Hochschule **Rosenheim**
University of Applied Sciences

Erweiterung der Reglerentwurfsumgebung pzMove und Demonstration anhand der Stabilisierung eines LEGO® MINDSTORMS® EV3-Roboters

Masterarbeit zum Erlangen des akademischen Grades
Master of Science (M.Sc.)
im Studiengang „Angewandte Forschung und Entwicklung“

Erstprüfer: Professor Dr.-Ing. Peter Zentgraf

Zweitprüfer: Professor Dr.-Ing. Wolfgang Schittenhelm

Autor:

Fischer Johannes

wohnhaft: Am Bachfeld 1, 83549 Eiselfing

geboren: 23.12.1991 in Wasserburg am Inn

Abgabedatum: 05. Dezember 2016

Erklärung zur Masterarbeit

Name, Vorname des Studierenden: Fischer, Johannes

Hochschule Rosenheim, Fakultät Ingenieurwissenschaften

Studiengang: Master „Angewandte Forschung und Entwicklung“

Hiermit erkläre ich, dass ich die Masterarbeit mit dem Thema

„Erweiterung der Reglerentwurfsumgebung pzMove und Demonstration anhand der Stabilisierung eines LEGO® MINDSTORMS® EV3-Roboters“

selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt, sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

Die Masterarbeit darf

- ☐ über die Fachhochschulbibliothek zugänglich gemacht werden.
- ☐ nach einer Sperrfrist von ...... Jahren über die Fachhochschulbibliothek zugänglich gemacht werden.
- ☐ nicht über die Fachhochschulbibliothek zugänglich gemacht werden.

....

(Datum) (Unterschrift des Studierenden)

Danksagung

Zu Beginn dieser Arbeit möchte ich mich bei allen Unterstützern bedanken. Arbeiten über diesen Umfang benötigen viel Zeit. Diese fehlt an anderer Stelle. Deshalb der Dank an meine Familie für das Verständnis.

Auch dem Erstprüfer, Herrn Professor Dr.-Ing. Peter Zentgraf, möchte ich meinen Dank für die Ausgabe des Themas und die umfangreiche Unterstützung meiner Tätigkeit aussprechen.

Abschließend gilt mein Dank Herrn Professor Dr.-Ing. Wolfgang Schittenhelm für die Übernahme der Zweitkorrektur.

Zielsetzung

Das Ziel dieser Arbeit ist die Erweiterung der Reglerentwurfsumgebung pzMove um die Kaskadenregelung und das Loop-Shaping-Verfahren zur Reglerauslegung. In pzMove sind viele Fehler vorhanden, die immer wieder zu Abstürzen führen und so die Verwendbarkeit der Anwendung einschränken. Somit ist im ersten Schritt neben der Vorbereitung von pzMove auf die folgenden Erweiterungen auch eine umfangreiche Fehlerkorrektur durchzuführen.

Die Anwendung unterstützt ausschließlich den Standardregelkreis. Für die spätere Anwendung auf den LEGO® MINDSTORMS® EV3-Roboter ist die Reglerentwurfumgebung um die Möglichkeit einer zweiten Kaskade zu erweitern.

Die zweite Erweiterung betrifft das Loop-Shaping-Verfahren. Dazu müssen verschiedene Korrekturglieder kombiniert werden können. Ausschlaggebend zur Bestimmung der Stabilität anhand des Nyquist-Kriteriums ist das Nyquist-Diagram des offenen Regelkreises. Deshalb muss die Nyquist-Kurve für alle Kaskaden für den offenen Regelkreis darstellbar sein.

Diese Erweiterungen sind abschließend anhand eines LEGO® MINDSTORMS® EV3-Roboters an der Problemstellung eines Segways® anzuwenden.

Inhaltsverzeichnis

Stichwortverzeichnis

Anfangsbereich Der Anfangsbereich des Nyquist-Diagramms bezeichnet in dieser Arbeit den Bereich der Nyquist-Kurve, in dem die Frequenzen sehr klein sind.

Endbereich Der Endbereich des Nyquist-Diagramms ist das Gegenstück zum Anfangsbereich und bezeichnet in dieser Arbeit den Bereich der Nyquist-Kurve, in dem die Frequenzen sehr hoch sind.

EV3-Brick Der EV3-Brick ist die Recheneinheit der LEGO® MINDSTORMS®-Serie. Alle Signale der Sensoren werden in dieser Recheneinheit verarbeitet und die Stellsignale der Motoren berechnet. Zudem beinhaltet der Brick einen Akku, mit dem dieser ohne externe Stromversorgung betrieben werden kann [5].

Graphical user interface development environment GUIDE ist der WYSIWYG-Editor von MATLAB® mit dem per Drag-and-Drop grafische Benutzeroberflächen für MATLAB®-Funktionen erstellt werden können [11].

