabei aus der Normalkraft multipliziert mit dem jeweiligen Reibungskoeffizienten:

$$F_{Reib} \leq \mu_{Boden} \cdot F_{Stuetz} \quad (6.57)$$

$$F_{Lagereibng} \leq \mu_{Lager} \cdot F_A \quad (6.58)$$

Die Reibungskoeffizienten lassen sich für viele Materialpaarungen in der Fachliteratur nachschlagen oder sie können experimentell ermittelt werden. Vorsicht ist bei der Unterscheidung zwischen Haft- und Gleitreibung angebracht. Die angegebenen Gleichungen gelten nur für Haftreibung. Dabei kann die Reibungskraft kleiner oder gleich dem Produkt aus Haftreibungskoeffizienten mal Normalkraft sein. Bei der Gleitreibung ist die Reibungskraft gleich dem Produkt aus Gleitreibungskoeffizienten und Normalkraft auf die Reibungsfläche.

Spindelkraft Aus den formulierten Gleichungen können alle unbekanntes Größen eliminiert und eine explizite Gleichung für die Spindelkraft $F_{Spindel}$ aufgestellt werden.

$$F_{Spindel} = \frac{F_{Stuetz} \cdot a \cdot (\cos \beta + \mu_{Boden} \cdot \sin \beta) \cdot (1 + \mu_{Lager} \cdot \tan \alpha)}{c \cdot (\sin \beta + \cos \beta \cdot \tan \alpha)} \quad (6.59)$$

Die Spindelkraft $F_{Spindel}$ hängt somit nur mehr von der Länge der Stützelemente, den Reibungskoeffizienten, der Winkel α und β sowie der Kraft auf die Stütze F_{Stuetz} ab. Die vorhandene Stützkraft kann den Berechnungen des mechanischen Modells entnommen werden kann.

Antriebsmoment

Mit der Längskraft $F_{Spindel}$ in der Bewegungsschraube lässt sich das erforderliche Moment für den Antrieb ermitteln. Wittel u. a. (2015) geben eine entsprechende Formel für die Berechnung des Antriebsmoments an, die hier angewendet werden kann. Die Formel lautet:

$$M_{Antrieb} = F_{Spindel} \cdot \frac{d_2}{2} \cdot \tan(\varphi + \varrho') \quad (6.60)$$

Die verwendeten Symbole sind in Tabelle 6.5 beschrieben.

Symbol	Einheit	Bedeutung
$M_{Antrieb}$	Nm	Erforderliches Antriebsmoment
$F_{Spindel}$	N	Längskraft in der Bewegungsschraube
d_2	m	Flankendurchmesser des Gewindes
φ	rad	Steigungswinkel des Gewindes
ϱ'	rad	Gewinde-Gleitreibungswinkel

Tabelle 6.5.: Antriebsmoment - Symbole

Beispielhafte Berechnung

Eine Berechnung dieser einfachen Gleichungen lässt sich mit Excel durchführen. In Tabelle 6.6 sind die Eingaben und in Tabelle 6.7 die erhaltenen Ergebnisse aus Excel angeführt. Die Werte für die Eingaben entsprechen den Stützen des Prototyps. Die Stützenkraft ist von der Maximalbelastung einer Stütze abgeleitet. Als Reibungskoeffizienten sind Gleit- und Haftreibungskoeffizienten angegeben. Die Gleitreibungskoeffizienten werden verwendet, um das Betriebsmoment zu ermitteln, die Haftreibungskoeffizienten für das maximale Anfahrmoment. Der Winkel β wurde für die Berechnung beliebig angenommen, da er je nach Ausfahrzustand der Stütze variiert. In Kapitel 6.1.4 wird der Verlauf der Momente über den Winkel β ermittelt.

Verlauf der Momente über den Ausfahrzustand (Winkel β)

Die vorangegangene Berechnung betrachtet immer nur eine bestimmte Winkellage der Stütze. Die Momente verändern sich aber mit dem Winkel während des Ausfahrens der Stütze. Der Verlauf der Momente ist im Diagramm in Abbildung 6.22 dargestellt. Darin ist zu erkennen, dass eine starke Abhängigkeit der Antriebsmomente vom Ausfahrwinkel β besteht. Vor allem bei einer eingefahrenen Stütze müssen unter gleichbleibender Belastung größere Momente aufgebracht werden. Jedoch wird im Bereich von $\beta < 15^\circ$ keine Belastung aufgebracht, da zuerst der Bodenkontakt hergestellt werden muss. Bis dahin fährt die Stütze unbelastet aus. Ein wichtiger Winkel ist jener, bei dem der Bodenkontakt schließlich hergestellt wird. Dort tritt die höchste Belastung für den Antrieb auf.

6. Hauptstudie

Eingaben	
Geometrie	
a	550 mm
b	280 mm
c	280 mm
β	45°
Reibungskoeffizienten	
Haften am Boden $\mu_{BodenHaft}$	0.7
Gleiten am Boden $\mu_{BodenGleit}$	0.6
Haften im Loslager $\mu_{LagerHaft}$	0.3
Gleiten im Loslager $\mu_{LagerGleit}$	0.2
Stützenkraft	
F_{Stuetz}	589 N
Gewindedaten	
Flankendurchmesser d_2	19 mm
Gewindesteigung φ	3.8°
Gewinde-Haftreibungswinkel ϱ'_{Haft}	10°
Gewinde-Gleitreibungswinkel ϱ'_{Gleit}	6°

Tabelle 6.6.: Eingaben für die Antriebsmomentberechnung

Ergebnisse	
Spindelkraft	
beim Anfahren $F_{SpindelAnfahr}$	1287 N
im Betrieb $F_{SpindelBetrieb}$	1110 N
Antriebsmoment	
beim Anfahren $M_{AntriebAnfahr}$	2.98 N m
im Betrieb $M_{AntriebBetrieb}$	1.82 N m

Tabelle 6.7.: Ergebnisse Antriebsmomentberechnung

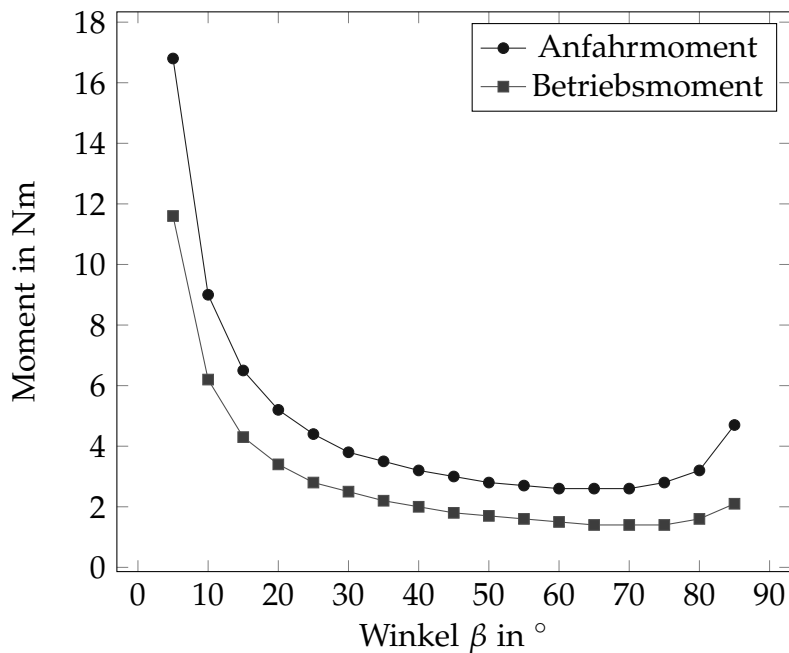


Abbildung 6.22.: Anfahrt- und Betriebsmoment einer Einzelstütze

6.2. Nivellierungsalgorithmus

Der Nivelliervorgang benötigt einen möglichst einfachen aber zugleich effizienten Algorithmus. Die Hauptaufgabe besteht in der Steuerung der Stützen, um eine nivellierte Lage zu erreichen. Als Versuchsobjekt ist bereits ein Prototyp vorhanden, der in Kapitel 7 näher vorgestellt wird. Dieser Prototyp besitzt vier Stützen die mit Elektromotoren betrieben werden. Da nur vier Stützen für mögliche Testläufe vorhanden sind, konzentriert sich die Entwicklung des Algorithmus ebenfalls auf das Ausfahren von vier Stützen. Mit den enthaltenen Grundprinzipien kann der entwickelte Algorithmus aber einfach auf sechs Stützen erweitert werden. An Stellen, die für zwei zusätzliche Stützen verändert werden müssen, wird auf die notwendige Veränderung eingegangen und diese kurz ausgeführt. Ein weiteres Merkmal des Algorithmus besteht darin, dass er hauptsächlich den Anforderungen an die Wohnmobilmivellierung gerecht wird. Der Aufbau wird als stabil genug angenommen, um das Eigengewicht mit den vier Stützen tragen zu können. Der Prototyp wird in den folgenden Absätzen als Wohnwagen bezeichnet, da dieser eigentlich als Modell für

einen Wohnwagen angedacht ist. Später wird noch eine weitere Variante des Algorithmus eingeführt, der mit vier Stützen nivelliert, aber den Hauptteil des Gewichts auf der Wohnwagenachse belässt und somit mehr der Wohnwagenvariante entspricht. Das führt zu einer eingeschränkten Funktionalität, da beispielsweise ein Rollvorgang mit dieser Randbedingung nicht möglich ist.

6.2.1. Algorithmusaufbau

Der Algorithmus kann grob in drei Phasen unterteilt werden, die analog zu den auszuführenden Aufgaben sind:

1. Herstellung von Bodenkontakt mit einer ausreichenden Anzahl von Stützen
2. Nivellierung des Wohnwagens oder Wohnmobils
3. Sicherstellen des Bodenkontakts der Stützen

In der ersten Phase gilt es, die aktuelle Lage des Wohnwagens festzustellen und den ersten Bodenkontakt mit den Stützen herzustellen. In der Regel genügen für den ersten Kontakt zwei Stützen, die den Bodenkontakt herstellen. Bei einer speziellen Lage kann es vorkommen, dass alle vier Stützen zum Boden gebracht werden müssen bevor eine Nivellierung starten kann. Dieses Szenario ist gegeben, wenn ein Ausheben der Achse des Wohnwagens unvermeidbar ist.

Die zweite Phase umfasst die Nivellierung selbst. Zweckmäßig ist es die Nickbewegung vor der Rollbewegung auszuführen. Das Nicken kann in der Regel mit dem Großteil des Gewichts auf der Achse erfolgen, außer wenn eine weitere Nivellierung nur mit einem Anheben der Achse möglich ist. Bei einer Rollbewegung muss hingegen immer zumindest ein Rad angehoben werden, da die Rollachse normal auf die Wagenachse steht.

Die dritte Phase stellt abschließend einen sicheren Stand des Wohnwagens her. Vor allem die Kraftsituation der Stützen, die im letzten Nivellierungsschritt nicht bewegt wurden, wird abschließend kontrolliert und ein sattes Aufliegen aller Stützen sichergestellt.

Diese drei Phasen können noch weiter in Zustände unterteilt werden. Befindet sich das System in einem bestimmten Zustand, ist eindeutig definiert welche Vorgänge ausgeführt werden müssen, um den nächsten Zustand zu erreichen. Das System durchläuft also bei einer Nivellierung verschiedene Zustände, bis der Wohnwagen in die gewünschte Endlage gebracht wurde. Zum Entwurf eines Algorithmus ist es hilfreich alle Status des Systems und deren Zusammenhänge zu identifizieren, beziehungsweise festzulegen. Eine Möglichkeit zur Veranschaulichung dieser Zustände und der Übergänge ist das Zustandsdiagramm, wie es in Kapitel 3.2.2 beschrieben wurde. Ein solches Diagramm ist für die Entwicklung des Algorithmus in Abbildung 6.23 dargestellt.

Aus dem Zustandsdiagramm ist ersichtlich, dass die drei Phasen unterschiedlich aufgebaut sind. Während die dritte Phase aus nur einem Zustand besteht (Bodenkontakt sicherstellen), sind in der zweiten Phase sechs Zustände enthalten. Diese sechs Zustände können zusätzlich mehrfach erreicht werden. In der ersten Phase ist hingegen nur ein aufeinander folgendes Durchlaufen der Zustände möglich, mehrfaches Erreichen ist ausgeschlossen. Bei unzulässigen Systemparametern kann jederzeit ein Abbruch des Algorithmus herbeigeführt werden. Ein spezieller Zustand ist „Alle Stützen zum Boden ausfahren“. Er ist fester Bestandteil der zweiten Phase, kann aber auch in der ersten Phase enthalten sein, je nach Lage des Wohnwagens. Die zweite Phase beginnt mit dem Zustand „Start der Nivellierung“, der über zwei Zustände erreicht werden kann und mit dem Erreichen des Zustandes: „Bodenkontakt sicherstellen“endet.

Beschreibung der Zustände:

Starte Algorithmus Hier ist der Einstiegspunkt in den Algorithmus. In diesem Zustand werden interne Parameter kontrolliert, die feststellen, ob sich die Stützen in einem zulässigen Zustand befinden. Im Idealfall sind sie vollständig eingefahren oder haben bereits eine Nivellierung absolviert. Näheres dazu wird im Kapitel 6.2.2 erläutert. Zusätzlich wird die Ausgangslage vom Lagesensor abgefragt und abgespeichert.

Erste Stütze zum Boden ausfahren Nach der Kontrolle, ob bereits ein Bodenkontakt einer oder mehrerer Stützen besteht, werden alle Stützen, wenn kein Bodenkontakt besteht, in Bewegung versetzt. In regelmäßigen Abständen wird eine Kontrolle des Verlaufs der Stromaufnahme durchgeführt und so ein Bodenkon-

6. Hauptstudie

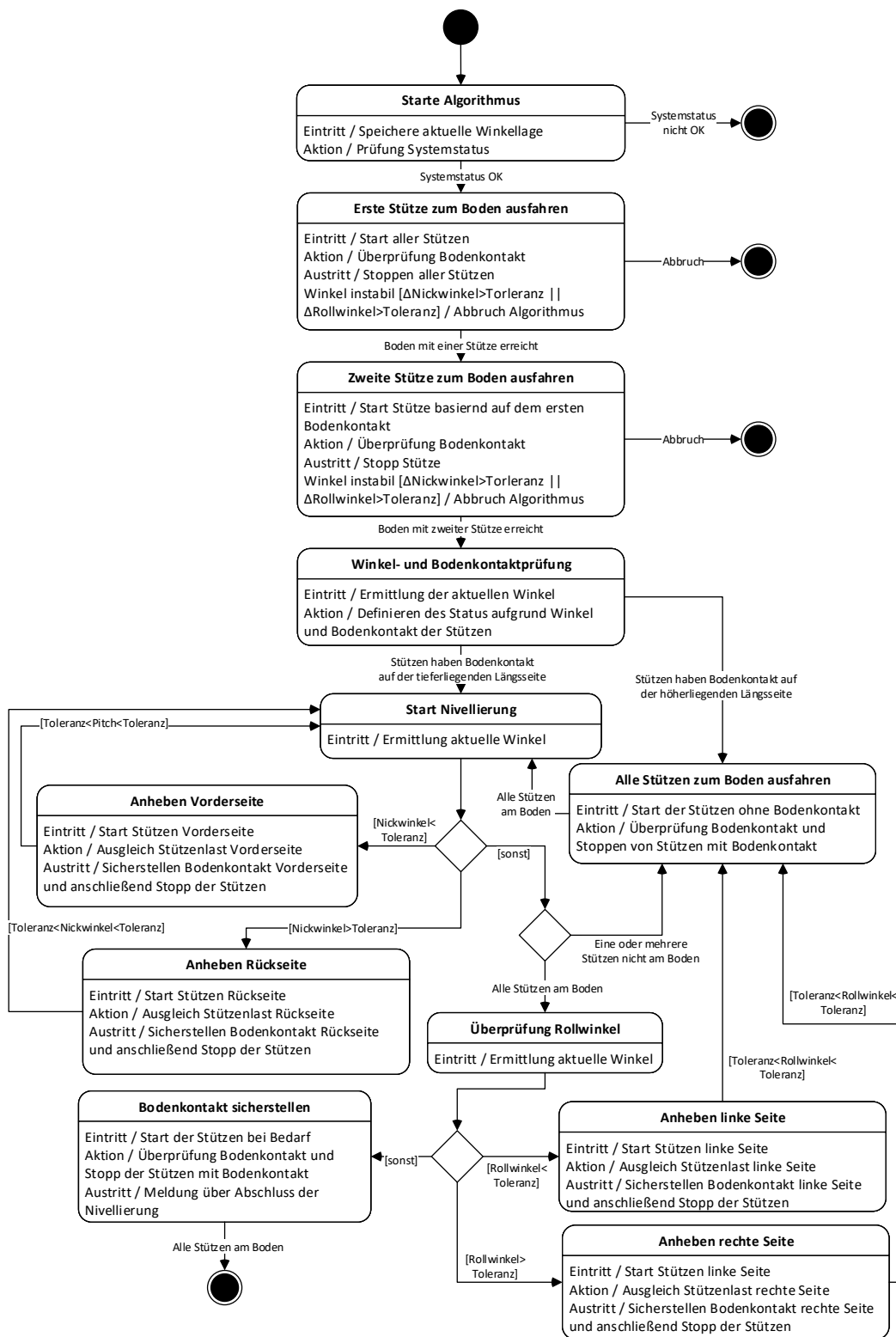


Abbildung 6.23.: Zustandsdiagramm Algorithmus

takt festgestellt. Die Methodik der Stromverlaufüberwachung ist im Kapitel 6.2.2 ausgeführt. Beim ersten Bodenkontakt einer Stütze werden alle Stützen gestoppt und der Übergang zum nächsten Zustand freigegeben. Während des gesamten Vorgangs wird die aktuelle Winkellage des Wohnwagens mit der gespeicherten Ausgangslage verglichen. Sollte es zu einer signifikanten Abweichung von der Ausgangslage kommen, wird der Algorithmus abgebrochen. Dies weist entweder auf einen internen Fehler oder auf eine unzulässige Verschiebung des Wohnwagens hin.

Zweite Stütze zum Boden ausfahren Bei Erreichen dieses Zustandes wird von einer ausreichend langen Bewegungszeit für das Festlegen der Leerlaufstromaufnahme ausgegangen. Die Vorhersagewerte werden festgelegt und für den weiteren Verlauf abgespeichert. Hauptaufgabe in diesem Zustand ist aber das Ausfahren der zweiten Stütze, die auf der gleichen Seite zur ersten Stütze mit Bodenkontakt liegt. Zusammengehörig sind dabei die Stützen eins und sechs sowie drei und vier. Die Stützen eins und sechs befinden sich in Fahrtrichtung vorne und die Stützen drei und vier entsprechend in Fahrtrichtung hinten. Die Stützen zwei und fünf sind bei der Vierstützenvariante nicht vorhanden. Bei Bodenkontakt der zweiten Stütze wird diese gestoppt und es findet der Übergang zum nächsten Zustand statt. Während des aktuellen Zustandes wird wieder die aktuelle Lage mit der Ausgangslage verglichen und bei einer unzulässigen Abweichung ein Abbruch des Algorithmus herbeigeführt.