Gyrodrift Der Gyrosensor misst die Winkelbeschleunigung des Pendels des Segways® und stellt dem EV3-Brick diskrete Werte zur Verfügung. Wird die Winkelbeschleunigung integriert und so der absolute Winkel berechnet, entsteht aufgrund der Zeitabstände zwischen den Einzelmesswerten eine Abweichung, die als Gyrodrift bezeichnet wird.

Inneres Moment Das innere Moment eines Motors bezeichnet das durch den Ankerstrom am Anker angreifende Moment. Dieses Moment teilt sich auf in das Schleppmoment, das Lastmoment und das Beschleunigungsmoment des Ankers.

Lastmoment Das Lastmoment ist das Moment, das vom Motor über die Welle auf das angetriebene Element übertragen wird. In dieser Arbeit wird der Motor und das Getriebe als eine Einheit betrachtet, wodurch mit Lastmoment das Moment an der Getriebewelle verstanden wird.

Motorencoder Die großen Motoren der LEGO® MINDSTORMS®-Serie haben Encoder integriert, die das Auslesen des absoluten Wellenwinkels, bezogen auf den Ausgangszustand am Programmstart, erlauben. Sie messen somit nicht den eigentlichen Motorwinkel, sondern den Winkel nach dem Getriebe.

Motorwinkel Der Motorwinkel des Segways® bezeichnet den absoluten Drehwinkel der Räder, bezogen auf den Pendelkörper.

Pendelwinkel Der Pendelwinkel bezeichnet den Winkel des Pendelkörpers des Segways®, bezogen auf seine aufrechte Position. Da der Pendelwinkel durch das manuelle Aufstellen des LEGO® MINDSTORMS® EV3-Roboters beim Programmstart eingeprägt wird, kann der gemessene Pendelwinkel auch beim Start eines Programms auf dem EV3-Brick bereits eine statische Abweichung haben.

Radwinkel Der Radwinkel ist der absolute Winkel der Räder des Segways® bezogen auf die Umgebung. Ist der Pendelwinkel gleich null, ist der Radwinkel identisch mit dem Motorwinkel.

Right-Half-Plane Als Right-Half-Plane wird der Bereich rechts von der imaginären Achse im Nyquist-Diagramm bezeichnet.

Schleppmoment Das Schleppmoment eines Motors bezeichnet die mechanischen Verluste eines Motors, zu denen unter anderem die Luftreibung und die Reibung in der Lagerung zählen. Zur Bestimmung des Schleppmoments ist der zu vermessende Motor mit einem anderen Motor mit konstanter Drehzahl anzutreiben und über eine Drehmomentmesswelle das benötigte Moment zu ermitteln.

Abkürzungsverzeichnis

AnalyDesign Der Analyse- und Design-Bereich von pzMove.

GUI Graphical user interface.

GUIDE Graphical user interface development environment.

RHP Right-Half-Plane.

WYSIWYAF What You See Is What You Asked For.

WYSIWYG What You See Is What You Get.

1 Einleitung

Diese Masterarbeit beschäftigt sich mit zwei großen Themenbereichen. Kenntnisse sind sowohl aus dem regelungstechnischen Bereichen der Kaskadenregelung und des Loop-Shapings, als auch aus dem Bereich der MATLAB®-Programmierung erforderlich. Die Aufgabenstellung beinhaltet die Aufarbeitung der Regelungsverfahren und Integration derer in die Reglerentwurfsumgebung pzMove inklusive der anschließenden Verifikation. Diese Arbeit soll in erster Linie nicht genau beschreiben, wie sich die Umsetzung der Aufgaben im Programmcode genau darstellen. Dies würde aufgrund der umfangreichen Überarbeitung des gesamten Programmcodes den Rahmen sprengen. Hingegen sind nachfolgend die Kaskadenregelung und das Loop-Shaping-Verfahren mit einer Schritt-für-Schritt-Anleitung für Studenten aufgearbeitet und dadurch für Lernende verständlich dargestellt. Des Weiteren wird gezeigt, dass die Umsetzung in pzMove auf diese Vorgehensweisen abgestimmt ist. So kann ein Student die Verfahren mit dem gezeigten Vorgehen 1:1 in der Reglerentwurfsumgebung pzMove umsetzen. Am Ende dieser Arbeit wird dies an einem LEGO® MINDSTORMS® EV3-Roboter exemplarisch durchgeführt.

1.1 Aufgaben der Reglerentwurfsumgebung pzMove

Die wichtigste Aufgabe der Reglerentwurfsumgebung pzMove besteht in der begleitenden Unterstützung der Vorlesung Regelungstechnik von Professor Dr.-Ing. Peter Zentgraf. Dazu wird pzMove als einfaches Veranschaulichungswerkzeug durch den Dozenten in der Vorlesung und auch als Bearbeitungswerkzeug von den Studenten in den Praktikas eingesetzt. Für einen außenstehenden Anwender können mögliche

Funktionen und Eigenheiten der Anwendung auf den ersten Blick unsinnig oder nicht zielführend erscheinen. Dies rührt daher, dass durch den Dozenten verschiedene Besonderheiten der Regelungstechnik darstellbar sein müssen.

Des Weiteren dient pzMove auch als Unterstützungstool für wissenschaftliche Mitarbeiter, Lehrbeauftragte und Studenten in deren jeweiligen Projekten. Deshalb muss neben der Darstellungs- und Lernfunktion auch der produktive Einsatz in einem möglichst großen Bereich der Regelungstechnik möglich sein. Somit bildet neben den Standardverfahren die Flexibilität der Anwendungsmöglichkeiten von pzMove ein weiteres, wichtiges Augenmerk bei der Anpassung und Erweiterung der Anwendung.

Für die Lehrfunktion sollten die Standardverfahren möglichst Step-by-Step dargeboten werden. Für den fortgeschrittenen Anwender können die Standardabläufe den flexiblen Einsatz der Einzelfunktionen behindern. Deshalb muss ein Mittelweg zwischen der gestützten Reglerauslegung und der Flexibilität im Einsatz gefunden werden. Die Gefahr besteht darin, dass der Mittelweg weder flexibel noch unterstützend ist und somit die gewünschte Funktion komplett verloren geht.

1.2 Vorgeschichte

Das Programm wurde bis zum aktuellen Stand von Professor Dr.-Ing. Peter Zentgraf und zwei Studenten entwickelt. Es besteht aus zwei Hauptprogrammen, die komplett unabhängig voneinander laufen. Der Identifier dient zum Berechnen einer Übertragungsfunktion aus der Messung von Eingangs- und Ausgangssignal. Der zweite Programmteil ermöglicht die Analyse der Übertragungsfunktionen und anschließend die Reglerauslegung mit verschiedenen Verfahren. Beide Programmteile können jeweils vom anderen aus geöffnet werden, wobei die Datenübergabe über die Hintergrunddaten des geöffneten Figures realisiert wurde.

2 Vorbereitung der Reglerentwurfsumgebung pzMove

Die Zweiteilung des Tools und die minimalinvasiven Erweiterungen führten zu zwei Programmteilen mit jeweils ca. 5000 Programmzeilen, die sich unübersichtlich und verschachtelt darstellten. Nach der Einarbeitung in den Programmcode waren folgende Punkte auffällig:

- Zwei eigenständige Programmoberflächen, welche mit einem gemeinsamen Datensatz in den Hintergrunddaten der Figure arbeiten und so keine klare Trennung der Teile zulassen.
- Verschachtelte, schwer verständliche Unterfunktionen.
- Veraltete Programmteile, die aufgrund von Anpassungen überflüssig geworden sind.
- Häufige Code-Wiederholungen, die bei Änderungen Mehrfachanpassungen notwendig machen.
- Skalierungsfehler in der durch GUIDE erstellten Programmoberfläche, abhängig von der verwendeten MATLAB®-Version.
- Zusätzliche Einzüge nach den Kommentaren zur vermeintlichen Strukturierung der langen Unterfunktionen, die keine Zuordnung der Code-Teile zu den einzelnen Strukturierungsfunktionen zulassen.

Aufgrund dieser Erkenntnisse stellte sich heraus, dass eine komplette Überarbeitung des Programms nötig war, um nicht nur die neuen Funktionen einfacher integrieren zu können, sondern auch für weitere Bearbeiter die Übersichtlichkeit zu wahren. Nachfolgend sind die groben Überarbeitungsmaßnahmen dargestellt, anhand derer auch zukünftige Änderungen gemacht werden sollten.

2.1 Zusammenführung von Identifier und Reglerentwurfsumgebung

Zuerst war es nötig, beide Programmteile zu einer gemeinsamen Oberfläche zusammenzuführen. Dabei sollte weiterhin eine vollständige Trennung der Programmteile als eigenständige Dateien erhalten bleiben und zudem eine Trennung der Hintergrunddaten von Identifier und AnalyDesign erfolgen. Des Weiteren waren die Projektdaten von den Einstellungen getrennt zu halten. MATLAB® bietet auch die Möglichkeit objektorientiert zu programmieren.

Bei der prozeduralen Programmierung besteht ein Programm hauptsächlich aus Funktionen mit lokalen und globalen Variablen. Der Programmablauf wird durch Ablaufstrukturen wie Schleifen und Abfragen gesteuert. Bei der objektorientierten Programmierung wird hingegen in Klassen und Objekten gedacht. Eine Klasse ist ein Bauplan für gleichartige Objekte. Jede Klasse definiert für alle aus ihr erzeugten Objekten die Objekteigenschaften („Properties“) und die Methoden („Methods“). Es ist möglich die Properties eines Objektes direkt auszulesen oder zu ändern. Über definierte Methoden einer Klasse können die Objekteigenschaften zudem ausgelesen oder geändert werden. Der Vorteil liegt darin, dass klasseninterne Änderungen über die sogenannten Getter- und Setter-Funktionen nach außen hin keine Änderungen verursachen. [13, S. 161+162]