Winkel- Bodenkontaktprüfung Die Lage des Wohnwagens wird mit den Bodenkontakten der Stützen verglichen. In diesem Zustand wird keine aktive Aktion mit den Stützen ausgeführt. Sind jene Stützen am Boden, die für die folgende Nickbewegung ausgefahren werden müssen, kann der Nivelliervorgang mit dem Zustand „Start Nivellierung“ gestartet werden. Ist das nicht der Fall, müssen alle Stützen über den Zustand „Alle Stützen zum Boden ausfahren“ ausgefahren werden.

Alle Stützen zum Boden ausfahren Der Bodenkontakt aller Stützen wird geprüft. Wenn sich eine Stütze nicht am Boden befindet, wird sie ausgefahren bis der Bodenkontakt hergestellt ist. Nachdem für alle Stützen der Bodenkontakt hergestellt wurde, findet der Übergang zum Zustand „Start Nivellierung“ statt.

Start Nivellierung Hier werden alle Daten zur aktuellen Winkellage und zum Systemzustand gesammelt, damit der richtige nachfolgende Zustand ausgewählt

werden kann. Bei einem zu kleinen Nickwinkel wird die Vorderseite, bei einem zu großen Nickwinkel wird die Rückseite des Wohnwagens angehoben. Das Anheben erfolgt entweder im Zustand „Anheben Vorderseite“ oder im Zustand „Anheben Rückseite“. Sollte sich der Winkel im Toleranzbereich für die Endlage befinden, wird wiederum der Bodenkontakt der einzelnen Stützen überprüft. Sollte eine Stütze nicht am Boden sein, wird in den „Alle Stützen zum Boden ausfahren“ Zustand übergegangen. Sind alle Stützen am Boden, kann mit der Überprüfung der Rolllage im Zustand „Überprüfung Rollwinkel“ fortgefahren werden.

Anheben Vorderseite Bei Eintritt in diesen Zustand werden die Stützen an der Vorderseite des Wohnwagens in Bewegung versetzt. Die Nickbewegung wird ausgeführt, bis sich der Nickwinkel im Toleranzbereich für die gewünschte Endlage des Wohnwagens befindet. Währenddessen wird die Last gleich auf die beiden bewegten Stützen verteilt. Zu diesem Zweck wird, wenn eine Stütze zu stark gegenüber der anderen belastet wird, diese angehalten, bis die andere Stütze wieder eine ausreichend große Last aufnimmt. Bevor der Übergang zurück zu „Start Nivellierung“ stattfindet, wird sichergestellt, dass die bewegten Stützen wieder Bodenkontakt haben.

Anheben Rückseite Dieser Zustand ist analog zum vorherigen aufgebaut, mit dem Unterschied das nun die Rückseite des Wohnwagens, unter gleichmäßiger Lastverteilung auf die beiden hinteren Stützen, angehoben wird. Ist der Nickwinkel im Toleranzbereich, geht der Zustand des Systems wieder in „Start Nivellierung“ über, wobei auf einen sicheren Bodenkontakt der beiden bewegten Stützen geachtet wird.

Überprüfung Rollwinkel Ist der Nickwinkel im Toleranzbereich und sind alle Stützen ausgefahren, wird der Rollwinkel überprüft. Sollte hier eine Abweichung außerhalb des Toleranzbereiches festgestellt werden, wird auf den entsprechenden Zustand zur Behebung dieser Abweichung übergegangen. Wenn keine Abweichung vorhanden ist, kommt der Algorithmus in die letzte Phase und zum Zustand „Bodenkontakt sicherstellen“.

Anheben linke Seite Die Vorgehensweise hier entspricht den beiden anderen anhebenden Zuständen. Der Folgezustand ist hier allerdings „Alle Stützen zum Boden ausfahren“, da sich beim Rollen möglicherweise eine Stütze vom Boden abgehoben haben könnte. Im nächsten Zustand erfolgt die wiederholte Prüfung

des Nickwinkels, da sich dieser beim Rollen verändert haben könnte.

Anheben rechte Seite Analog zur linken Seite.

Bodenkontakt sicherstellen Das ist der letzte Zustand, der den sicheren Stand des Wohnwagens gewährleisten muss. Sollten Stützen nicht fest am Boden stehen, werden sie hier noch einmal ausgefahren. Wenn dieser Zustand verlassen wird, ist der Algorithmus beendet.

Allgemeine Abbruchbedingung: Über den gesamten Algorithmus wird der Zustand der Stützen überwacht. Sollte eine der Stützen eine zu hohe Belastung aufweisen oder zu weit ausgefahren werden, wird der Algorithmus abgebrochen. Diese Bedingung ist im Zustandsdiagramm nicht dargestellt, da es sich hier um eine allgemeine Bedingung für alle Zustände handelt und eine Darstellung im Diagramm keinen Informationsgewinn gebracht hätte, sondern nur eine Fülle zusätzlicher Verbindungen und damit einen Verlust an Übersichtlichkeit. Die Funktionsweise dieser Abbruchbedingung wird in Kapitel 6.2.2 näher betrachtet.

6.2.2. Bestandteile des Algorithmus

Neben dem Aufbau des Algorithmus gibt es spezielle Bestandteile, die gesondert behandelt werden. Der Ablauf des Algorithmus an sich ist relativ einfach. Während der beschriebenen Zustände im vorherigen Kapitel werden Methoden aufgerufen, die bestimmte Aufgaben übernehmen. Die nichttrivialen Methoden und Bestandteile des Algorithmus werden im Folgenden aufgegriffen und erklärt.

Start des Algorithmus

Für den Algorithmus werden beim Ausfahren der Stützen die Leerlaufströme gemessen, da sie die Basis für die weiteren Methoden bilden. Damit eine ausreichend lange Zeit für die Messung vorhanden ist, sind Bedingungen für den Start des Algorithmus definiert. Der Algorithmus startet, wenn

- sich keine der Stützen am Boden befindet oder

Symbol	Bedeutung
y_t^*	Geglätteter Wert zum Zeitpunkt t
y_{t-1}^*	Geglätteter Wert zum Zeitpunkt $t - 1$
y_t	Wert zum Zeitpunkt t
α	Gewichtung $0 \leq \alpha \leq 1$

Tabelle 6.8.: Exponentielle Glättung - Symbole

- bereits Werte für den Leerlauf von einer vorangegangenen Nivellierung vorhanden sind.

Sollte keiner dieser Bedingungen erfüllt sein, müssen die Stützen vollständig eingefahren werden. Im Anschluss lässt sich der Nivelliervorgang starten.

Finden von Bodenkontakt

Ein zentrales Element im Algorithmus ist das Finden von Bodenkontakt. Durch die Anforderungen an das System ist die zur Verfügung stehende Sensorik sehr eingeschränkt. Zur Detektierung des Bodenkontakts steht nur eine Messung der Stromaufnahme der Elektromotoren zur Verfügung. Anhand des Stromverlaufs muss festgestellt werden, ob ein Bodenkontakt gegeben ist. Der Lagesensor liefert ein zusätzliches Feedback. Bewegt sich der Wohnwagen ohne Bodenkontakt einer Stütze, wird ein Fehler gemeldet und der Algorithmus abgebrochen.

Als Lösung für das Problem werden zwei mögliche Verfahren implementiert: die exponentielle Glättung und ein einfacher Mittelwert. Mit dem Verfahren der exponentiellen Glättung werden aktuelle Messwerte stärker gewichtet als vergangene. Wie stark die Gewichtung der aktuellen Messwerte ausfällt, kann mit einem Faktor angepasst werden. Das Verfahren verdeutlicht sich durch Betrachtung der zugehörigen mathematischen Formel:

$$y_t^* = \alpha \cdot y_t + (1 - \alpha)y_{t-1}^* \quad t \in \mathbb{N} \quad (6.61)$$

Die verwendeten Symbole sind in Tabelle 6.8 erklärt.

Symbol	Bedeutung
\bar{y}	Mittelwert
y_i	i -ter Messwert
n	Gesamtzahl der Messwerte

Tabelle 6.9.: Mittelwert - Symbole

Diese Formel liefert für jeden Zeitpunkt einen Schätzwert für y_{t+1} . Je nach Wahl von α wird der Schätzwert aus dem letzten Zeitschritt mehr oder weniger stark berücksichtigt. Bei einem Wert von $\alpha = 0$ verändert sich die Vorhersage gar nicht. Bei einem Wert von $\alpha = 1$ entspricht die Vorhersage dem aktuellen Wert.

Alternativ kann ein Mittelwert aller Messwerte zur Beurteilung des Bodenkontakts angewendet werden. Dieser folgt der bekannten Formel:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (6.62)$$

Die verwendeten Symbole sind in Tabelle 6.9 erklärt.

Als Veranschaulichung der beiden Methoden wird eine Messreihe des Prototyps (siehe Kapitel 7.4) herangezogen. Die Messwerte, sowie die Vorhersagen beider Methoden sind in Abbildung 6.24 dargestellt.

Für die tatsächliche Erkennung des Bodens wird ein oberer Grenzwert für die Abweichung des gemessenen Stroms von der Vorhersage festgelegt. Eine Überschreitung dieses Werts wird als Bodenkontakt gewertet. Der Grenzwert kann an die Gegebenheiten im verbauten System angepasst werden. Dabei ist ein Mittelweg zwischen einer fehlerhaften Erkennung und keiner Erkennung des Bodens zu finden.

Beide gezeigten Methoden liefern gute Ergebnisse bei der Bodenerkennung und sind im Algorithmus implementiert. Der Programmierer kann entscheiden, welche Methode tatsächlich angewendet wird. In einer späteren Entwicklungsstufe kann diese Entscheidung dem versierten Anwender übertragen werden. Für den Algorithmus des Prototyps wird die Methode der Mittelwertbildung angewendet.

Die Berechnung der Vorhersage für den Leerlaufstrom erfolgt, während die Stützen vom vollkommen eingefahrenen Zustand zum ersten Bodenkontakt ausfahren.

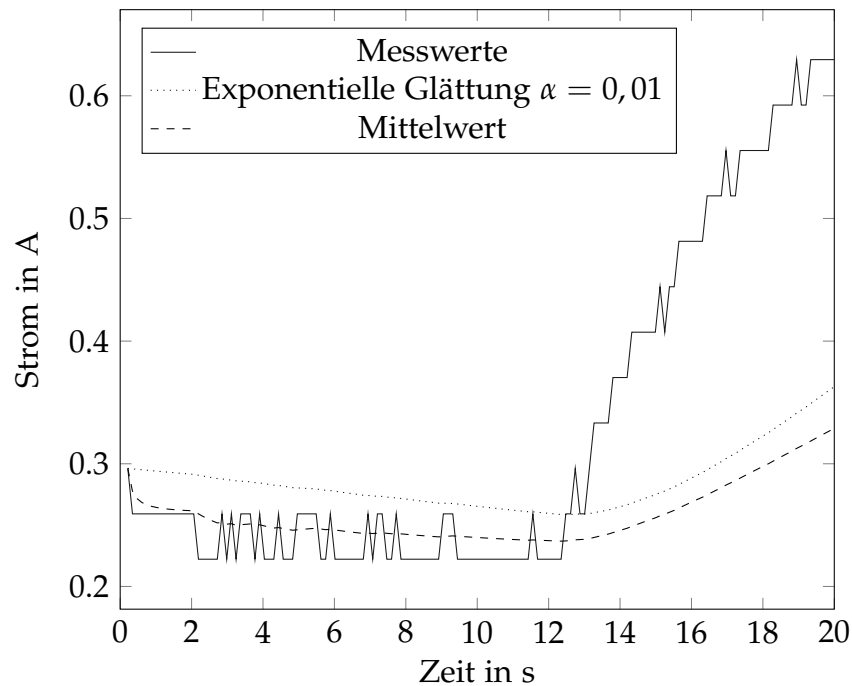


Abbildung 6.24.: Vergleich der Vorhersagemethoden

Nachdem dieser Punkt für die jeweilige Stütze erreicht ist, wird der Wert für die Vorhersage des Leerlaufstroms festgelegt. Sollte mit der gleichen Stütze ein zweites Mal der Boden gefunden werden müssen, wird der Messwert mit dem gespeicherten Leerlaufstrom überwacht. Der obere Grenzwert, mit dem die Kontrolle stattfindet, bildet sich aus dem Leerlaufstrom plus einer änderbaren Toleranz. Sollte der obere Grenzwert überschritten werden, wird die Stütze gestoppt. Dieses Vorgehen verhindert, dass die Vorhersage durch erhöhte Stromwerte innerhalb der oberen Grenze bei leichtem Bodenkontakt verfälscht wird.

Lastverteilung beim Nivellieren

Beim Nivellieren werden zum Ausgleich von Schräglagen des Wohnwagens zwei Stützen parallel auf der zu hebenden Seite bewegt. Verschiedene Abweichungen im mechanischen und elektrischen System führen zu einem ungleichmäßigen Ausfahren der verschiedenen Stützen. Für eine gleichmäßige Lastverteilung muss ein

Ausgleich der Lasten beziehungsweise der Geschwindigkeiten erfolgen. Im aktuellen Entwicklungsstand erfolgt die Steuerung der Stützen über ein Aktivieren und Deaktivieren des Motors, eine Geschwindigkeitseinstellung während des Betriebs ist nicht vorgesehen. Die Lastmessung erfolgt über eine Strommessung, die nur bei einer aktivierten Stütze durchgeführt werden kann. Zur Realisierung eines Ausgleichs muss eine überbelastete Stütze gestoppt werden und wieder gestartet werden, bevor die zweite fahrende Stütze überbelastet wird. Als Normalbereich, in dem beiden Stützen betrieben werden könne, wird eine zulässige Abweichung zwischen den gemessenen Strömen definiert. Damit eine im Leerlauf vorhandene Abweichung nicht das Ergebnis verfälscht, wird die Vorhersage des Leerlaufstroms, wie in Kapitel 6.2.2 berechnet, herangezogen. Als Referenzwert dient der Unterschied in den vorhergesagten Leerlaufströmen nach dem ersten Bodenkontakt einer Stütze. Dieser Wert wird bei der Ermittlung des Normalbereichs für den Betrieb berücksichtigt.

Sollte die Stromdifferenz vom Normalbereich abweichen, wird die höher belastete Stütze gestoppt. Die gestoppte Stütze bleibt solange im Stillstand bis die fahrende Stütze einen bestimmten Stromwert über den jeweiligen Schätzwert für den Leerlaufstrom erreicht hat. Die Methodik ähnelt dem Auffinden des Bodenkontakts, ohne Änderung der Vorhersage für den Leerlaufstrom, wobei jedoch ein anderer Grenzwert verwendet wird. Dadurch entsteht der geforderte Ausgleich der Last.

Bei Erreichen der gewünschten Lage des Wohnwagens kann der Fall eintreten, dass eine Stütze nicht in Bewegung ist. Das kann eine zu geringe Belastung der Stütze zur Folge haben. Um das zu verhindern wird, vor Beendigung dieses Teils des Algorithmus, der Bodenkontakt mit der in Kapitel 6.2.2 vorgestellten Methode sichergestellt.

Strommessung

Bei der Strommessung können Ausreißer innerhalb einer Messung auftreten, die das Messergebnis verfälschen und so beispielsweise eine Erkennung von Bodenkontakt auslösen, obwohl dies nicht der Fall ist. Mögliche Ausreißer müssen somit früh identifiziert und ausgeschieden werden. Dafür werden bei einer Messung des Stroms immer mehrere Messwerte aufgenommen und auf Ausreißer untersucht. Die Untersuchung der Messreihe erfolgt mit den statistischen Größen Mittelwert und

Symbol	Bedeutung
\bar{x}	Mittelwert
x_i	i -ter Messwert
s	Standardabweichung
n	Gesamtzahl der Messwerte

Tabelle 6.10.: Mittelwert und Standardabweichung - Symbole

Standardabweichung einer Stichprobe. Diese Werte errechnen sich entsprechend den Formeln:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (6.63)$$

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (6.64)$$

Die verwendeten Symbole sind in Tabelle 6.10 erklärt.

Ein Ausreißer liegt nun vor, wenn ein einzelner Messwert um ein bestimmtes Vielfaches der Standardabweichung vom Mittelwert abweicht. Die Grenze für diese Abweichung wird als D_{max} bezeichnet. Die Bedingung lässt sich mathematische formulieren als:

$$D_{max} \leq \left| \frac{x_i - \bar{x}}{s} \right| \quad (6.65)$$

Ist diese Bedingung erfüllt, wird der geprüfte Messwert als Ausreißer betrachtet und aus der Messreihe ausgeschieden. Als Messwert für den Strom wird der Mittelwert der übrigen Messwerte, die nicht ausgeschiedenen wurden, abgespeichert und im Algorithmus weiter verarbeitet.

Sicherheitsfunktionen

Durch die eingeschränkte Sensorik ist die Überwachung der Stützenbewegung mit Herausforderungen verbunden. Bei einem Fehlverhalten muss eine Zerstörung des Systems vermieden werden. Für das Ausfahren müssen sowohl die Ausfahrlänge als auch die Höchstbelastung überwacht werden. Beim Einfahren muss eine Stütze abschalten, wenn sie den vollkommen eingefahrenen Zustand erreicht hat.

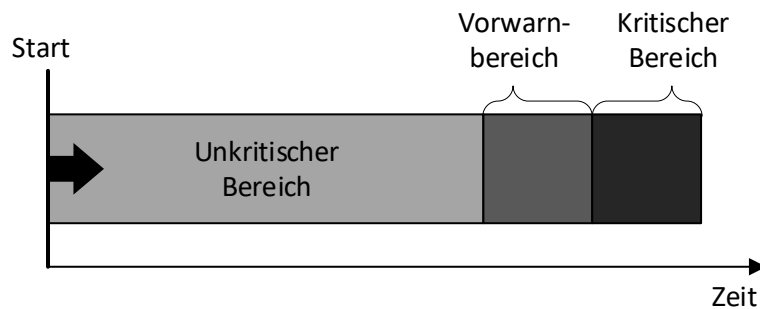


Abbildung 6.25.: Zeitbereiche für das Ausfahren einer Stütze

Die Überwachung der Maximalbelastung einer Stütze kann durch die Strommessung überwacht werden. Da ein direkter Zusammenhang zwischen Drehmoment und Strom besteht (siehe Kapitel 2.4.2), kann ein Grenzwert für den Strom festgelegt werden, der zugleich das Moment des Elektromotors begrenzt. Damit ist ebenfalls die Kraft auf die Stütze begrenzt. Die Stützkraft ist von der Geometrie der Stütze abhängig und vom deren Ausfahrzustand (siehe Kapitel 6.1.4). Ein fester Grenzwert nutzt somit die Maximalbelastung der Stütze nicht voll aus. Für den Prototypen genügt diese Art der Begrenzung. Möchte man näher an die Maximalbelastung kommen, muss der Grenzwert abhängig vom Ausfahrzustand der Stütze gemacht werden, entsprechend der Berechnung aus Kapitel 6.1.4.

Die Ausfahrlänge ist als direkter Messwert nicht vorhanden. Deshalb wird für die Beschränkung die Ausfahrzeit betrachtet. Der verbaute Elektromotor hat bei einer konstanten Spannung eine konstante Drehzahl, die unter Last nur geringfügig abfällt. Durch diese weitgehend konstante Drehzahl lassen sich Grenzwerte für die Ausfahrzeit und damit für den Ausfahrweg festlegen. Es wird eine Unterteilung in drei Bereiche festgelegt: Ein unkritischer Bereich, ein zulässiger Vorwarnbereich und ein kritischer Bereich (siehe Abbildung 6.25). Der unkritische Bereich kennzeichnet den Normalbetrieb. Bei einer Warnung kann der Nutzer über die Fortsetzung der Nivellierung entscheiden. Im kritischen Bereich erfolgt unter allen Umständen ein sofortiger Abbruch des Algorithmus. Im Vorwarnbereich kann dem Nutzer eine Warnung ausgegeben werden.

Eine weitere Sicherheitsfunktion, die bereits in Kapitel 6.2.1 genannt wurde, ist die Überprüfung der Stabilität bei der Suche nach Bodenkontakt. Insbesondere in der ersten Phase des Algorithmus wird der Wohnwagen von den Stützen nicht bewegt. Wird dennoch eine große Lageänderung festgestellt, muss der Algorithmus abgebrochen werden. Zur Feststellung von Abweichungen sind im Algorithmus Toleranzbereiche für Roll- und Nickwinkel definiert. Ist die Winkelabweichung von der Startlage größer, als der Toleranzbereich vorsieht, wird ein Fehler ausgelöst und die Nivellierung gestoppt. Wird der Wohnwagen während der Nivellierung von externen Kräften bewegt, kann keine sichere und korrekte Nivellierung gewährleistet werden.

7. Systemintegration

Zur Zusammenführung und Integration der Einzelkomponenten in ein Gesamtsystem wurde von der Firma Smart Lift ein Prototyp in Form eines Modells gebaut. Der Prototyp, siehe Abbildung 7.1, dient zur Veranschaulichen des Konzepts und ist eine sichere Möglichkeit verschiedene Algorithmen zu testen und auf Tauglichkeit zu prüfen. Der Prototyp ist im Vergleich zu seinem Gewicht sehr stabil. Bei einem Fehler im Algorithmus treten in der Regel keine schweren Schäden auf, anders als bei einem Testaufbau mit einem richtigen Wohnwagen.



Abbildung 7.1.: Prototyp der Firma Smart Lift

7.1. Aufbau und Komponenten

Der Prototyp ist aus Standardkomponenten aufgebaut, die in den nächsten Abschnitten genauer erläutert werden. Als Gesamtkonzept sind die Hauptabmessungen so konzipiert, dass ein Transport in einem Van möglich ist, womit eine größtmögliche Flexibilität beim Transport erreicht werden kann. Für das Grundkonstrukt wurden Holzplatten verwendet, die im Verhältnis zu den Belastungen am Prototyp sehr stabil sind. Schäden an der Mechanik sind hier nicht zu erwarten, auch nicht bei einer Fehlfunktion des Algorithmus.

7.1.1. Mechanische Komponenten

Die wesentlichen mechanischen Komponenten sind die vier Stützen. Diese sind bereits mit einem Elektromotor ausgestattet und haben somit einen Teil der elektrischen Komponenten mit eingebaut. Die Stützen wurden von der Firma AL-KO entwickelt und patentiert. Das EU-Patent hat die Nummer 1257 450 B1 (Beck und Hartung, 2005). Die Stützen werden in der Regel als Gesamtset, inklusive einer einfachen Steuerung zum Ein- und Ausfahren der Stützen, verkauft. Für den Prototyp werden die elektrischen Stützen ohne Steuerung verbaut, da die Steuerung durch den entwickelten Algorithmus erfolgt. Eine Stütze ist in Abbildung 7.2 dargestellt.

7.1.2. Elektrische Komponenten

Der Elektromotor ist, wie schon erwähnt, Teil der Stützen und ist dort bereits verbaut. Anhand des Typenschildes können die in Tabelle 7.1 aufgelisteten Daten für den Motor angegeben werden.

Die Ansteuerung der Motoren erfolgt mit zwei Doppel-H-Brücken. Jede H-Brücke kann jeweils zwei Motoren steuern. Die H-Brücken bilden die Schnittstelle zwischen Elektronik und Leistungselektronik. Eine der verbauten H-Brücken ist in Abbildung 7.3 dargestellt.

7. Systemintegration



Abbildung 7.2.: AL-KO E.C.S Elektronische Stütze (Alois Kober GmbH, 2009, S.31)

Antriebseinheit E.C.S.	
U_b	12 V DC
I_n	5.5 A
M_n	24 N m

Tabelle 7.1.: Motordaten

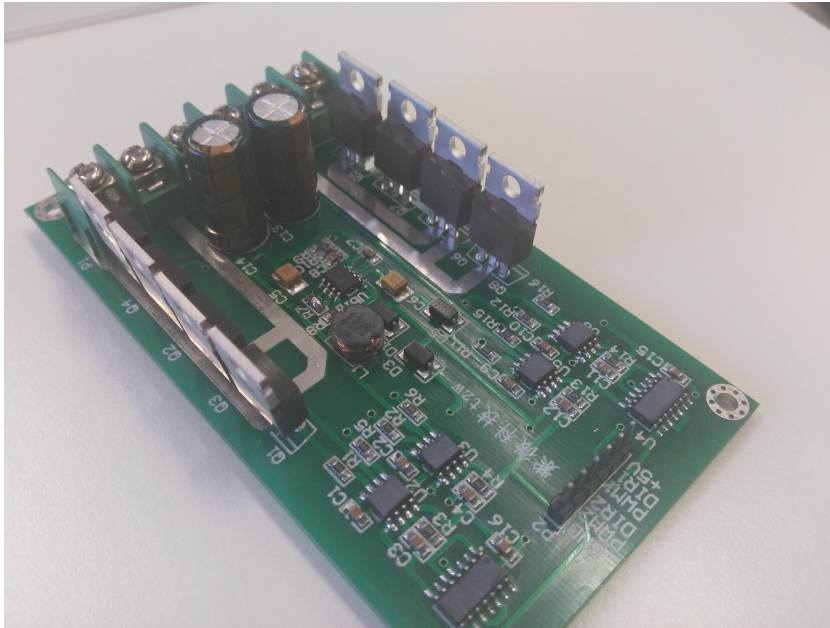


Abbildung 7.3.: H-Brücke

Zur Messung des Stroms werden Messbrücken verwendet, die als Hall-Sensor ausgeführt sind. Die Messung erfolgt mit einer seriellen Schaltung der Sensoren zwischen H-Brücke und Elektromotor. Je nach Stärke des Stromdurchflusses kann eine Messspannung zwischen 0 und 5 V abgenommen werden. Im Datenblatt des Sensors ist der Zusammenhang zwischen der gemessenen Spannung und dem Strom angegeben. Für den verbauten Sensor ist das: $\frac{U}{I} = 66 \text{ mV A}^{-1}$. Das Datenblatt ist im Anhang A.1 beigefügt.

Zur Bestimmung der Lage des Prototyps wird ein Beschleunigungs- und Gyrosensor vom Typ MPU6050 verwendet. Aus den Daten dieses Sensors lassen sich Nick- und Rollwinkel des Prototyps ableiten.

Die Energieversorgung erfolgt über einen handelsüblichen 12V-Bleiakkumulator, umgangssprachlich als Autobatterie bezeichnet.

7.1.3. Informationsverarbeitung

Zur Informationsverarbeitung kommt ein Arduino zum Einsatz, genauer gesagt handelt es sich um eine Arduino Mega 2560. Arduino ist ein Open-Source-Projekt, das die Umsetzung von Projekten mit vorgefertigten, günstigen Controllern ermöglicht. Bei dem Arduino Mega 2560 handelt es sich um einen Embedded Controller, der bereits viele nützliche Schnittstellen verbaut hat. Es können alle Sensoren und Aktoren einfach verbunden werden. Zur Kommunikation nach außen besitzt der Arduino serielle Schnittstellen, die eine Datenübertragung ermöglichen. Als Basisschnittstelle ist ein USB-Anschluss verbaut. Zusätzlich wird ein Bluetooth-Chip verwendet, da der Prototyp über ein Smartphone gesteuert werden soll. Der Chip kann über eine der seriellen Schnittstellen verbunden werden. Näher Informationen zum Arduino Mega können auf der Website www.arduino.cc (Arduino, 2017) eingeholt werden.

7.2. Benutzerschnittstelle

Als Endnutzerschnittstelle wird in einem parallelen Projekt, im Rahmen einer Bachelorarbeit, eine Applikation für Smartphones entwickelt. Die Oberfläche der Applikation ist in Abbildung 7.4 dargestellt.

Der Prototyp muss in geeigneter Weise mit der Applikation kommunizieren können, um deren Steuerbefehle umzusetzen und ein Feedback an den Nutzer zu liefern. Im Fall des verbauten Arduino Mega 2650 bieten sich, aufgrund der seriellen Kommunikation, Strings als Steuerbefehle an. Die Befehle sind aus mehreren Ebenen aufgebaut. Der aktuelle Satz von Steuerbefehlen umfasst die in Tabelle 7.2 angeführten Möglichkeiten. Die Befehle beinhalten die wichtigsten Funktionen für die Steuerung des Systems. Am Beginn und zwischen den Ebenen wird eine Raute zur Trennung der Befehle verwendet. Ein beispielhafter Befehl hat die Gestalt: „#CRTL#Pil_Stop#All“. Dieser Befehl bringt alle Stützen zum Stehen. Hier würde eine mögliche Nivellierung, die gerade in Gang ist, abgebrochen.

Ein zusätzlicher Befehl, der vom Smartphone bei Ausführung der Applikation gesendet wird, ist „#ALIVE“. „#ALIVE“ wird in regelmäßigen Abständen übertragen

Steuerungsbefehle			
Ebene 1	Ebene 2	Ebene 3	Bedeutung
CRTL	Do_Niv	-	Start der Nivellierung
CRTL	Pil_Ground	All; RF; LF; RB; LB	In Ebene 3 spezifizierte Stütze wird bis zum Boden ausgefahren
CRTL	Pil_Up	All; RF; LF; RB; LB	In Ebene 3 spezifizierte Stütze wird eingefahren
CRTL	Pil_Down	All; RF; LF; RB; LB	In Ebene 3 spezifizierte Stütze wird ausgefahren
CRTL	Pil_Stop	All; RF; LF; RB; LB	In Ebene 3 spezifizierte Stütze wird gestoppt.
CRTL	Offset	Set	Die aktuelle Lage wird als Ziellage für die Nivellierung gespeichert
CRTL	Offset	Zero	Die Ziellage wird auf den Standardwert zurückgesetzt
ALGO	1,2...10	-	Algorithmusauswahl

Tabelle 7.2.: Steuerbefehle

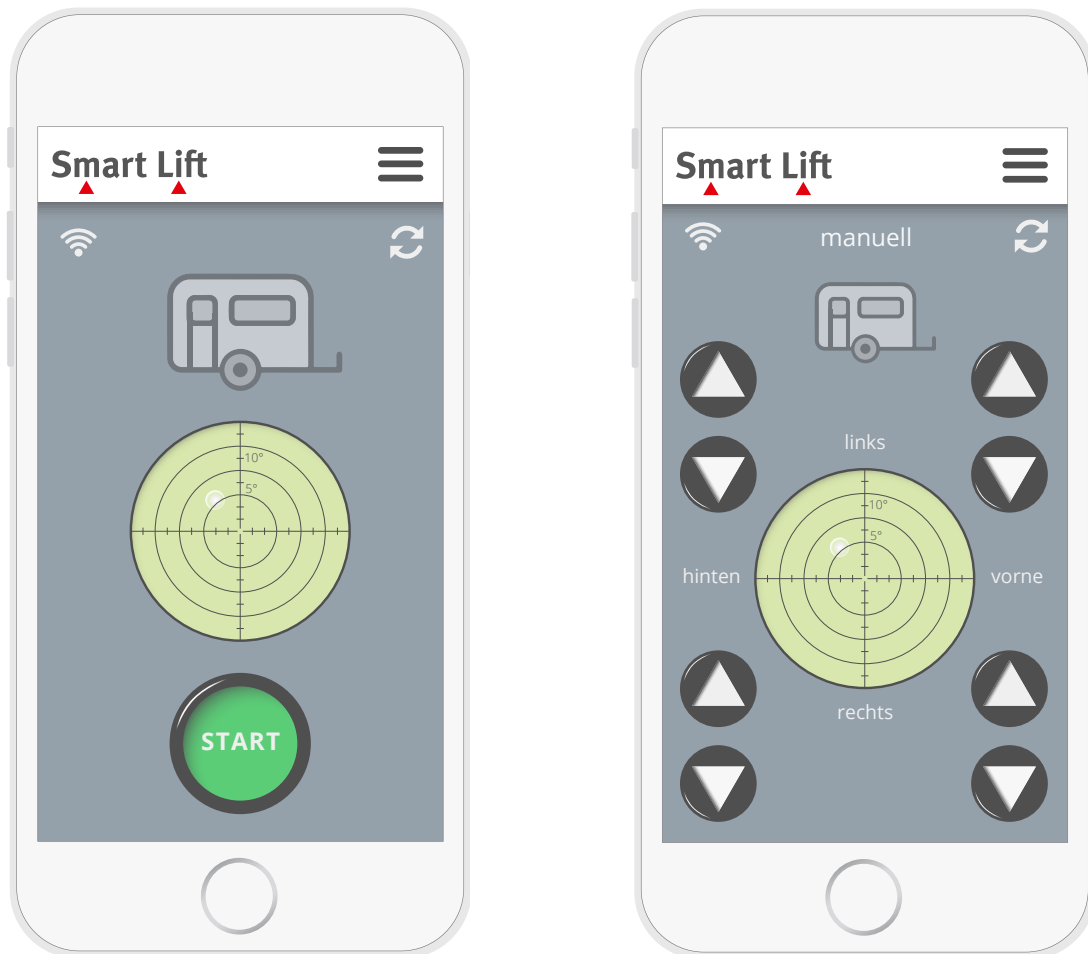


Abbildung 7.4.: Benutzerschnittstelle

und ermöglicht so die Überprüfung der Verbindung. Sollte nach einer einstellbaren Zeitspanne kein „#ALIVE“ empfangen werden, werden alle Stützen gestoppt und ein laufender Algorithmus ebenfalls abgebrochen.

Vom Mikrocontroller werden nicht nur Daten empfangen, sondern auch zum Smartphone gesendet. Dabei handelt es sich um die aktuelle Lage des Wohnwagens für die Darstellung am Display. Ein Datensatz umfasst Nick- und Rollwinkel in der Form: „#Pitch:1.1#Roll:2.2“. Der Datensatz wird in regelmäßigen Abständen gesendet.

7.3. Test und Bewertung von Algorithmen

Der Prototyp bietet die Möglichkeit Algorithmen unter definierten und abgesicherten Bedingungen zu testen. Bei schweren Fehlfunktionen sind die Schäden in der Regel minimal und es müssen nur einzelne Bauteile ersetzt werden. Somit ist das Testen des gesamten sowie von Teilen des Algorithmus sehr gut möglich. Da der Prototyp sehr vereinfacht ist, muss aber klar sein, dass nicht alle Aspekte eines Wohnwagens damit abgebildet werden können. Beispielsweise ist ein Ausheben der Achse kein Problem, bei einem Wohnwagen würde das zu schweren Schäden führen. Ein anderes Beispiel betrifft die Erkennung des Bodenkontakts. Beim Prototyp ist die Belastung der Stützen viel geringer, womit die Bodenerkennung schwieriger wird als bei einem Wohnwagen.

Die Bewertung der Algorithmen erfolgt vor allem vor dem Gesichtspunkt der Anforderungen, die in Kapitel 5.2 genauer behandelt wurden. Anforderungen, die unter die Kategorie „Muss“ fallen, sind vom Algorithmus, auch beim Prototyp, in jedem Fall zu erfüllen. Algorithmen, die das nicht leisten können, werden sofort ausgeschlossen oder sind soweit zu verbessern, dass diese Anforderungen eingehalten werden können. Als Kriterium für die weitere Bewertung sind die „Soll“-Anforderungen relevant. Je besser diese erfüllt werden, desto besser ist ein Algorithmus zu bewerten.

Übergeordnet steht das Thema der Sicherheit. Gefahren sollten möglichst bereits in der Entwicklung vermieden werden (siehe Kapitel 2.5). Algorithmen sind auf Sicherheit zu prüfen und mögliche Mängel und Gefahren systematisch zu reduzieren oder ganz zu eliminieren.

7.4. Messung der tatsächlichen Stromaufnahme

Damit eine Auswertung der tatsächlichen Stromaufnahme der vier Motoren möglich ist, kann eine Messung der Stromaufnahme durchgeführt werden. Bei dieser Messung werden alle Stützen ausgefahren und der gemessene Strom in Echtzeit über die USB-Schnittstelle des Arduino übertragen. Die Aufnahme und Speicherung der gesendeten

7. Systemintegration

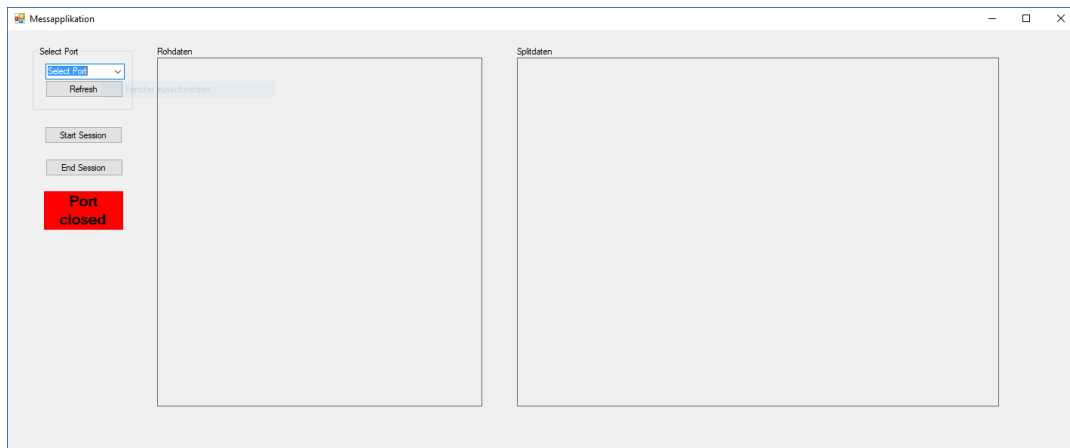


Abbildung 7.5.: Benutzerschnittstelle der Applikation zur Messdatenaufnahme

Daten erfolgt mittels einer eigens entwickelten Applikation. Die Benutzerschnittstelle dieser Applikation ist in Abbildung 7.5 dargestellt.