Nachfolgend ein kurzes Gedankenbeispiel, um den hier entscheidenden Vorteil demonstrieren zu können. Eine Beispielklasse könnte „Student“ heißen und die Eigenschaften „Name“ und „Alter“ haben. In Matlab würde der Code dieser Klasse wie folgt lauten [12]:

```
classdef Student
    properties
        Name;
        Alter;
    end
    methods
        function initialize(obj, Name, Alter)
          obj.Name = Name;
          obj.Alter = Alter;
        end
        function Alter = getAlter(obj)
          Alter = obj.Alter;
        end
        function setAlter(obj, Alter)
          obj.Alter = Alter;
        end
    end
end
```

Die Klasse „Student“ ist jedoch nur der Bauplan. Anschließend muss aus der Klasse ein reales Objekt wie zum Beispiel „StudentXY“ aus der Klasse erzeugt werden. In diesem Beispiel soll der „StudentXY“ mit der Eigenschaft „Name = Mustermann“ und „Alter = 99“ erzeugt werden.

```
StudentXY = Student;
initialize(StudentXY, "Mustermann", 99);
```

Nun kann das „Alter“ des Studenten „StudentXY“ an gegebener Stelle wie folgt ausgelesen werden:

```
AlterXY = getAlter(StudentXY);
```

Merkt man nun, dass jeder Student jedes Jahr ein Jahr älter wird und möchte die Altersanpassung verhindern, kann man zum Beispiel statt dem Alter das Geburtsjahr als Eigenschaft festlegen. Dazu müsste man die Klasse wie folgt anpassen:

```
classdef Student
    properties
        Name;
        Geburtsjahr;
    end
    methods
        function initialize(obj, Name, Alter)
          obj.Name = Name;
          Systemzeit = clock;
          obj.Geburtsjahr = Systemzeit(1)-Alter;
        end
        function Alter = getAlter(obj)
          Systemzeit = clock;
          Alter = Systemzeit(1)-obj.Geburtsjahr;
        end
        function setAlter(obj, Alter)
          Systemzeit = clock;
          obj.Geburtsjahr = Systemzeit(1)-Alter;
        end
    end
end
```

Dadurch würden sich für den Rest des Programmcodes keine Änderungen ergeben. Der Aufruf der Methoden zur Initialisierung, Altersdefinition und Altersausgabe hat sich nicht geändert. Diese Unabhängigkeit ist der entscheidende Vorteil, der für die Zusammenführung des Identifiers und des Analyse-Design-Teils wichtig ist. Die gesamte Programmstruktur ist in Abb. 2.1 dargestellt. Sie wird aus drei Hauptkomponenten gebildet. Der Identifier und der Analyse-Design-Teil (AnalyDesign) bilden jeweils eine eigene Klasse mit eigenen Eigenschaften und Funktionen. Das Hauptfenster leitet von beiden Klassen je ein Objekt ab und initialisiert dieses. Bei der Initialisierung wird den Objekten das Hauptpanel des Hauptfensters übergeben, das als Parent für die Oberflächen vom Identifier und AnalyDesign fungiert. Daraus ergeben sich mehrere Vorteile:

- Die Code-technische Trennung zwischen Identifier und AnalyDesign bleibt bestehen.
- Die Klassen des Identifiers und von AnalyDesign haben jeweils die Strukturen „data“ und „settings“ als Eigenschaften, wodurch die Daten beider Teile voneinander vollständig getrennt und zudem je Teil auch noch in Daten und Einstellungen aufgetrennt sind.
- Die graphische Oberfläche ist nur noch code-technisch in drei Teile (Hauptfenster, Identifier und AnalyDesign) aufgeteilt. Bei der Darstellung ist das GUI eine Einheit.
- Die Projektverwaltung kann mit Getter und Setter der Klassen realisiert werden und ist somit unabhängig von Änderungen in den Klassen bei entsprechender Anpassung der Get- und Set-Funktionen.

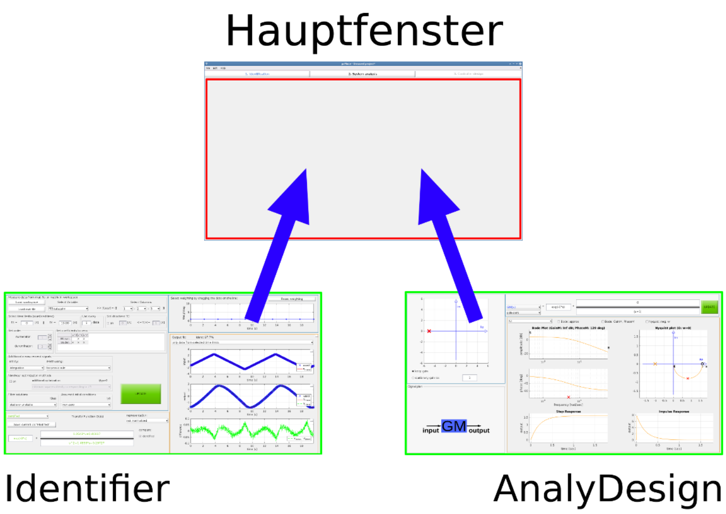


Abb. 2.1: Programmstruktur pzMove

Die Trennung in drei Hauptkomponenten und die Verwendung des objektorientierten Ansatzes für den Identifier und den AnalyDesign-Teil ermöglicht somit die unabhängige Änderung jedes einzelnen Teils ohne der Beeinflussung der jeweils beiden anderen Teile.