Die Applikation besteht aus den Grundfunktionen:

- Verbindung zum Arduino herstellen
- String der Messdaten aufnehmen
- Aufspalten des Strings zur Unterscheidung der Daten
- Ablegen der Daten in einer CSV-Datei

Die Applikation nimmt den String des Arduino auf und speichert die empfangenen Daten in geeigneter Weise für die weitere Verarbeitung ab. Durch das Speichern in einer Textdatei vom Typ CSV ist eine Auswertung mit verschiedenen Programmen möglich.

7.5. Validierung des mechanischen Modells

Das mechanische Modell liefert Daten zu den Kräften bei verschiedenen Lagen des Wohnwagens oder Wohnmobils. Das Modell kann mit den Parametern auf den Prototyp angepasst werden und liefert somit die Werte für die zu erwartenden Kräfte. Aus dem Modell müssten aber zu jeder erdenklichen Winkellage Kräfte errechnet werden, um sie anschließend mit den Messwerten zu vergleichen. Zur Vereinfachung

c_1	$10\,000\text{ N m}^{-1}$
c_3	$10\,000\text{ N m}^{-1}$
c_4	$10\,000\text{ N m}^{-1}$
c_6	$10\,000\text{ N m}^{-1}$
h_0	0.3 m
m_w	90 kg

Tabelle 7.3.: Parameter Prototyp

werden in 5° -Schritten für den Nick- und Rollwinkel die auftretenden Kräfte ermittelt und in je einer Matrix pro Stütze dargestellt. Für einen bestimmten Winkel können die Werte anschließend interpoliert werden.

$$\mathbf{F}_i(\alpha, \beta) = \begin{pmatrix} F_i(-25^\circ, -25^\circ) & F_i(-25^\circ, -20^\circ) & \cdots & F_i(-25^\circ, 25^\circ) \\ F_i(-20^\circ, -25^\circ) & F_i(-20^\circ, -20^\circ) & \cdots & F_i(-20^\circ, 25^\circ) \\ \vdots & \vdots & \ddots & \vdots \\ F_i(= 25^\circ, -25^\circ) & F_i(20^\circ, -20^\circ) & \cdots & F_i(25^\circ, 25^\circ) \end{pmatrix} \quad i = 1 \dots 6 \quad (7.1)$$

Beim Prototyp kann die Stromaufnahme der Stützen gemessen werden. Anhand dieser wird auf das aufgebrachte Drehmoment zurückgerechnet. Mit dem Moment lässt sich schließlich die auf den Stützen liegende Last berechnen. Das ist jene Last, die im mechanischen Modell bereits berechnet wurde. Die beiden Ergebnisse werden zum Vergleich gegenübergestellt.

Die Parameter für die Berechnung der Kräfte über das mechanische Modell sind in Tabelle 7.3 dargestellt. Diese umfassen die Steifigkeiten und das Gewicht des Prototyps. Die Dämpfungen werden nicht angegeben, da mit dem vereinfachten statischen Modell gerechnet wird.

Neben den Parametern ist die Position der Stützen für das mechanische Modell von Bedeutung. Der Prototyp hat einen sehr einfachen symmetrischen Aufbau. Die x- und y-Achse entsprechen den Symmetrieachsen. Die Abstände sind in Abbildung 7.6 dargestellt. Zusätzlich zu den Stützen wird die Position des Schwerpunktes benötigt,

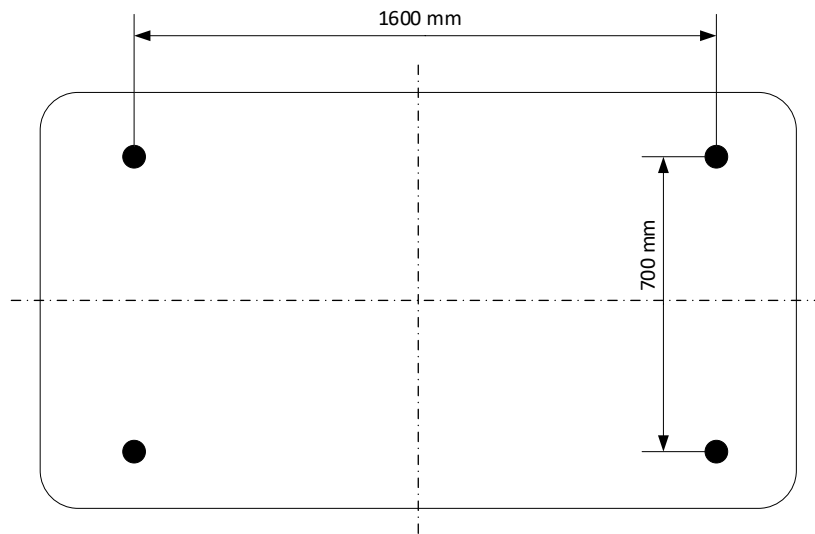


Abbildung 7.6.: Stützenposition Prototyp

die abgeschätzt wird als:

$$w r_S = \begin{pmatrix} 0.1 \text{ m} \\ 0 \text{ m} \\ 0.05 \text{ m} \end{pmatrix} \quad (7.2)$$

Die Kräfte, die auf die Stützen des Prototyps wirken, werden über die Messung der Stromaufnahme an den einzelnen Stützen errechnet. Eine sinnvolle Messung der Stromaufnahme ist aber nur möglich, wenn die Motoren eingeschaltet und in Bewegung sind. Die einfachste Möglichkeit, das umzusetzen, besteht im gleichzeitigen Ausfahren aller Stützen. Dabei wird der Strom gemessen, wie in Kapitel 7.4 beschrieben.

Zur Berechnung der Kraft aus der Stromaufnahme sind einige weitere Parameter nötig. Es wird wieder die Stützengeometrie aus Kapitel 6.1.4 herangezogen. Die dort eingeführten Variablen bleiben bestehen. Die Haftreibung wird in diesem Kontext nicht benötigt. Dafür ist der Ausfahrwinkel der Stützen in Abhängigkeit der Zeit zu berechnen. Die Drehzahl n_{min} der verbauten Motoren kann mit 22 Umdrehungen pro Minute angegeben werden. Damit kann berechnet werden, wie sich der Abstand der

oberen Lagerpunkte der Stütze verändert:

$$\Delta \overline{AB} = -n_{min} \cdot P \cdot t_{min} \quad (7.3)$$

P stellt dabei die Gewindesteigung der Bewegungsschraube und t_{min} die Zeit in Minuten dar. Damit kann die Länge \overline{AB} in Abhängigkeit von der Ausfahrzeit berechnet werden. Über die Stützengeometrie können die Winkel α_S und β_S der Stütze bestimmt werden.

$$\alpha_S = \arccos \frac{b^2 + \overline{AB}^2 - c^2}{2 * b * \overline{AB}} \quad (7.4)$$

$$\alpha_S = \arccos \frac{c^2 + \overline{AB}^2 - b^2}{2 * c * \overline{AB}} \quad (7.5)$$

Somit sind alle Größen zur Umrechnung der gemessenen Ströme in Kräfte bekannt. Die Ergebnisse aus dem mechanischen Modell und der Messungen können gegenübergestellt werden. Eine solche Gegenüberstellung ist in den nächsten vier Diagrammen durchgeführt. Die Bezeichnung der Stützen wurde beim Prototyp abgeändert¹. Die unterschiedliche Benennung ist in der getrennten Entwicklung von Prototyp und mechanischem Modell zu begründen. Es korrespondieren folgende Stützenbezeichnungen:

- 1 ↔ RB (right back)
- 3 ↔ RF (right front)
- 4 ↔ LF (left front)
- 6 ↔ LB (left back)

Damit die Koordinatensysteme und Winkel übereinstimmen entsprechen die vorderen Stützen im Prototyp den hinteren Stützen im Algorithmus. Somit liegt der Schwerpunkt ebenfalls in der, von Seiten des mechanischen Modells, erwarteten Lage.

Die Ergebnisse werden für jede Stütze in einem Diagramm dargestellt (Abbildungen 7.7 bis 7.10). Die darin zu erkennenden Abweichungen sind auf die begrenzte Möglichkeit zur Messung der Kräfte zurückzuführen. Die Messung findet während der Bewegung des Prototyps statt und erfolgt nur indirekt über den aufgenommenen

¹Die Bezeichnungen entsprechen der Lage der Stützen in Englisch.

7. Systemintegration

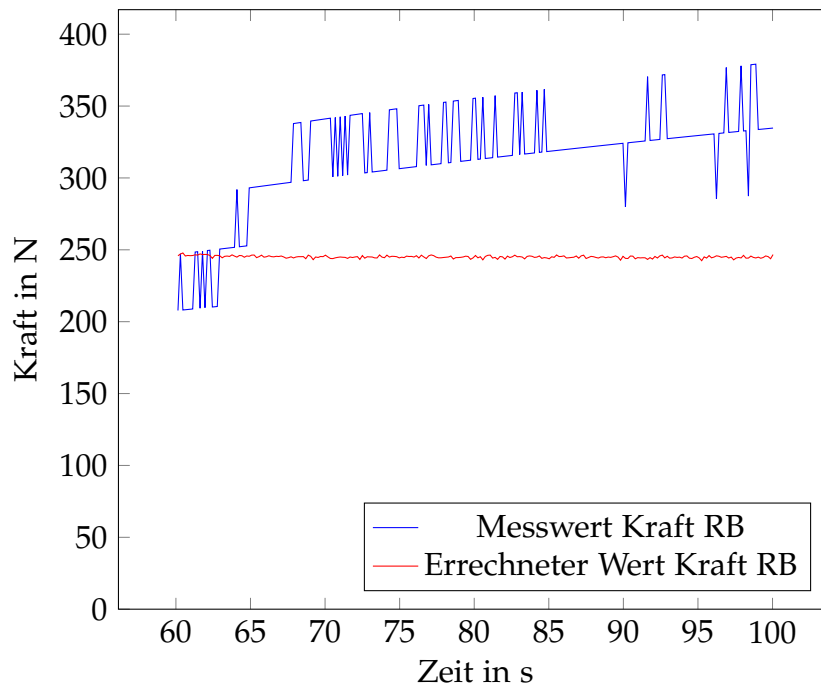


Abbildung 7.7.: Vergleich Kraft der Stütze RB

Strom. Eine weitere Abweichung liegt im nicht exakt definierten Startpunkt der Stützen. Dadurch sind manche Stützen weiter ausgefahren als andere und besitzen aus diesem Grund eine andere Belastung. Beispielsweise sind in dieser Messung die Stützen RB und LF höher belastet als errechnet, während die Stütze LB entlastet ist.

Die vorliegenden Messungen bieten also nur eine grobe Vergleichsmöglichkeit. Eine vollkommene Abweichung der errechneten Werte von den Messergebnissen kann aber ausgeschlossen werden. Genauere Untersuchungen erfordern eine Verbesserung der Messung.

7. Systemintegration

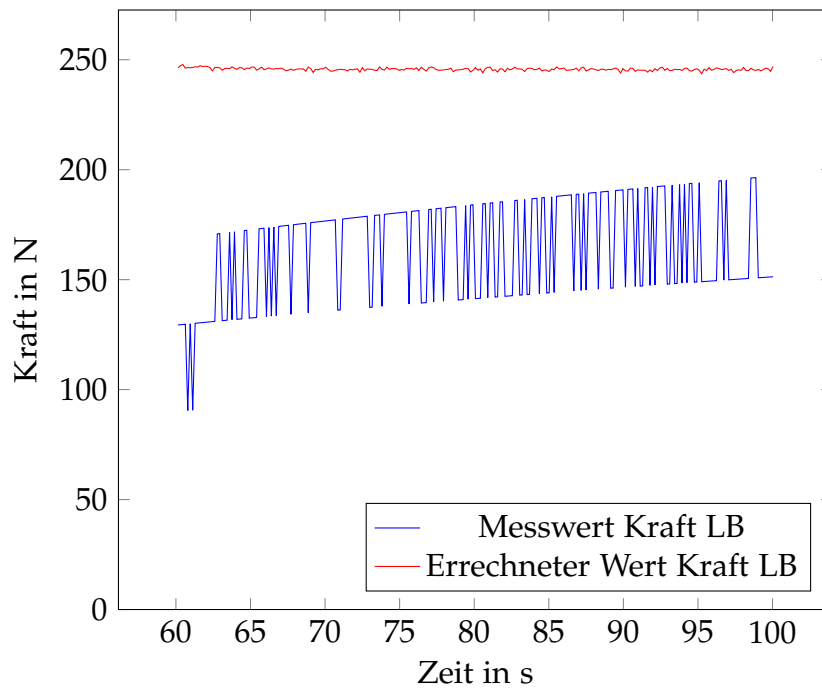


Abbildung 7.8.: Vergleich Kraft der Stütze LB

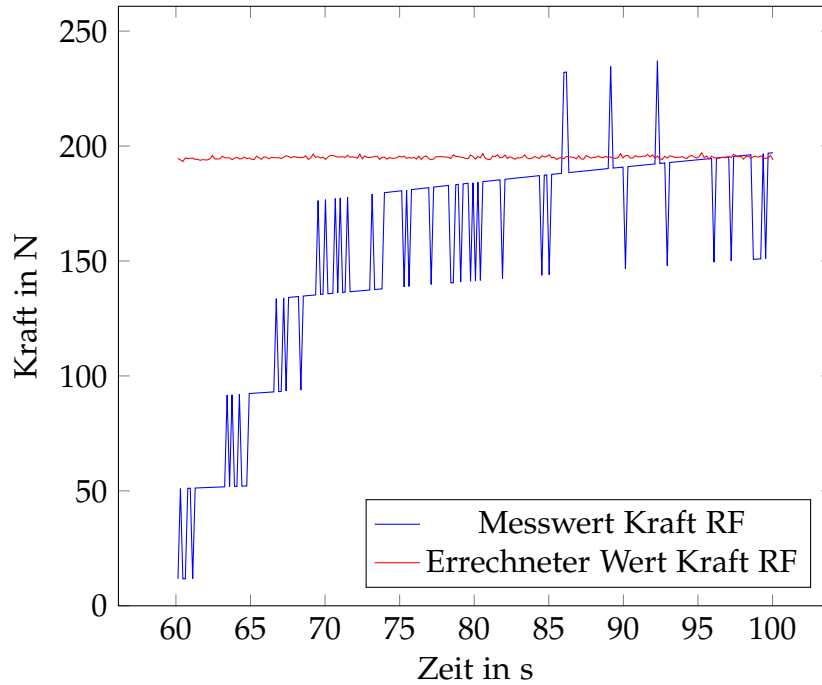


Abbildung 7.9.: Vergleich Kraft der Stütze RF

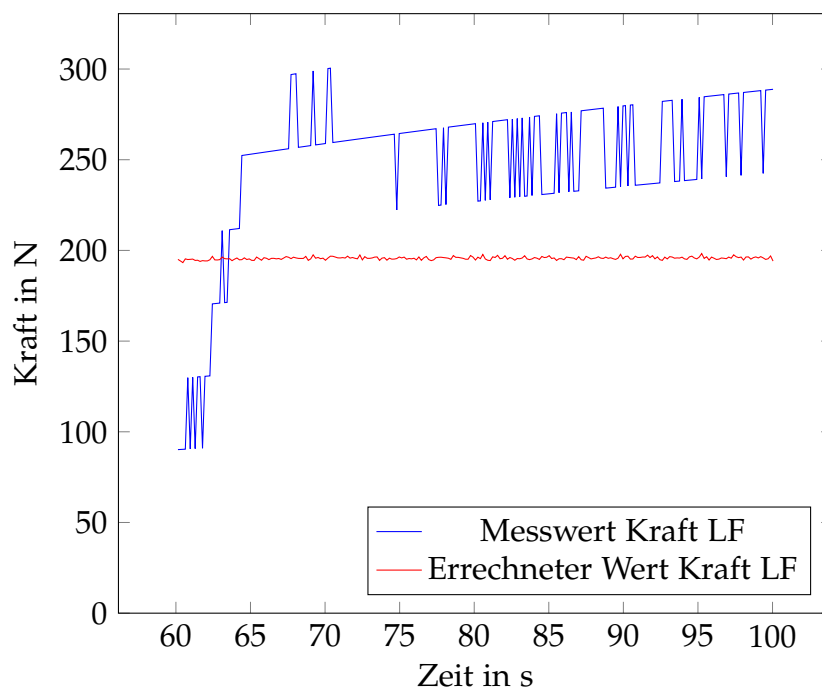


Abbildung 7.10.: Vergleich Kraft der Stütze LF

8. Resümee

Für die Entwicklung eines mechatronischen Systems ist eine systematische Vorgehensweise, wie sie von Haberfellner u. a. (2012) und vom VDI-Fachbereich Produktentwicklung und Mechatronik (2004) beschrieben wird, unabdingbar. Der Makrozyklus schafft ein geeignetes Vorgehen, das von einem Systementwurf über die Entwicklung der Teilsysteme hin zu einem integrierten System führt. Der Mikrozyklus zur Problemlösung findet in jeder Phase Anwendung. Mit diesen beiden Zyklen ist es möglich die Entwicklung eines mechatronischen Systems voranzutreiben und Irrwege zu vermeiden. Diese Vorgehensweisen haben in der gesamten Arbeit Eingang gefunden. Der Makrozyklus zieht sich über die gesamte Arbeit, beginnend mit der Vorstudie bis zur Systemintegration. Der Mikrozyklus steht hinter vielen Lösungen, die präsentiert wurden. Dieser ist weniger gut zu erkennen, da er nicht explizit erwähnt wurde und der Fokus auf der erarbeitenden Lösung liegt.

In der Vorstudie wurde mit der genauen Spezifizierung der Anforderungen der Grundstein für die folgende Entwicklung gelegt. Es gilt eine genaue Definition mit allen Stakeholdern durchzuführen, da mit fehlerhaften Anforderungen eine fehlerhafte und unerwünschte Entwicklung mit einhergeht. In dieser Phase des Projekts sollte ein großer Umfang an Energie aufgewendet werden. Wird darauf verzichtet, müssen Fehler und Abweichungen in späteren Projektphasen behandelt und verbessert werden. Die Kosten für das Projekt steigen damit unweigerlich.

In der Hauptphase wurde der systematische Entwicklungsprozess in den jeweiligen Teildisziplinen fortgeführt. Das mechanische Modell verschafft einen Überblick über die Situation der Kräfte bei einem Wohnwagen oder einem Wohnmobil. Durch eine Vielzahl an Parametern lässt es sich auf verschiedene Anwendungsfälle anpassen. Eine besondere Anpassung besteht in der Entwicklung eines statischen Modells. In manchen Fällen interessiert die statische Ruhelage und die dabei auftretenden Kräfte

im System. Eine Anpassung des Modells vermeidet, dass Bewegungen abgebildet werden, die in diesem Kontext nicht von Interesse sind. Der Berechnungsaufwand verringert sich dadurch erheblich und es können weniger spezialisierte Programme zur Auswertung eingesetzt werden.

Die Berechnung der Einzelstütze rundet das mechanische Modell ab. Bei einem mechatronischen System ist die Belastung der Motoren eine wichtige Größe. Sie kann über die Betrachtung einer einzelnen Stütze errechnet werden. Somit ist es einerseits möglich die erforderlichen Momente für eine Motorauslegung zu errechnen und andererseits durch Messung der Stromaufnahme der Motoren auf die Belastung der Stützen zu schließen.

Die gesammelten Informationen fließen in die Entwicklung eines Algorithmus zur automatischen Nivellierung ein. Die großen Herausforderungen für den Algorithmus bestehen in der minimalen Sensorik und der größtmöglichen Flexibilität. Durch die Anwendung an einem Fahrzeug ändern sich die Randbedingungen ständig. Der Untergrund, auf dem ein Wohnwagen oder Wohnmobil zu stehen kommt, kann von verschiedenster Art und Konsistenz sein. Feste Grenzwerte im Algorithmus sind nur mit Vorsicht verwendbar, beispielsweise zur Absicherung von physikalischen Grenzen, deren Überschreitung zu einer Beschädigung der Bauteile führen würde. Ein Bodenkontakt der Stützen muss aus dem Verlauf der gemessenen Größen abgeleitet werden.

Flexibilität bedeutet aber nicht nur dynamische Schwellwerte, sondern dass der Wohnwagen oder das Wohnmobil, auf dem das System zur Anwendung kommt, im Vorfeld nicht bekannt ist. Dennoch sollen die Anpassungen, die im Algorithmus notwendig sind, möglichst gering und beispielsweise mit einzelnen Parametern umsetzbar sein. So können verschieden Algorithmen angewählt werden, die aus den vorgestellten Zuständen bestehen. Für neue Anwendungsfälle, die zum aktuellen Zeitpunkt noch nicht bekannt sind, ist es möglich die Zustände selbst zu einem gewünschten Ablauf zusammen zu setzen. Dazu muss allerdings der Programmcode des Mikrocontrollers abgeändert werden, wozu Programmierkenntnisse notwendig sind.

Als letzter Teil erfolgte die Integration aller Bestandteile zu einem funktionierenden System, in diesem Fall zu einem Prototyp, der modellhaft einem Wohnwagen

nachempfunden ist. Der Vorteil liegt darin, dass er nur schwer beschädigt werden kann, im Vergleich zu einem tatsächlichen Wohnwagen. Sollten Umbauten notwendig sein, durch Beschädigung oder Umkonzeptionierung, ist das ebenfalls einfacher zu bewerkstelligen. Der Prototyp bietet die Möglichkeit eine Reihe verschiedenster Algorithmen und Vorgehensweise zu testen und so ein Optimum zu finden, das an einem Wohnwagen eingesetzt werden kann.

8.1. Ausblick

Vor dem Abschluss dieser Arbeit konnte bereits ein Test des Algorithmus an einem Wohnwagen erfolgreich absolviert werden. Der Algorithmus wurde dabei an die Bedürfnisse des Wohnwagens angepasst. Vorgabe war es, den Wohnwagen mit vier Stützen zu nivellieren und dabei aber auf eine Rollbewegung zu verzichten, da ein Rollen eine Beschädigung des Wohnwagens zur Folge hätte. Die geforderten Bewegungen konnten ohne großen Aufwand und mit wenigen Änderungen erfolgreich umgesetzt werden. Die Anpassungsfähigkeit und Funktionalität wurde also bereits erfolgreich getestet.

Längerfristig gesehen, soll das Nivellierungssystem zur Serienreife gebracht werden. Dafür müssen noch einige Anpassungen und Verbesserungen vorgenommen werden, die alle Teilbereiche des Systems betreffen. Die aktuellen Stützen sollen durch eine angepasste Variante ersetzt werden. Damit einhergehend findet die Auswahl eines neuen Motors statt, der besser auf die Anforderungen ausgelegt ist. Ein neuer Mikrocontroller soll entwickelt werden, der alle Funktionalitäten, die für das System notwendig sind auf einer Platine vereinigt. Das V-Modell als Makrozyklus muss also noch mehrmals durchlaufen werden, um zur Serienreife zu gelangen. Der Grundstein dafür wurde, mit der Entwicklung des Modells und des entsprechenden Algorithmus, sowie den positiven Tests am Prototypen, jedenfalls gelegt.

Literatur

- Alois Kober GmbH (2009). *HIGH-TECH-KOMPONENTEN FÜR CARAVANS. Anhängerkomponenten*. Hrsg. von Alois Kober GmbH. Kötz.
- Arduino (2017). *Homepage Arduino Projekt*. URL: www.arduino.cc (besucht am 23.08.2017).
- Beck, Gerhard und Rudolf Hartung (2005). »SUPPORTING DEVICE FOR VEHICLES«. EP1257450(B1) (Europa). KOBER AG und HINRICHS ELECTRONIC GMBH.
- Bolton, William (2004). *Bausteine mechatronischer Systeme*. 3. Aufl. München [u.a.]: Pearson Studium.
- Bumiller, Horst und Klaus Tkotz (2016). *Fachkunde Elektrotechnik*. 30. überarbeitete und erweiterte Auflage. Europa-Fachbuchreihe für elektrotechnische Berufe. 680 Seiten.
- Busch, Rudolf (2011). *Elektrotechnik und Elektronik. Für Maschinenbauer und Verfahrenstechniker ; mit 136 Übungsaufgaben mit Lösungen*. 6., erw. und überarb. Aufl. Studium. Wiesbaden: Vieweg + Teubner.
- Friedenthal, Sanford, Alan Moore und Rick Steiner (2012). *A practical guide to SysML. The systems modeling language*. 2. ed. Waltham. Mass.: Morgan Kaufmann.
- Geranmehr, Behdad, Kamran Vafae und Aliakbar Sadeqi, Hrsg. (2014). *High precision electro-hydraulic self leveling platform system*. Conference Proceedings. IEEE. 029-034.
- Haberfellner, Reinhard u. a. (2012). *Systems Engineering. Grundlagen und Anwendung*. 12., völlig neu bearb. und erw. Aufl. Zürich: Orell Füssli.
- Leveson, Nancy Gail (2011). *Engineering a safer world. Systems thinking applied to safety*. Engineering systems. Cambridge, Mass. [u.a.]: MIT Press.
- Magnus, Kurt und Hans Heinrich Müller-Slany (2005). *Grundlagen der Technischen Mechanik*. 7., durchges. und erg. Aufl. Leitfäden der angewandten Mathematik und Mechanik LAMM. Wiesbaden: Teubner.

- Rill, Georg und Thomas Schaeffer (2014). *Grundlagen und Methodik der Mehrkörpersimulation. Vertieft in Matlab-Beispielen, Übungen und Anwendungen*. 2., überarb. u. erw. Aufl. Lehrbuch. Wiesbaden: Springer Vieweg.
- Roddeck, Werner (2012). *Einführung in die Mechatronik*. Vieweg Teubner Verlag.
- Shampine, L. F. und M. W. Reichelt (1997). »The MATLAB ODE suite«. In: *SIAM JOURNAL ON SCIENTIFIC COMPUTING* 18 (1), S. 1–22.
- Tränkler, Hans-Rolf und Gerhard Fischerauer (2014). *Messtechnik*. Das Ingenieurwissen. Berlin u.a.: Springer Vieweg. IX, 96 S.
- VDI-Fachbereich Produktentwicklung und Mechatronik, Hrsg. (2004). *Entwicklungsmethodik für mechatronische Systeme*. VDI. Düsseldorf: VDI-Verl.
- Wittel, Herbert u. a. (2015). *Roloff/Matek Maschinenelemente. Normung, Berechnung, Gestaltung*. ger. 22., überarb. und erw. Aufl. Wiesbaden: Springer Vieweg. 50103 S.

Appendix

Anhang A.

Datenblätter

A.1. Datenblatt Hallsensor

Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor

Features and Benefits

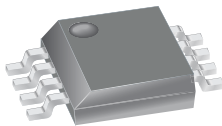
- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5 μ s output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at $T_A = 25^\circ\text{C}$
- Small footprint, low-profile SOIC8 package
- 1.2 m Ω internal conductor resistance
- 2.1 kV_{RMS} minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratiometric output from supply voltage



TÜV America
Certificate Number:
U8V 06 05 54214 010



Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1



Description

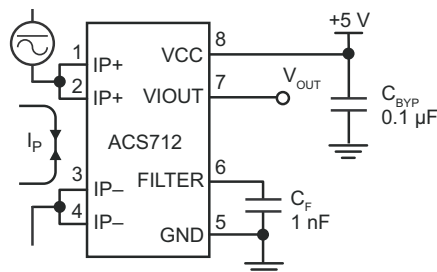
The Allegro[®] ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection.

The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope ($>V_{IOUT(Q)}$) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sensing. The internal resistance of this conductive path is 1.2 m Ω typical, providing low power

Continued on the next page...

Typical Application



Application 1. The ACS712 outputs an analog signal, V_{OUT} , that varies linearly with the uni- or bi-directional AC or DC primary sensed current, I_P , within the range specified. C_F is recommended for noise management, with values that depend on the application.

ACS712

Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor

Description (continued)

loss. The thickness of the copper conductor allows survival of the device at up to 5× overcurrent conditions. The terminals of the conductive path are electrically isolated from the sensor leads (pins 5 through 8). This allows the ACS712 current sensor to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The ACS712 is provided in a small, surface mount SOIC8 package. The leadframe is plated with 100% matte tin, which is compatible with standard lead (Pb) free printed circuit board assembly processes. Internally, the device is Pb-free, except for flip-chip high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.

Selection Guide

Part Number	Packing*	T _A (°C)	Optimized Range, I _p (A)	Sensitivity, Sens (Typ) (mV/A)
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	-40 to 85	±5	185
ACS712ELCTR-20A-T	Tape and reel, 3000 pieces/reel	-40 to 85	±20	100
ACS712ELCTR-30A-T	Tape and reel, 3000 pieces/reel	-40 to 85	±30	66

*Contact Allegro for additional packing options.

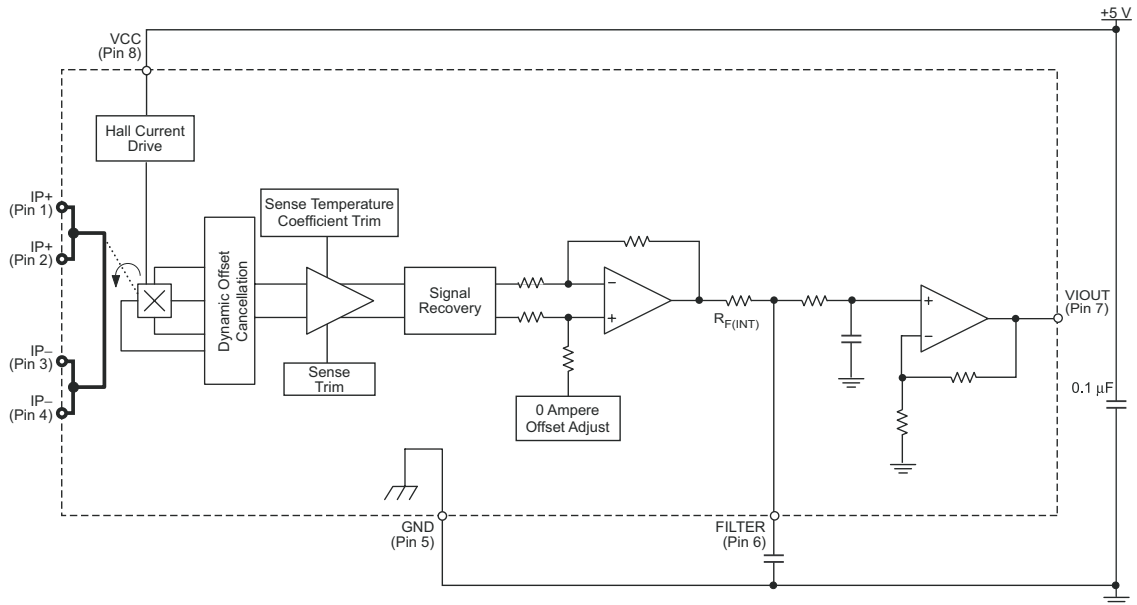
Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Supply Voltage	V _{CC}		8	V
Reverse Supply Voltage	V _{RCC}		-0.1	V
Output Voltage	V _{IOUT}		8	V
Reverse Output Voltage	V _{RIOUT}		-0.1	V
Reinforced Isolation Voltage	V _{ISO}	Pins 1-4 and 5-8; 60 Hz, 1 minute, T _A =25°C	2100	V
		Voltage applied to leadframe (I _p + pins), based on IEC 60950	184	V _{peak}
Basic Isolation Voltage	V _{ISO(bsc)}	Pins 1-4 and 5-8; 60 Hz, 1 minute, T _A =25°C	1500	V
		Voltage applied to leadframe (I _p + pins), based on IEC 60950	354	V _{peak}
Output Current Source	I _{IOUT(Source)}		3	mA
Output Current Sink	I _{IOUT(Sink)}		10	mA
Overcurrent Transient Tolerance	I _p	1 pulse, 100 ms	100	A
Nominal Operating Ambient Temperature	T _A	Range E	-40 to 85	°C
Maximum Junction Temperature	T _{J(max)}		165	°C
Storage Temperature	T _{stg}		-65 to 170	°C

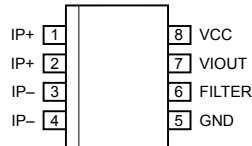
Parameter	Specification
Fire and Electric Shock	CAN/CSA-C22.2 No. 60950-1-03 UL 60950-1:2003 EN 60950-1:2001



Functional Block Diagram



Pin-out Diagram



Terminal List Table

Number	Name	Description
1 and 2	IP+	Terminals for current being sensed; fused internally
3 and 4	IP-	Terminals for current being sensed; fused internally
5	GND	Signal ground terminal
6	FILTER	Terminal for external capacitor that sets bandwidth
7	VIOUT	Analog output signal
8	VCC	Device power supply terminal

COMMON OPERATING CHARACTERISTICS¹ over full range of T_A , $C_F = 1$ nF, and $V_{CC} = 5$ V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
ELECTRICAL CHARACTERISTICS						
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Supply Current	I_{CC}	$V_{CC} = 5.0$ V, output open	–	10	13	mA
Output Capacitance Load	C_{LOAD}	VIOUT to GND	–	–	10	nF
Output Resistive Load	R_{LOAD}	VIOUT to GND	4.7	–	–	k Ω
Primary Conductor Resistance	$R_{PRIMARY}$	$T_A = 25^\circ\text{C}$	–	1.2	–	m Ω
Rise Time	t_r	$I_P = I_P(\text{max})$, $T_A = 25^\circ\text{C}$, $C_{OUT} = \text{open}$	–	5	–	μs
Frequency Bandwidth	f	–3 dB, $T_A = 25^\circ\text{C}$; I_P is 10 A peak-to-peak	–	80	–	kHz
Nonlinearity	E_{LIN}	Over full range of I_P	–	1.5	–	%
Symmetry	E_{SYM}	Over full range of I_P	98	100	102	%
Zero Current Output Voltage	$V_{IOUT(Q)}$	Bidirectional; $I_P = 0$ A, $T_A = 25^\circ\text{C}$	–	$V_{CC} \times 0.5$	–	V
Power-On Time	t_{PO}	Output reaches 90% of steady-state level, $T_J = 25^\circ\text{C}$, 20 A present on leadframe	–	35	–	μs
Magnetic Coupling ²			–	12	–	G/A
Internal Filter Resistance ³	$R_{F(INT)}$			1.7		k Ω

¹Device may be operated at higher primary current levels, I_P , and ambient, T_A , and internal leadframe temperatures, T_A , provided that the Maximum Junction Temperature, $T_J(\text{max})$, is not exceeded.

²1G = 0.1 mT.

³ $R_{F(INT)}$ forms an RC circuit via the FILTER pin.

COMMON THERMAL CHARACTERISTICS¹

			Min.	Typ.	Max.	Units
Operating Internal Leadframe Temperature	T_A	E range	–40	–	85	$^\circ\text{C}$
					Value	Units
Junction-to-Lead Thermal Resistance ²	$R_{\theta JL}$	Mounted on the Allegro ASEK 712 evaluation board			5	$^\circ\text{C/W}$
Junction-to-Ambient Thermal Resistance	$R_{\theta JA}$	Mounted on the Allegro 85-0322 evaluation board, includes the power consumed by the board			23	$^\circ\text{C/W}$

¹Additional thermal information is available on the Allegro website.

²The Allegro evaluation board has 1500 mm² of 2 oz. copper on each side, connected to pins 1 and 2, and to pins 3 and 4, with thermal vias connecting the layers. Performance values include the power consumed by the PCB. Further details on the board are available from the Frequently Asked Questions document on our website. Further information about board design and thermal performance also can be found in the Applications Information section of this datasheet.

ACS712

Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor

x05B PERFORMANCE CHARACTERISTICS $T_A = -40^{\circ}\text{C}$ to 85°C ¹, $C_F = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-5	-	5	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^{\circ}\text{C}$	180	185	190	mV/A
Noise	$V_{\text{NOISE(PP)}}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 185 mV/A programmed Sensitivity, $C_F = 47\text{ nF}$, $C_{\text{OUT}} = \text{open}$, 2 kHz bandwidth	-	21	-	mV
Zero Current Output Slope	$\Delta I_{\text{OUT(Q)}}$	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.26	-	mV/ $^{\circ}\text{C}$
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.08	-	mV/ $^{\circ}\text{C}$
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.054	-	mV/A/ $^{\circ}\text{C}$
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.008	-	mV/A/ $^{\circ}\text{C}$
Total Output Error ²	E_{TOT}	$I_P = \pm 5\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(\text{max})}$, is not exceeded.