2.2 Anpassung an verschiedene Plattformen

Die Verwendung von pzMove sollte auf verschiedenen Plattformen möglich sein. Die verschiedenen Plattformen wie Windows oder Linux verwenden unterschiedliche Ordner-Trennzeichen. Dabei ist auf die Verwendung von „/“ oder „\“ zu verzichten. Anstelle kann die MATLAB®-Funktion „filesep“ [10] verwendet werden. Abhängig von der Plattform, auf der MATLAB® ausgeführt wird, gibt diese Funktion das entsprechende Trennzeichen zurück.

2.3 Neuaufbau der Oberfläche

Die Zusammenführung der Oberfläche, die Verwendung der Klassen und die Tatsache, dass bei verschiedenen MATLAB®-Versionen mit GUIDE Skalierungsfehler auftraten, bedurften einen kompletten Neuaufbau aller graphischen Oberflächen. Auf die Verwendung von GUIDE wurde komplett verzichtet, da nur dadurch die Skalierungsfehler, wie in Abb. 2.2 zu sehen, auszuschließen sind.

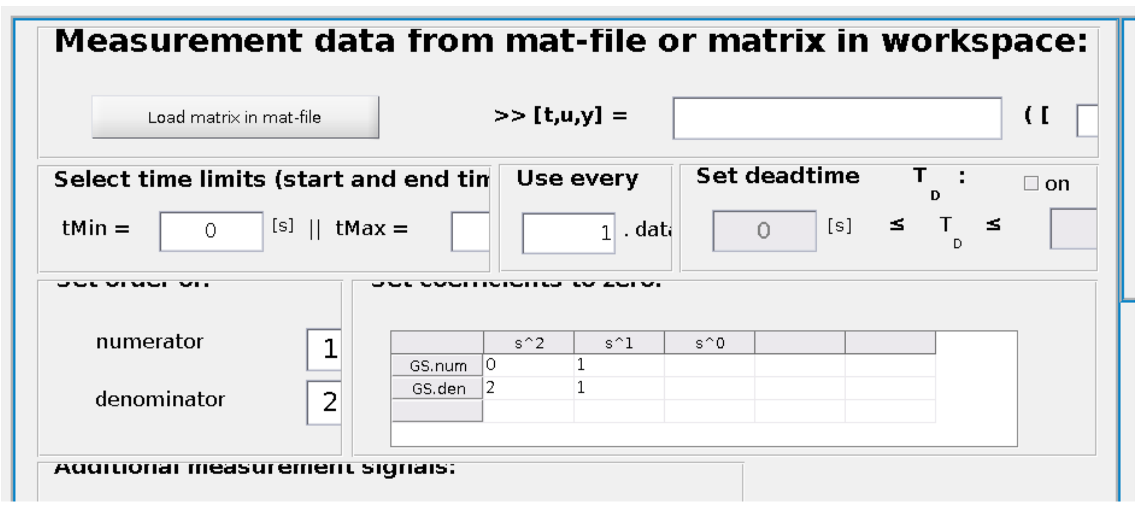


Abb. 2.2: Verdeckte GUI-Elemente aufgrund von Skalierungsfehlern

Des Weiteren stellt GUIDE ein grundsätzliches Problem dar. Durch den WYSIWYG-Ansatz („What You See Is What You Get“) ist es nur schwer möglich, auf im Hintergrund versteckte Objekte zuzugreifen. Für eine einfache Oberfläche ohne übereinanderliegende Objekte ist dieser Ansatz einfach handzuhaben. Der Unterschied zwischen dem WYSIWYG- und dem WYSIWYAF-Ansatz („What You See Is What You Ask For“) ist in Abb. 2.3 dargestellt. Der Nachteil der hardgecodeten Oberfläche eines WYSIWYAF-Ansatzes liegt auf der Hand. Es muss der Code für die Oberfläche per Hand eingegeben werden und erst beim Ausführen der Anwendung ist das Resultat ersichtlich. Dies führt zu einer wesentlich langwierigeren Einarbeitung. Beim AnalyDesign-Teil liegen hingegen mehrere Panels mit deren Kind-Objekten übereinander. Möchte man ein verdecktes Panel in GUIDE ändern, kann es zu ungewollten Änderungen in unbeteiligten Oberflächenkomponenten kommen, wodurch der Aufwand für derartige Änderungen sehr schnell ungewollte Ausmaße annehmen kann. Bei einer gecodeten Oberfläche muss hingegen nur der entsprechende Code-Teil gefunden werden. Ist die entsprechende Stelle lokalisiert, sind Änderungen an Eigenschaften wie zum Beispiel Farbe, Größe oder Position mit einfachen Anpassungen in den entsprechenden Programmteilen möglich, ohne dass dadurch Veränderungen an ungewollten Stellen eintreten können.