²Percentage of I_P , with $I_P = 5\text{ A}$. Output filtered.

x20A PERFORMANCE CHARACTERISTICS $T_A = -40^{\circ}\text{C}$ to 85°C ¹, $C_F = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-20	-	20	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^{\circ}\text{C}$	96	100	104	mV/A
Noise	$V_{\text{NOISE(PP)}}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 100 mV/A programmed Sensitivity, $C_F = 47\text{ nF}$, $C_{\text{OUT}} = \text{open}$, 2 kHz bandwidth	-	11	-	mV
Zero Current Output Slope	$\Delta I_{\text{OUT(Q)}}$	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.34	-	mV/ $^{\circ}\text{C}$
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.07	-	mV/ $^{\circ}\text{C}$
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.017	-	mV/A/ $^{\circ}\text{C}$
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.004	-	mV/A/ $^{\circ}\text{C}$
Total Output Error ²	E_{TOT}	$I_P = \pm 20\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(\text{max})}$, is not exceeded.

²Percentage of I_P , with $I_P = 20\text{ A}$. Output filtered.

x30A PERFORMANCE CHARACTERISTICS $T_A = -40^{\circ}\text{C}$ to 85°C ¹, $C_F = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-30	-	30	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^{\circ}\text{C}$	64	66	68	mV/A
Noise	$V_{\text{NOISE(PP)}}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 66 mV/A programmed Sensitivity, $C_F = 47\text{ nF}$, $C_{\text{OUT}} = \text{open}$, 2 kHz bandwidth	-	7	-	mV
Zero Current Output Slope	$\Delta I_{\text{OUT(Q)}}$	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.35	-	mV/ $^{\circ}\text{C}$
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.08	-	mV/ $^{\circ}\text{C}$
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.007	-	mV/A/ $^{\circ}\text{C}$
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.002	-	mV/A/ $^{\circ}\text{C}$
Total Output Error ²	E_{TOT}	$I_P = \pm 30\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(\text{max})}$, is not exceeded.

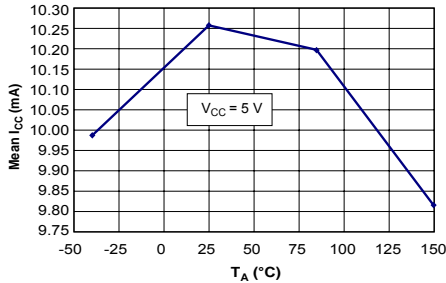
²Percentage of I_P , with $I_P = 30\text{ A}$. Output filtered.



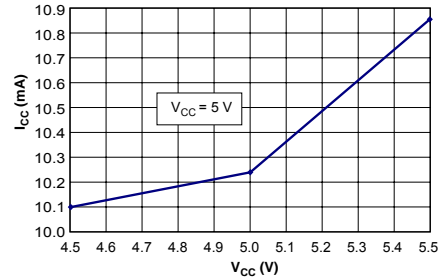
Characteristic Performance

$I_P = 5\text{ A}$, unless otherwise specified

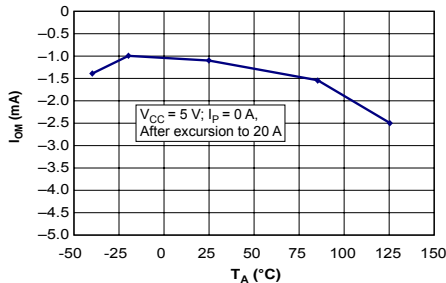
Mean Supply Current versus Ambient Temperature



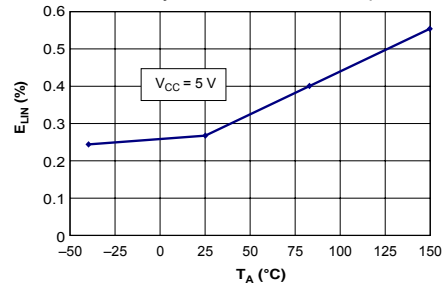
Supply Current versus Supply Voltage



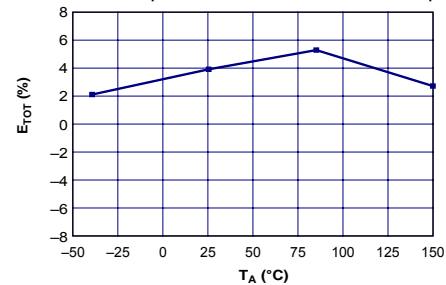
Magnetic Offset versus Ambient Temperature



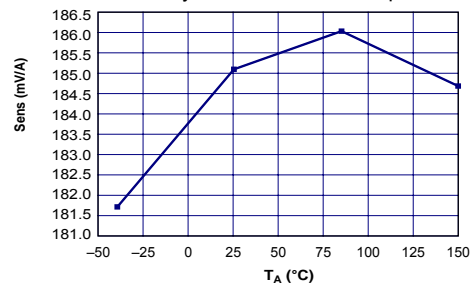
Nonlinearity versus Ambient Temperature



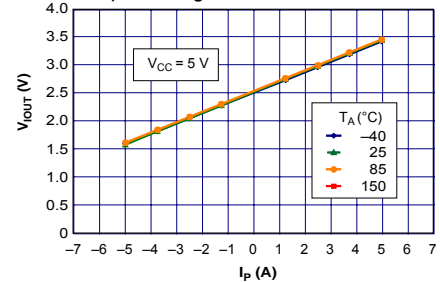
Mean Total Output Error versus Ambient Temperature



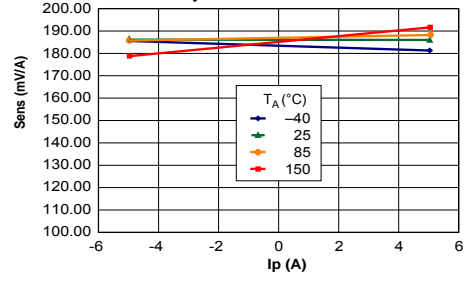
Sensitivity versus Ambient Temperature



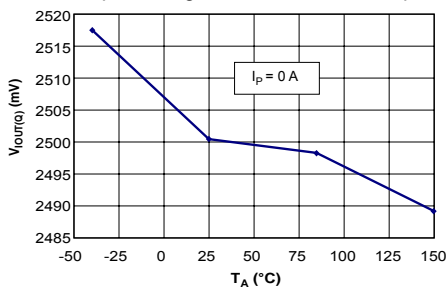
Output Voltage versus Sensed Current



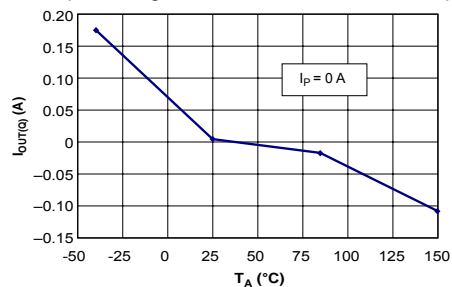
Sensitivity versus Sensed Current



0 A Output Voltage versus Ambient Temperature

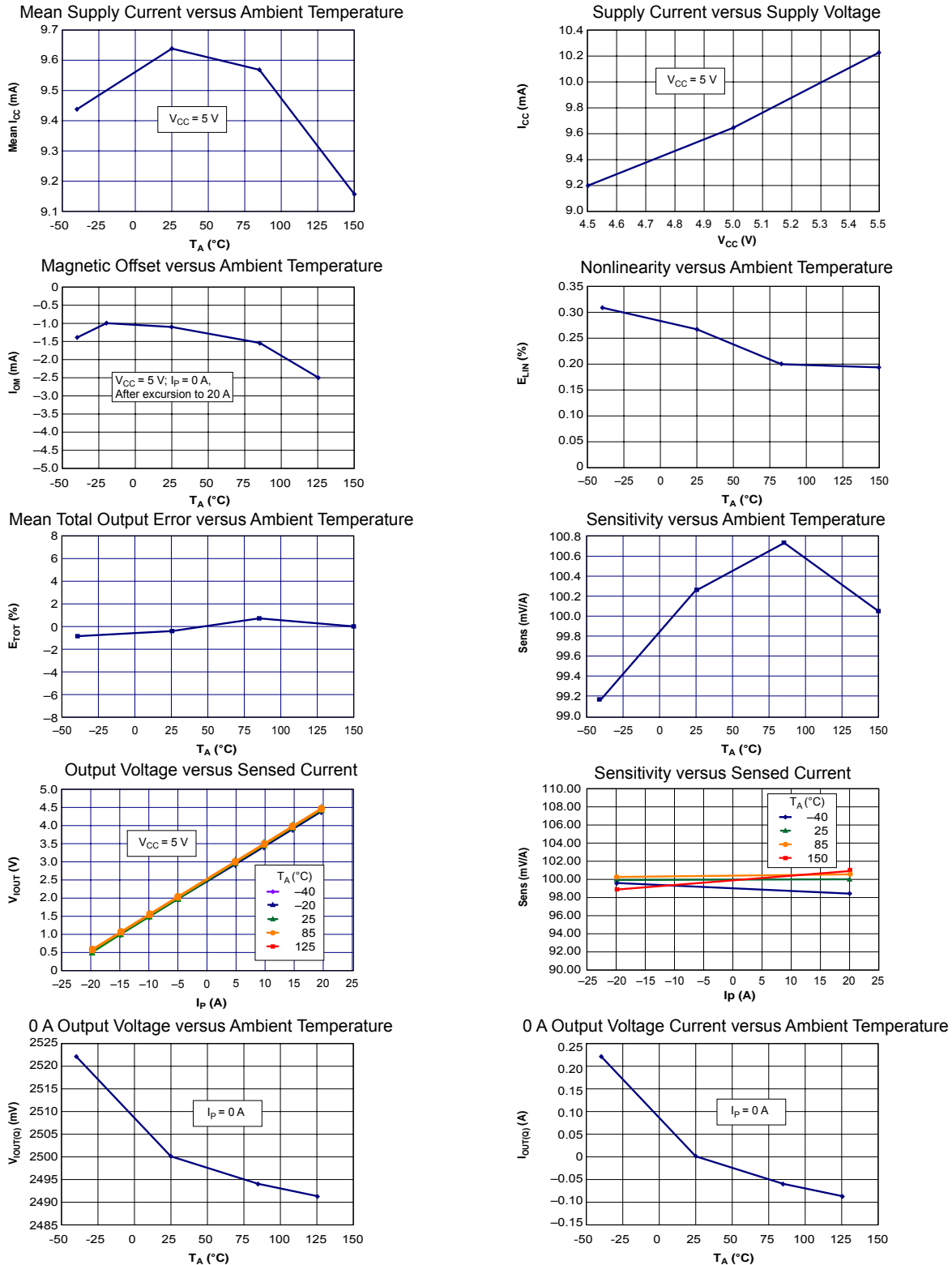


0 A Output Voltage Current versus Ambient Temperature



Characteristic Performance

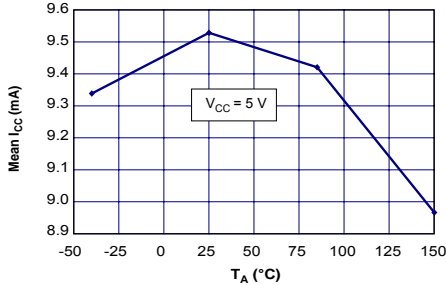
$I_P = 20\text{ A}$, unless otherwise specified



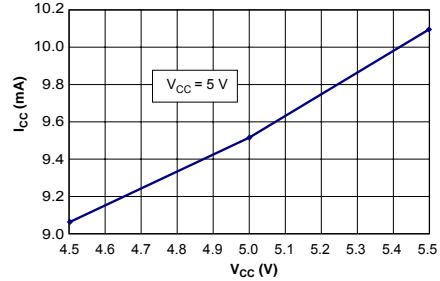
Characteristic Performance

$I_P = 30$ A, unless otherwise specified

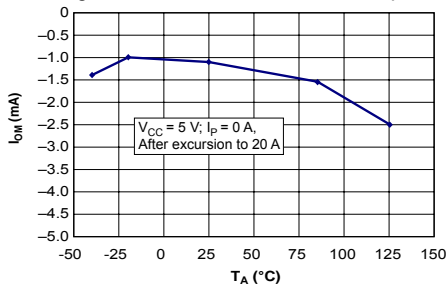
Mean Supply Current versus Ambient Temperature



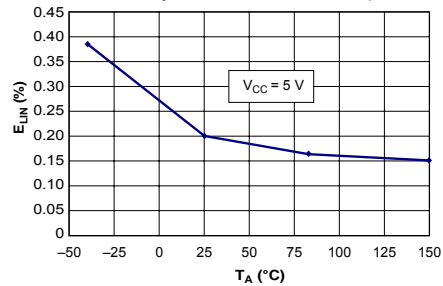
Supply Current versus Supply Voltage



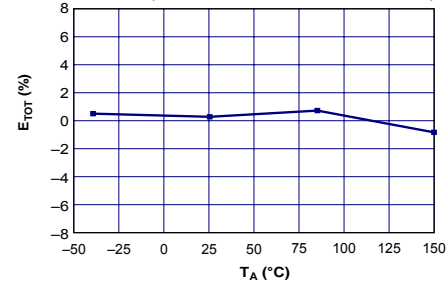
Magnetic Offset versus Ambient Temperature



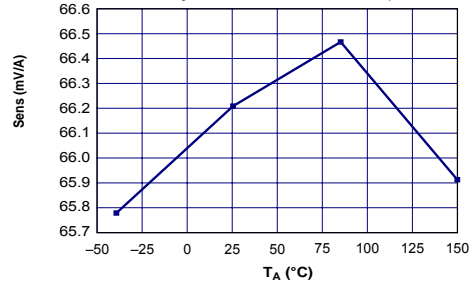
Nonlinearity versus Ambient Temperature



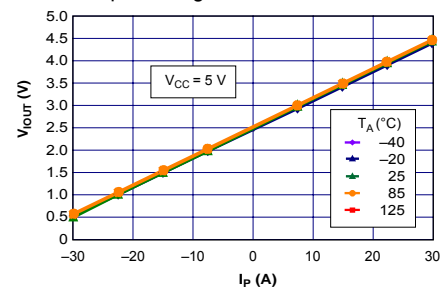
Mean Total Output Error versus Ambient Temperature



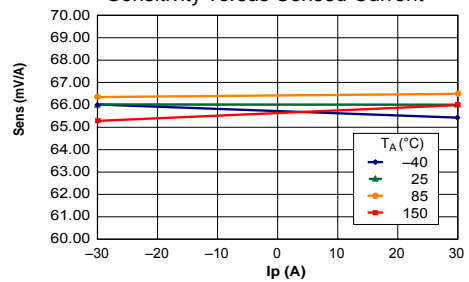
Sensitivity versus Ambient Temperature



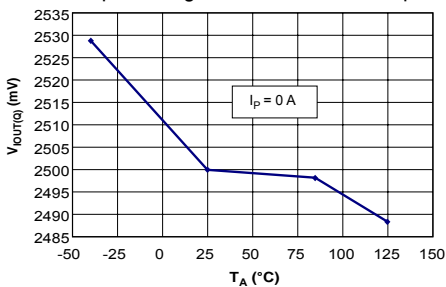
Output Voltage versus Sensed Current



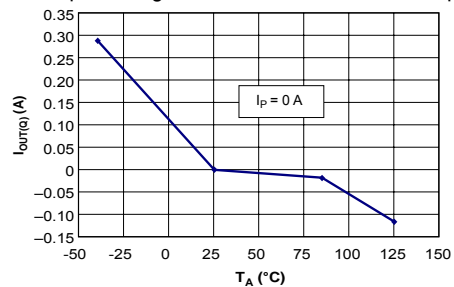
Sensitivity versus Sensed Current



0 A Output Voltage versus Ambient Temperature



0 A Output Voltage Current versus Ambient Temperature



Definitions of Accuracy Characteristics

Sensitivity (Sens). The change in sensor output in response to a 1 A change through the primary conductor. The sensitivity is the product of the magnetic circuit sensitivity (G/A) and the linear IC amplifier gain (mV/G). The linear IC amplifier gain is programmed at the factory to optimize the sensitivity (mV/A) for the full-scale current of the device.

Noise (V_{NOISE}). The product of the linear IC amplifier gain (mV/G) and the noise floor for the Allegro Hall effect linear IC (≈ 1 G). The noise floor is derived from the thermal and shot noise observed in Hall elements. Dividing the noise (mV) by the sensitivity (mV/A) provides the smallest current that the device is able to resolve.

Linearity (E_{LIN}). The degree to which the voltage output from the sensor varies in direct proportion to the primary current through its full-scale amplitude. Nonlinearity in the output can be attributed to the saturation of the flux concentrator approaching the full-scale current. The following equation is used to derive the linearity:

$$100 \left\{ 1 - \left[\frac{\Delta \text{ gain} \times \% \text{ sat} (V_{\text{IOUT_full-scale amperes}} - V_{\text{IOUT(Q)}})}{2 (V_{\text{IOUT_half-scale amperes}} - V_{\text{IOUT(Q)}})} \right] \right\}$$

where $V_{\text{IOUT_full-scale amperes}}$ = the output voltage (V) when the sensed current approximates full-scale $\pm I_p$.

Symmetry (E_{SYM}). The degree to which the absolute voltage output from the sensor varies in proportion to either a positive or negative full-scale primary current. The following formula is used to derive symmetry:

$$100 \left(\frac{V_{\text{IOUT_+ full-scale amperes}} - V_{\text{IOUT(Q)}}}{V_{\text{IOUT(Q)}} - V_{\text{IOUT_full-scale amperes}}} \right)$$

Quiescent output voltage (V_{IOUT(Q)}). The output of the sensor when the primary current is zero. For a unipolar supply voltage, it nominally remains at $V_{CC}/2$. Thus, $V_{CC} = 5$ V translates into $V_{\text{IOUT(Q)}} = 2.5$ V. Variation in $V_{\text{IOUT(Q)}}$ can be attributed to the resolution of the Allegro linear IC quiescent voltage trim and thermal drift.

Electrical offset voltage (V_{OE}). The deviation of the device output from its ideal quiescent value of $V_{CC}/2$ due to nonmagnetic causes. To convert this voltage to amperes, divide by the device sensitivity, Sens.

Accuracy (E_{TOT}). The accuracy represents the maximum deviation of the actual output from its ideal value. This is also known as the total output error. The accuracy is illustrated graphically in the output voltage versus current chart at right.

Accuracy is divided into four areas:

- **0 A at 25°C.** Accuracy of sensing zero current flow at 25°C, without the effects of temperature.
- **0 A over Δ temperature.** Accuracy of sensing zero current flow including temperature effects.
- **Full-scale current at 25°C.** Accuracy of sensing the full-scale current at 25°C, without the effects of temperature.
- **Full-scale current over Δ temperature.** Accuracy of sensing full-scale current flow including temperature effects.