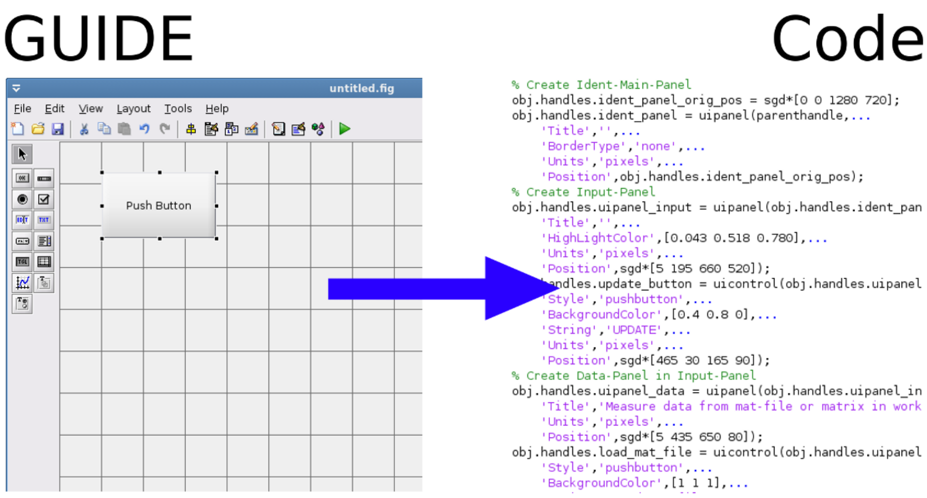


Abb. 2.3: Unterschied zwischen WYSIWYG (links) und WYSIWYAF (rechts)

Die Oberflächenprogrammierung in MATLAB® stellt sich bei näherer Betrachtung als sehr einfach dar. Meist ist jedes Objekt mit drei bis sechs Parametern anzulegen. Dabei ist immer der erste Parameter der Parent-Parameter, also auf welches Eltern-Element sich die Angaben von Größe und Position beziehen. Parameter zwei ist die Einheit, in der die Größe angegeben wird. In dieser Umsetztung ist die Größe jedes Objektes in Pixel angegeben. Dies erleichtert die Größenberechnung. So können zum Beispiel Abstände zwischen zwei Panels immer mit 5 Pixel angegeben werden. Sind alle Objekte definiert, ist anschließend mit wenigen einfachen Programmzeilen die gesamte Oberfläche skalierbar zu machen. Der dritte Parameter, den jedes Objekt besitzt, ist die Größe. Dieser Parameter besteht aus vier Komponenten. Die erste und zweite Komponente stellen den Abstand von links und von unten zum Elternelement dar. Komponenten drei und vier sind die Breite und die Höhe. Die weiteren Parameter der Objekte unterscheiden sich bei den unterschiedlichen Objekttypen. Auf die genaue Darstellung wird hier verzichtet. [9]

Die Oberfläche ist in einer festgelegten Größe im Format 16:9 mit 1290x760 Pixel aufgebaut und anschließend skalierbar gemacht. Dabei sind zwei Probleme zu beobachten:

- Die Skalierung der Schrift ist abhängig von der Höhe des Figures. Wird das Fenster nur schmäler gemacht, aber die Höhe nicht verändert, bleibt die Schriftgröße identisch. Dies führt zu Überlauf an manchen Stellen. Deshalb ist die Oberfläche so ausgelegt, dass bei einem Format von 4:3 gerade kein Überlauf auftritt. So ist garantiert, dass auch bei älteren Bildschirmen mit diesem Format keine Darstellungsfehler auftreten.

- MATLAB® ist nicht in der Lage eine graphische Benutzeroberfläche aufzubauen, welche größer als die zur Verfügung stehende Auflösung des Bildschirms ist. Durch den fixen Aufbau mit 1290x760 Pixel treten somit Darstellungsfehler auf, wenn die Auflösung des verwendeten Bildschirms kleiner als 1290x760 Pixel ist. Damit dieser Fehler nicht auftritt, wird vor jede Größenangabe bei der Objektdefinition ein Scale-Faktor eingefügt. Der Scale-Faktor in pzMove lautet „sgd“ und beträgt 0,85. Dieser Wert wird an die „intialize“-Methoden der Klassen übergeben und ist nur einmal im Hauptprogramm definiert. Nur das Hauptfenster mit dem Identifier und dem AnalyDesign sind skalierbar. Die Menüfenster sind zur Vereinfachung nicht skalierbar. Deren Größe ist der ausschlaggebende Faktor für die Verwendung eines Scale-Faktors von 0,85. Ist ein kleinerer Scale-Faktor nötig, treten unvermeidbar Darstellungsprobleme bei den Menüfenstern auf. Soll zukünftig die Verwendung auf Bildschirmen mit noch geringeren Auflösungen möglich sein, muss der Scale-Faktor kleiner gewählt werden und zudem sind die Menüfenster skalierbar zu machen.