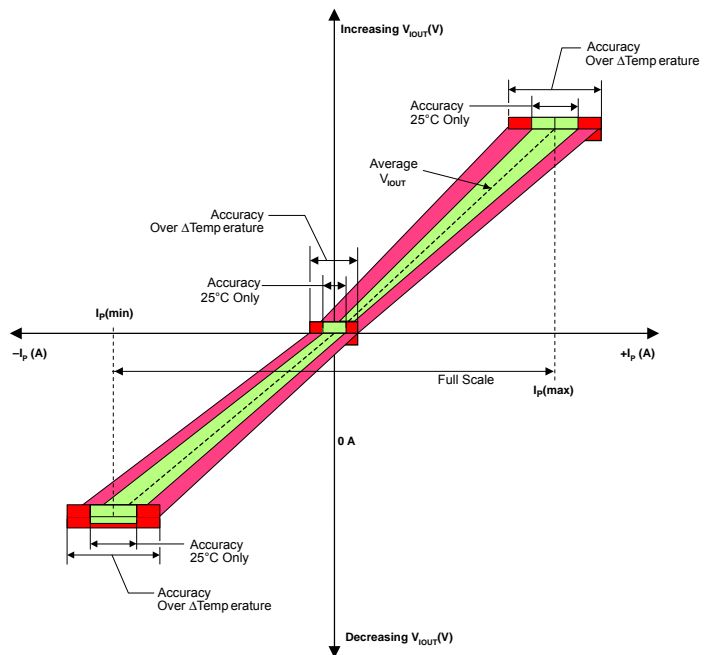
Ratiometry. The ratiometric feature means that its 0 A output, $V_{\text{IOUT(Q)}}$, (nominally equal to $V_{CC}/2$) and sensitivity, Sens, are proportional to its supply voltage, V_{CC} . The following formula is used to derive the ratiometric change in 0 A output voltage, $\Delta V_{\text{IOUT(Q)RAT}}$ (%).

$$100 \left(\frac{V_{\text{IOUT(Q)VCC}} / V_{\text{IOUT(Q)5V}}}{V_{CC} / 5 \text{ V}} \right)$$

The ratiometric change in sensitivity, $\Delta \text{Sens}_{\text{RAT}}$ (%), is defined as:

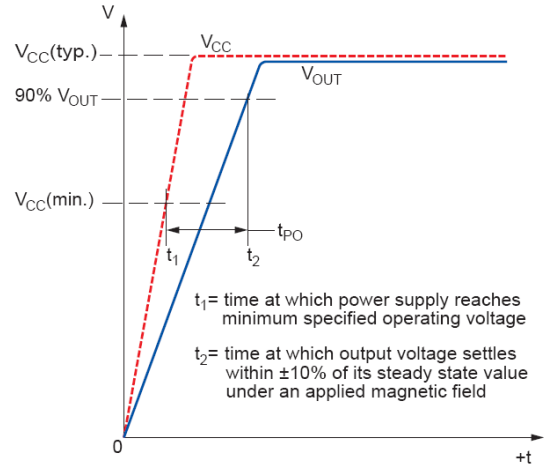
$$100 \left(\frac{\text{Sens}_{\text{VCC}} / \text{Sens}_{5\text{V}}}{V_{CC} / 5 \text{ V}} \right)$$

Output Voltage versus Sensed Current
Accuracy at 0 A and at Full-Scale Current

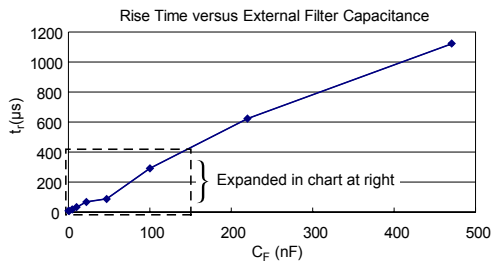
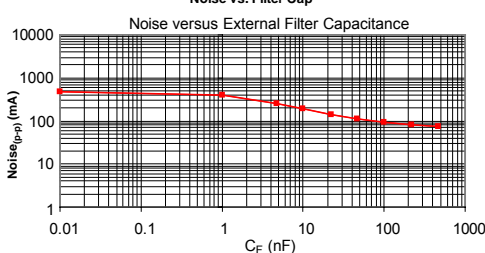
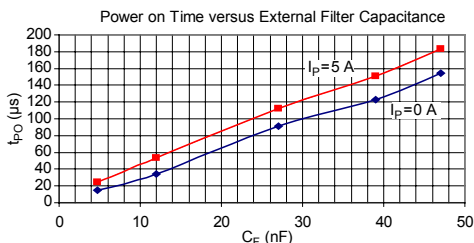
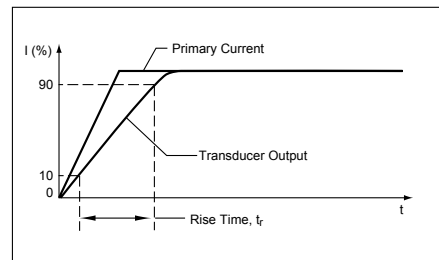


Definitions of Dynamic Response Characteristics

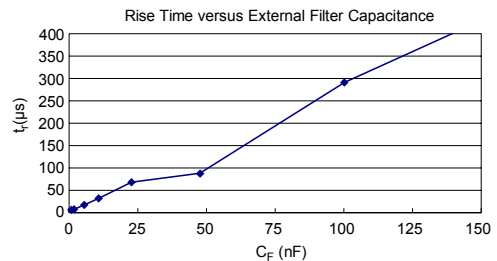
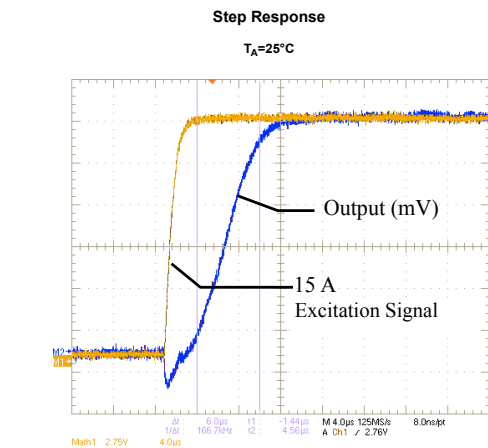
Power-On Time (t_{PO}). When the supply is ramped to its operating voltage, the device requires a finite time to power its internal components before responding to an input magnetic field. Power-On Time, t_{PO} , is defined as the time it takes for the output voltage to settle within $\pm 10\%$ of its steady state value under an applied magnetic field, after the power supply has reached its minimum specified operating voltage, $V_{CC(min)}$, as shown in the chart at right.



Rise time (t_r). The time interval between a) when the sensor reaches 10% of its full scale value, and b) when it reaches 90% of its full scale value. The rise time to a step response is used to derive the bandwidth of the current sensor, in which $f(-3 \text{ dB}) = 0.35/t_r$. Both t_r and $t_{RESPONSE}$ are detrimentally affected by eddy current losses observed in the conductive IC ground plane.



C_F (nF)	t_r (μ s)
0	6.6
1	7.7
4.7	17.4
10	32.1
22	68.2
47	88.2
100	291.3
220	623.0
470	1120.0

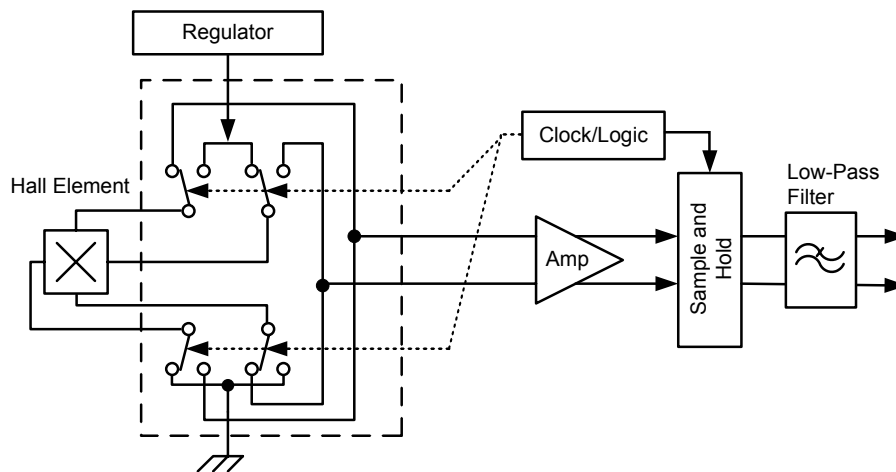


Chopper Stabilization Technique

Chopper Stabilization is an innovative circuit technique that is used to minimize the offset voltage of a Hall element and an associated on-chip amplifier. Allegro patented a Chopper Stabilization technique that nearly eliminates Hall IC output drift induced by temperature or package stress effects. This offset reduction technique is based on a signal modulation-demodulation process. Modulation is used to separate the undesired dc offset signal from the magnetically induced signal in the frequency domain. Then, using a low-pass filter, the modulated dc offset is suppressed while the magnetically induced signal passes through the filter.

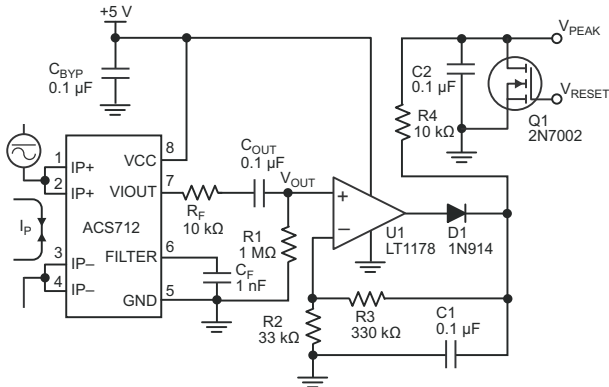
As a result of this chopper stabilization approach, the output voltage from the Hall IC is desensitized to the effects of temperature and mechanical stress. This technique produces devices that have an extremely stable Electrical Offset Voltage, are immune to thermal stress, and have precise recoverability after temperature cycling.

This technique is made possible through the use of a BiCMOS process that allows the use of low-offset and low-noise amplifiers in combination with high-density logic integration and sample and hold circuits.

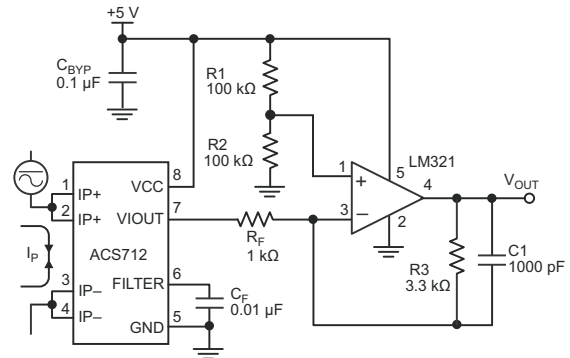


Concept of Chopper Stabilization Technique

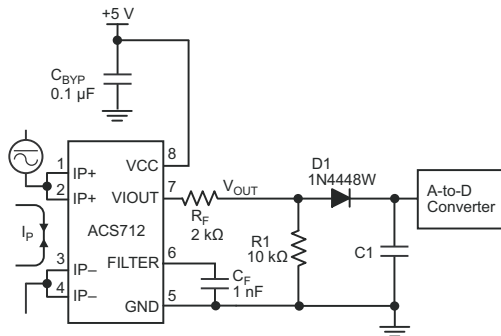
Typical Applications



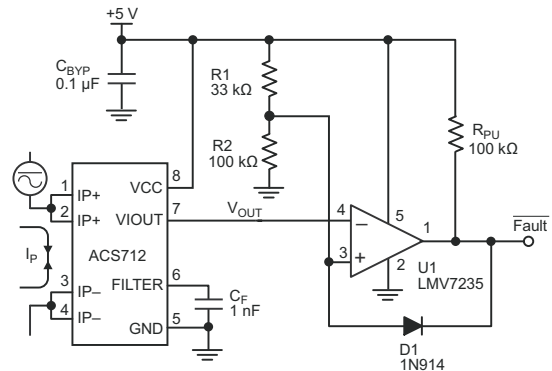
Application 2. Peak Detecting Circuit



Application 3. This configuration increases gain to 610 mV/A (tested using the ACS712ELC-05A).



Application 4. Rectified Output. 3.3 V scaling and rectification application for A-to-D converters. Replaces current transformer solutions with simpler ACS circuit. C1 is a function of the load resistance and filtering desired. R1 can be omitted if the full range is desired.



Application 5. 10 A Overcurrent Fault Latch. Fault threshold set by R1 and R2. This circuit latches an overcurrent fault and holds it until the 5 V rail is powered down.

Improving Sensing System Accuracy Using the FILTER Pin

In low-frequency sensing applications, it is often advantageous to add a simple RC filter to the output of the sensor. Such a low-pass filter improves the signal-to-noise ratio, and therefore the resolution, of the sensor output signal. However, the addition of an RC filter to the output of a sensor IC can result in undesirable sensor output attenuation — even for dc signals.

Signal attenuation, ΔV_{ATT} , is a result of the resistive divider effect between the resistance of the external filter, R_F (see Application 6), and the input impedance and resistance of the customer interface circuit, R_{INTFC} . The transfer function of this resistive divider is given by:

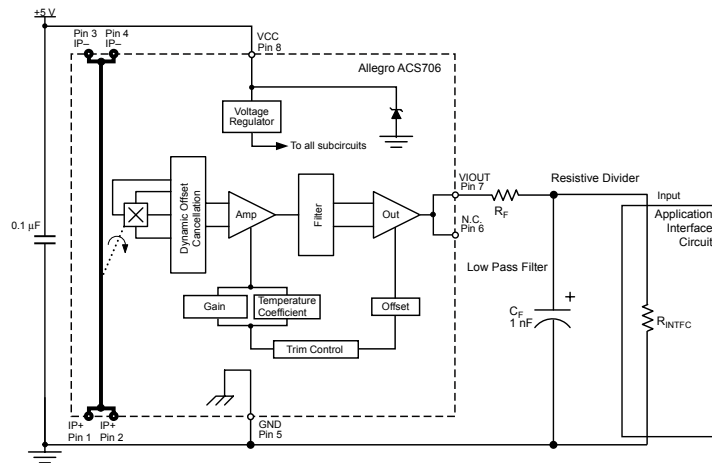
$$\Delta V_{ATT} = V_{IOUT} \left(\frac{R_{INTFC}}{R_F + R_{INTFC}} \right)$$

Even if R_F and R_{INTFC} are designed to match, the two individual resistance values will most likely drift by different amounts over

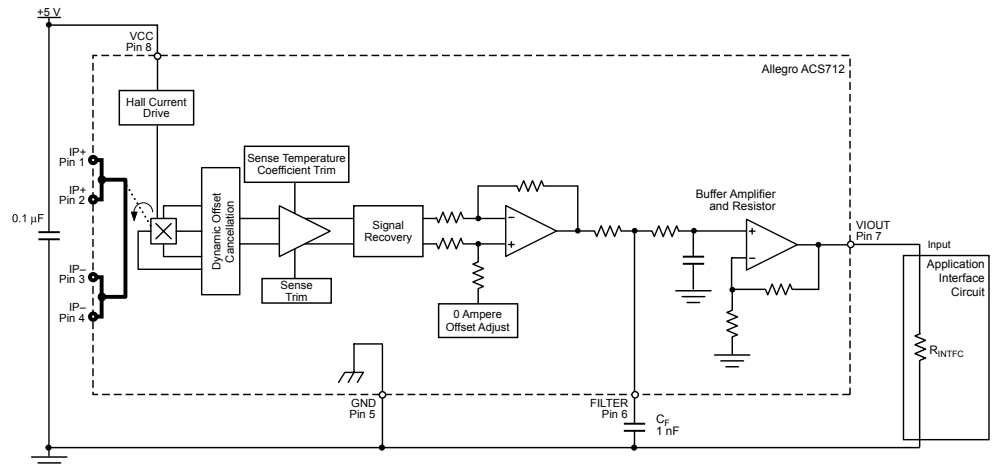
temperature. Therefore, signal attenuation will vary as a function of temperature. Note that, in many cases, the input impedance, R_{INTFC} , of a typical analog-to-digital converter (ADC) can be as low as 10 k Ω .

The ACS712 contains an internal resistor, a FILTER pin connection to the printed circuit board, and an internal buffer amplifier. With this circuit architecture, users can implement a simple RC filter via the addition of a capacitor, C_F (see Application 7) from the FILTER pin to ground. The buffer amplifier inside of the ACS712 (located after the internal resistor and FILTER pin connection) eliminates the attenuation caused by the resistive divider effect described in the equation for ΔV_{ATT} . Therefore, the ACS712 device is ideal for use in high-accuracy applications that cannot afford the signal attenuation associated with the use of an external RC low-pass filter.

Application 6. When a low pass filter is constructed externally to a standard Hall effect device, a resistive divider may exist between the filter resistor, R_F , and the resistance of the customer interface circuit, R_{INTFC} . This resistive divider will cause excessive attenuation, as given by the transfer function for ΔV_{ATT} .



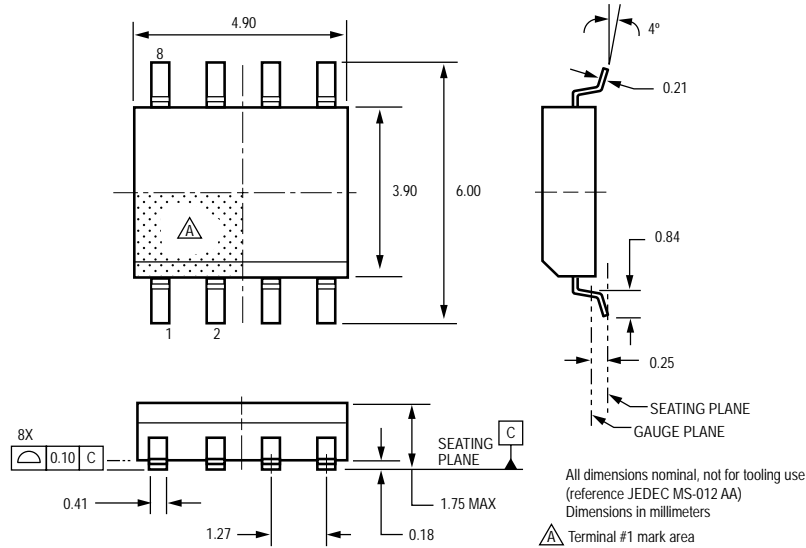
Application 7. Using the FILTER pin provided on the ACS712 eliminates the attenuation effects of the resistor divider between R_F and R_{INTFC} , shown in Application 6.



ACS712

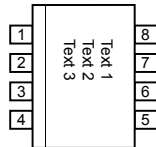
Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor

Package LC, 8-pin SOIC



Package Branding

Two alternative patterns are used



ACS712T RLCPPP YYWWA	ACS	Allegro Current Sensor
	712	Device family number
	T	Indicator of 100% matte tin leadframe plating
	R	Operating ambient temperature range code
	LC	Package type designator
	PPP	Primary sensed current
YY	Date code: Calendar year (last two digits)	
WW	Date code: Calendar week	
A	Date code: Shift code	

ACS712T RLCPPP L...L YYWW	ACS	Allegro Current Sensor
	712	Device family number
	T	Indicator of 100% matte tin leadframe plating
	R	Operating ambient temperature range code
	LC	Package type designator
	PPP	Primary sensed current
L...L	Lot code	
YY	Date code: Calendar year (last two digits)	
WW	Date code: Calendar week	

Copyright ©2006, 2007, Allegro MicroSystems, Inc.