2.4 Überarbeitung des gesamten Programmcodes zur Strukturierung

Mit den vorangegangenen Überarbeitungsmaßnahmen wurde die Hauptstruktur der Gesamtanwendung komplett neu aufgebaut. Der Programmcode an sich wurde dadurch schwer nachvollziehbar. Der Code wurde deshalb anhand nachfolgender Richtlinien überarbeitet:

- Klare Funktionsabfolgen mehrerer Unterfunktionen: Der Identifier und der AnalyDesign-Teil folgen einem übersichtlichen Funktionsablauf. So werden bei einer Aktion an der graphischen Benutzeroberfläche zuerst die Eingaben von der GUI eingelesen und deren Validität überprüft. Im weiteren Schritt werden die Hintergrunddaten neu berechnet und eventuelle Fehler oder Warnhinweise aktiviert. Abschließend erfolgt der Aufruf der Unterfunktionen zum Aktualisieren der Oberfläche und zum Darstellen der Diagramme. Das Einlesen, Berechnen und Ausgeben erfolgt wiederum in mehreren Unterfunktionen, die sich gegenseitig aufrufen. Somit ist es möglich eine Zwischenfunktion aus der GUI heraus aufzurufen und dadurch Teile der gesamten Berechnungsabfolge zu umgehen. Dies erhält die klare Struktur in den Programmteilen und ermöglicht zudem die Beschränkung der Berechnungen im Hintergrund auf ein Minimum.
- Anstelle der Verwendung von Copy-and-Paste sind Unterfunktionen zu erstellen:

Diese Regel reduzierte vor allem beim Identifier den Programmcode von 5210 Programmzeilen auf 1589 Zeilen. Dadurch ist der Programmcode zwar nicht weiter strukturiert, aber durch die Kürzung erheblich einfacher zu verstehen und zu ändern, ohne dass dabei ein großer Aufwand durch Mehrfachänderungen oder Probleme durch lokale Zusammenhänge entstehen können. So wurden im Identifier die Programmzeilen für die Plots der drei Diagramme im rechten unteren Teil des Programmfenster 15 mal, wie in Abb. 2.4 zu sehen, in fast identischer Weise erneut eingefügt. Verschiebt man die Programmzeilen einmal in eine Unterfunktion und passt die Zeilen mit Abfragen entsprechend den Eingabewerten an, ist man in der Lage diese 170 Programmzeilen 14 mal einzusparen, was bereits eine Programmkürzung von 2380 Zeilen ermöglicht.

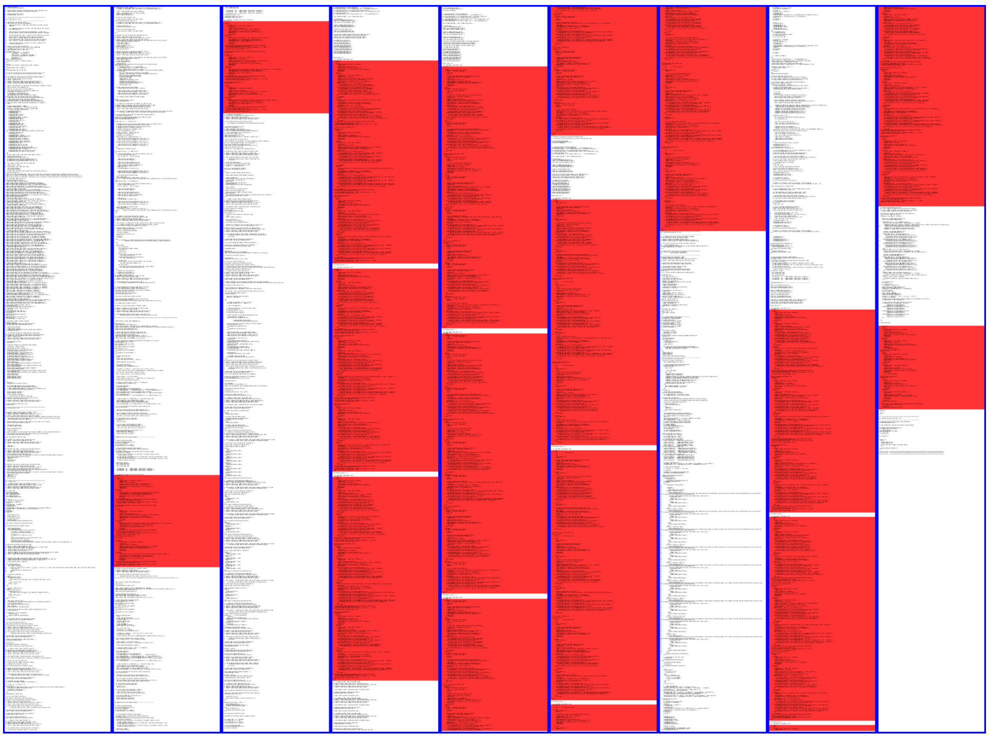


Abb. 2.4: Überflüssige Code-Teil-Wiederholung im bestehenden Identifier (rot dargestellt)