The products described herein are manufactured under one or more of the following U.S. patents: 5,045,920; 5,264,783; 5,442,283; 5,389,889; 5,581,179; 5,517,112; 5,619,137; 5,621,319; 5,650,719; 5,686,894; 5,694,038; 5,729,130; 5,917,320; and other patents pending.

Allegro MicroSystems, Inc. reserves the right to make, from time to time, such departures from the detail specifications as may be required to permit improvements in the performance, reliability, or manufacturability of its products. Before placing an order, the user is cautioned to verify that the information being relied upon is current.

Allegro's products are not to be used in life support devices or systems, if a failure of an Allegro product can reasonably be expected to cause the failure of that life support device or system, or to affect the safety or effectiveness of that device or system.

The information included herein is believed to be accurate and reliable. However, Allegro MicroSystems, Inc. assumes no responsibility for its use; nor for any infringement of patents or other rights of third parties which may result from its use.

For the latest version of this document, visit our website:

www.allegromicro.com



Allegro MicroSystems, Inc.
115 Northeast Cutoff
Worcester, Massachusetts 01615-0036 U.S.A.
1.508.853.5000; www.allegromicro.com

Anhang B.

Programmcode

PROGRAMMCODE MODELBAU

```
void CRUnit::algorithm7()

{
float pitch, roll;

//define nivState
int start=1, pillarToGround=2, pillarToGroundsecond = 3,
checkGroundandPitch = 4, startLeveling = 5, pillarsToGroundAll= 6,
pitchBackside = 7, pitchFrontside = 8, checkRoll = 10;
int leftsideUp=11, rightsideUp = 12, checkState = 13,
pillarsToGroundAllFinal = 14 ;

//***** Start
if (this->nivState == start)
{
bt.write(F("Niv state: Start \n"));
for (int getTimes = 0; getTimes < 20; getTimes++)
{
this->mpu.getPR(&pitch_old, &roll_old, 'B');
}
// Serial.print(F("Pitch "));
// Serial.print(pitch_old);
// Serial.print(F(" Roll "));
// Serial.println(roll_old);

if(LB.groundContactNiv==true || RB.groundContactNiv==true ||
LB.groundContactNiv==true || RF.groundContactNiv==true )
{
if(pillarCurrentOffset[0]==0 || pillarCurrentOffset[1]==0 ||
pillarCurrentOffset[2]==0 ||pillarCurrentOffset[3]==0)
{
// Serial.print(F("\nMove pillars up to start\n"));
bt.write(F("#Error:3"));
this->nivState = 0;
}
else
{
this->nivState = pillarToGround;
bt.write(F("Niv state: Move pillar to ground\n"));
}
}
else
{
this->nivState = pillarToGround;
bt.write(F("Niv state: Move first pillar to ground\n"));
}
}
//***** Find ground with one of the pillars
if (this->nivState == pillarToGround)
{
//Start pillars and check current
this->pillarToGround(LB, false, LB.csGround, this->deltaVoltage);
this->pillarToGround(RB, false, RB.csGround, this->deltaVoltage);
this->pillarToGround(LF, false, LF.csGround, this->deltaVoltage);
this->pillarToGround(RF, false, RF.csGround, this->deltaVoltage);

//If one pillar is on ground Stop all other pillars
```

```

if(LB.pillarGround == true||RB.pillarGround == true||LF.pillarGround ==
true||RF.pillarGround == true)
{
RF.pillarStop();
LF.pillarStop();
RB.pillarStop();
LB.pillarStop();
delay(100);
if(LB.pillarGround == true)
this->pillarShortUp(LB);
if(RB.pillarGround == true)
this->pillarShortUp(RB);
if(LF.pillarGround == true)
this->pillarShortUp(LF);
if(RF.pillarGround == true)
this->pillarShortUp(RF);

this->nivState = pillarToGroundsecond;
bt.write(F("Niv state: move second pillar to ground\n"));
}

//check stability of pitch and roll
this->stabilityPitchRoll();
}
//***** Check which Pillar is first and move second on the same
side to ground
if (this->nivState == pillarToGroundsecond)
{
//Set Offset
this->setOffset();

//LB first RB second
if(LB.pillarGround == true)
{
this->pillarToGround(RB, true, RB.csGround, this->deltaVoltage);
if(RB.pillarGround == true)
{
bt.write(F("LB and then RB on ground\n"));
this->nivState = checkGroundandPitch;
}
}
//RB first LB second
else if(RB.pillarGround == true)
{
this->pillarToGround(LB, true, LB.csGround, this->deltaVoltage);
if(LB.pillarGround == true)
{
bt.write(F("RB and then LB on ground\n"));
this->nivState = checkGroundandPitch;
}
}
//LF first RF second
else if(LF.pillarGround == true)
{
this->pillarToGround(RF, true, RF.csGround, this->deltaVoltage);
if(RF.pillarGround == true)
{
bt.write(F("LF and then RF on ground\n"));
this->nivState = checkGroundandPitch;
}
}
//RF first LF second
else if(RF.pillarGround == true)

```

```

{
this->pillarToGround(LF, true, LF.csGround, this->deltaVoltage);
if(LF.pillarGround == true)
{
bt.write(F("RF and then LF on ground\n"));
this->nivState = checkGroundandPitch;
}
}
else
{
this->nivState == pillarToGround;
}

//check stability of pitch and roll
this->stabilityPitchRoll();
}
//Check if all pillars have to be on ground
if (this->nivState == checkGroundandPitch)
{
for (int getTimes = 0; getTimes < 20; getTimes++)
{
this->mpu.getPR(&pitch, &roll, 'B');
}
bt.write(F("checkGroundandPitch\n"));

if (RB.pillarGround == true && LB.pillarGround == true)
{
if (pitch > 0.5)
{
this->nivState = startLeveling;
bt.write(F("Niv state: StartLeveling\n"));
}
else
{
this->nivState = pillarsToGroundAll;
bt.write(F("Niv state: pillarsToGroundAll\n"));
}
}
else if (RF.pillarGround == true && LF.pillarGround == true)
{
if (pitch < -0.5)
{
this->nivState = startLeveling;
bt.write(F("Niv state: Start Leveling\n"));
}
else
{
this->nivState = pillarsToGroundAll;
bt.write(F("Niv state: pillarsToGroundAll\n"));
}
}
else
{
bt.write(F("Failure checkGroundandPitch\n"));
this->nivState = pillarToGround;
}
}
if (this->nivState == pillarsToGroundAll)
{
int checksum;
checksum=LB.pillarGround+RB.pillarGround+RF.pillarGround+LF.pillarGround;

this->pillarToGround(LB, true, LB.csGround, this->deltaVoltage);

```

```

this->pillarToGround(RF, true, RF.csGround, this->deltaVoltage);
this->pillarToGround(RB, true, RB.csGround, this->deltaVoltage);
this->pillarToGround(LF, true, LF.csGround, this->deltaVoltage);

if(LF.pillarMoving=='D') LF.needFinalToGround=true;
if(RF.pillarMoving=='D') RF.needFinalToGround=true;
if(LB.pillarMoving=='D') LB.needFinalToGround=true;
if(RB.pillarMoving=='D') RB.needFinalToGround=true;

//If all pillars are on ground move to the next step
if (LB.pillarGround == true && RB.pillarGround == true && LF.pillarGround
== true && RF.pillarGround == true)
{
this->nivState = checkState;
bt.write(F("Niv state: checkState\n"));
}
else
if(checksum<(LB.pillarGround+RB.pillarGround+RF.pillarGround+LF.pillarGroun
d))
{
// bt.write(F("Here Checked"));
if(LF.pillarMoving=='D')
{
LF.pillarStop();
LF.pillarMoveDown(0);
}
if(RF.pillarMoving=='D')
{
RF.pillarStop();
RF.pillarMoveDown(0);
}
if(RB.pillarMoving=='D')
{
RB.pillarStop();
RB.pillarMoveDown(0);
}
if(LB.pillarMoving=='D')
{
LB.pillarStop();
LB.pillarMoveDown(0);
}
delay(this->delayMoveDown);
}
}
if (this->nivState == startLeveling)
{
for (int getTimes = 0; getTimes < 20; getTimes++)
{
this->mpu.getPR(&pitch, &roll, 'B');
}
// Serial.print("\nStart Level\n");
bt.write(F("\nStart Level\n"));
//Serial.print(pitch);

//Tolerance for Pitch 0.5°
if (pitch > 0.5)
{
this->nivState = pitchBackside;
bt.write(F("pitchBackside\n"));
}
else if (pitch < -0.5)
{
this->nivState = pitchFrontside;

```



```

bt.write(F("pitchFrontside\n"));
}
else
{
if(LB.pillarGround == true && RB.pillarGround == true && LF.pillarGround ==
true && RF.pillarGround == true)
{
this->nivState = checkState;
}
else
{
this->nivState = pillarsToGroundAll;
}
}
}

if (this->nivState == pitchFrontside)
{
//deltaVoltage: maybe separate value for the following part
for (int getTimes = 0; getTimes < 20; getTimes++)
{
this->mpu.getPR(&pitch, &roll, 'B');
}
// Serial.print("\nPitch "); Serial.print(pitch); Serial.print(" ");

if(pitch > 0.5)
{
if(RF.groundContactNiv==true && LF.groundContactNiv==true)
{
RF.pillarStop();
LF.pillarStop();

//go back and bring all pillars to Ground
this->nivState = pillarsToGroundAll;

bt.write(F("\npillarsToGroundAll"));
// Serial.print("\npillarsToGroundAll");

}
else if (RF.groundContactNiv==true)
{
RF.pillarStop();
}
else if (LF.groundContactNiv==true)
{
LF.pillarStop();
}

//make sure currently moving pillars are on ground
if(RF.groundContactNiv==false && RF.pillarMoving != 'D')
{
RF.pillarMoveDownSlow(0);
delay(delayMoveDown);
}
else if(LF.groundContactNiv==false && LF.pillarMoving != 'D')
{
LF.pillarMoveDownSlow(0);
delay(delayMoveDown);
}

if(LF.pillarMoving=='D')
{
if(LF.checkCurrentMean(this->deltaVoltageNiv, Pillar::csNivConst))

```

```

this->nivState = this->maxMovementerror;
}
if(RF.pillarMoving=='D')
{
if(RF.checkCurrentMean(this->deltaVoltageNiv, Pillar::csNivConst))
this->nivState = this->maxMovementerror;
}
}
else
{
//Lift one side
//0...Back, 1...Front 2...Left, 3...Right
this->oneSideUp(1);
}
}

if (this->nivState == pitchBackside)
{
for (int getTimes = 0; getTimes < 20; getTimes++)
{
this->mpu.getPR(&pitch, &roll, 'B');
}
// Serial.print("\nPitch "); Serial.print(pitch); Serial.print(" ");

if(pitch <-0.5)
{
if(RB.groundContactNiv==true && LB.groundContactNiv==true )
{
RB.pillarStop();
LB.pillarStop();

//go back and bring all pillars to Ground
this->nivState = pillarsToGroundAll;

bt.write(F("\npillarsToGroundAll"));
// Serial.print("\npillarsToGroundAll");

}
else if (RB.groundContactNiv==true)
{
RB.pillarStop();
}
else if (LB.groundContactNiv==true )
{
LB.pillarStop();
}
//make sure currently moving pillars are on ground
if(RB.groundContactNiv==false && RB.pillarMoving != 'D')
{
RB.pillarMoveDownSlow(0);
delay(delayMoveDown);
}
else if(LB.groundContactNiv==false && RB.pillarMoving != 'D')
{
LB.pillarMoveDownSlow(0);
delay(delayMoveDown);
}

if(LB.pillarMoving=='D')
{
if(LB.checkCurrentMean(this->deltaVoltageNiv, Pillar::csNivConst))
this->nivState = this->maxMovementerror;
}
}

```

```

if(RB.pillarMoving=='D')
{
if(RB.checkCurrentMean(this->deltaVoltageNiv, Pillar::csNivConst))
this->nivState = this->maxMovementerror;
}
}
else
{
//Lift one side
//0...Back, 1...Front 2...Left, 3...Right
this->oneSideUp(0);
}
}

if (this->nivState == checkState)
{
// set groundcontact to false for last check
if(LB.needFinalToGround) LB.pillarGround=false;
if(RB.needFinalToGround) RB.pillarGround=false;
if(RF.needFinalToGround) RF.pillarGround=false;
if(LF.needFinalToGround) LF.pillarGround=false;

this->nivState = pillarsToGroundAllFinal;
bt.write(F("Niv state: pillarsToGroundAllFinal \n"));
}

if (this->nivState == pillarsToGroundAllFinal)
{
this->pillarToGroundFinal(LB, Pillar::csNivConst, this->deltaVoltageNiv);
this->pillarToGroundFinal(RB, Pillar::csNivConst, this->deltaVoltageNiv);
this->pillarToGroundFinal(LF, Pillar::csNivConst, this->deltaVoltageNiv);
this->pillarToGroundFinal(RF, Pillar::csNivConst, this->deltaVoltageNiv);

//If all pillars are on ground move to the next step
if (LB.pillarGround == true && RB.pillarGround == true && LF.pillarGround
== true && RF.pillarGround == true)
{
this->nivState = 0;
bt.write(F("Niv state: Finished \n"));
}
}
if (this->nivState == this->maxMovementerror)
{
LB.pillarStop();
RB.pillarStop();
LF.pillarStop();
RF.pillarStop();
this->nivState = 0;
this->sendMovementerror();
}
}

```

METHODEN DES ALGORITHMS

```
void CRUnit::writeAngle()
{
float pitch, roll;

// if(this->nivState!=0)
// {
for (int getTimes = 0; getTimes < 20; getTimes++)
{
this->mpu.getPR(&pitch, &roll, 'B');
}

bt.write(F("#Pitch:")); bt.write(String(pitch,1));
bt.write(F("#Roll:")); bt.write(String(roll,1));
// }
}

//*****
//One Pillar to Ground
//*****
void CRUnit::pillarToGround(Pillar &pil, bool moveShortUp, const String
&groundType, float deltaV)
{
//Start Pillar
if((!pil.pillarGround) && (pil.pillarMoving=='N'))
{
pil.pillarMoveDown(0);
delay(this->delayMoveDown);
}
//Check Current
if(pil.pillarMoving=='D')
{
if(pil.checkCurrentMean(deltaV, groundType))
this->nivState = this->maxMovementerror;

//Move up after groundcontact if "moveShortUp" is true
if(pil.pillarGround && moveShortUp)
{
pillarShortUp(pil);
// pil.pillarMoveUpShort();
// delay(this->delayMoveUpShort);
// pil.pillarStop();
}
}

//*****
//One Pillar to Ground slow
//*****
void CRUnit::pillarToGroundFinal(Pillar &pil, const String &groundType,
float deltaV)
{
//Start Pillar
if((!pil.pillarGround) && (pil.pillarMoving=='N'))
{
pil.pillarMoveDownSlow(0);
delay(this->delayMoveDown);
}
if(pil.pillarMoving=='D')
{
```

```

if(pil.checkCurrentMean(deltaV, groundType))
this->nivState = this->maxMovementerror;
}
}
//*****
//Move pillar up if on ground
//*****
void CRUnit::pillarShortUp(Pillar &pil)
{
pil.pillarMoveUpShort();
delay(this->delayMoveUpShort);
pil.pillarStop();
}
//*****
//check stability of pitch and roll
//*****
void CRUnit::stabilityPitchRoll()
{
float pitch, roll;

for (int getTimes = 0; getTimes < 20; getTimes++)
{
this->mpu.getPR(&pitch, &roll, 'B');
}
// Serial.print("Pitch ");
// Serial.print(pitch);
// Serial.print(" Roll ");
// Serial.print(roll);

if(abs(this->pitch_old-pitch)> this->pitch_tol)
{
bt.write("Pitch instabil! Nivellierung muss neu gestartet werden");
RF.pillarStop();
LF.pillarStop();
RB.pillarStop();
LB.pillarStop();
this->nivState = 0;
}

if(abs(this->roll_old-roll)> this->roll_tol)
{
bt.write("Roll instabil! Nivellierung muss neu gestartet werden");
RF.pillarStop();
LF.pillarStop();
RB.pillarStop();
LB.pillarStop();
this->nivState = 0;
}
}
//*****
// set offset
//*****
void CRUnit::setOffset()
{
this->pillarCurrentOffset[0] = RB.getLastForecast()-LB.getLastForecast();
this->pillarCurrentOffset[1] = RF.getLastForecast()-LF.getLastForecast();
this->pillarCurrentOffset[2] = LF.getLastForecast()-LB.getLastForecast();
this->pillarCurrentOffset[3] = RF.getLastForecast()-RB.getLastForecast();
}
//*****
// one side up
//*****
void CRUnit::oneSideUp(int offsetnumber)

```

```

{
Pillar * p1, * p2;
switch ( offsetnumber )
{
case 0:
{
p1=&RB;
p2=&LB;
}
break;
case 1:
{
p1=&RF;
p2=&LF;
}
break;
case 2:
{
p1=&LF;
p2=&LB;
}
break;
case 3:
{
p1=&RF;
p2=&RB;
}
break;
}

if(p1->pillarMoving=='D' && p2->pillarMoving=='D')
{
if((p2->getCurrent() + this->pillarCurrentOffset[offsetnumber] + this->deltaVoltageNiv) < p1->getCurrent())
{
p1->pillarStop();
}
else if((p2->getCurrent() + this->pillarCurrentOffset[offsetnumber]) > (this->deltaVoltageNiv + p1->getCurrent()))
{
p2->pillarStop();
}
}
else if (p1->pillarMoving=='D')
{
if(p1->checkCurrentMean(this->deltaVoltageNiv, Pillar::csNivConst))
this->nivState = this->maxMovementerror;
if(p1->groundContactNiv==true)
{
p2->pillarMoveDownSlow(0);
}
}
else if (p2->pillarMoving=='D')
{
if(p2->checkCurrentMean(this->deltaVoltageNiv, Pillar::csNivConst))
this->nivState = this->maxMovementerror;
if(p2->groundContactNiv==true)
{
p1->pillarMoveDownSlow(0);
}
}
}
else

```

```
{
p1->pillarMoveDownSlow(0);
p2->pillarMoveDownSlow(0);
}
}
//*****
//Send Movementerror
//*****

void CRUnit::sendMovementerror()
{
if(LB.getMovementstatus()>=100 || RB.getMovementstatus()>=100 ||
LF.getMovementstatus()>=100 || RF.getMovementstatus()>=100)
{
bt.write("#Error:2"); // Maximaler Ausfahrtsweg erreicht
}
else
{
bt.write("#Error:1"); // Überbelastung
}
}
```