- Unterfunktionen mit aussagekräftigen Funktionsnamen verbessern die Übersichtlichkeit in der aufrufenden Funktion: Bei Funktionen mit verschiedenen

Teilaufgaben, die abhängig von Eingangsvariablen sind, sind viele Abfragen nötig, um die gewünschten Aktivitäten zu erreichen. Das führt zu großen Funktionen in denen die Code-Teile der Einzelfunktionen ineinander vermischt sind. Dies war vor allem beim AnalyDesign-Teil der Fall. Funktionen mit über 300 Zeilen und mehreren verschachtelten Einzelbereichen ergeben einen unübersichtlichen Code, der weder für den Einsteiger verständlich, noch für den fortgeschrittenen Programmierer nachvollziehbar ist. In Abb. 2.5 ist auf der linken Seite eine Funktion mit 321 Zeilen dargestellt. Auf der rechten Seite im Vergleich dazu die überarbeitete und trotz vieler Erweiterungen einfachere Funktion mit klar definierten Unterfunktionen, deren Aufrufe grün dargestellt sind. Durch die Zusammenführung der Einzelbereiche der verschiedenen Teilaufgaben in Unterfunktionen mit aussagekräftigen Namen, erhält man in der eigentlichen Funktion die verschiedenen Abfragen, in denen die Unterfunktionen aufgerufen werden. Somit ergeben sich die Hauptfunktion mit einfachen Abfragebedingungen und mehrere Unterfunktionen mit jeweils zusammengehörenden Codezeilen.

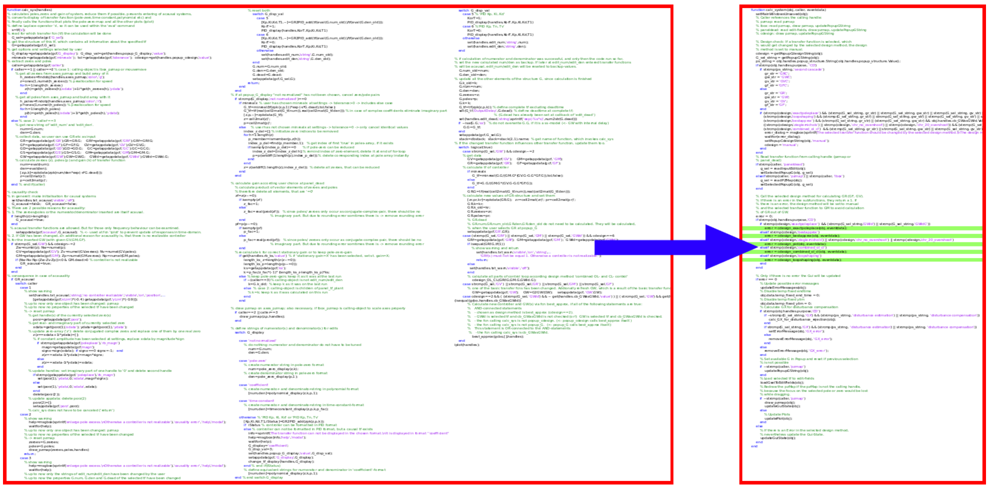


Abb. 2.5: Vereinfachung von Funktionen durch weitere Unterfunktionen

2.5 Fehlerbeseitigung

Die letzten Schritte der Überarbeitung von pzMove betreffen die Fehlerbeseitigung, die Funktionsklarstellung und die Verbesserung, wobei bei der Verbesserung das Augenmerk auf der Intuitivität liegt. Die Fehlerbeseitigung wurde bereits in Abschnitt 2.4 größten Teils abgedeckt. Bei der vorangegangenen Überarbeitung und Umstrukturierung war die Erkennbarkeit von bestehenden Fehlern sehr groß. Das eigentliche Problem der Fehlerbehebung stellen die neu eingearbeiteten Fehler dar. Trotz vieler Kontrollen und Tests sind diese Fehler nur schwer zu erkennen. Mit der Projektverwaltung kann das gespeicherte Projekt bei einem Fehler zusammen mit einer kurzen Beschreibung erheblich zur Behebung des Fehlers beitragen. So ist bei zukünftig auftretenden Fehlern sofort eine Sicherung des Projekts und eine kurze Fehlerbeschreibung anzulegen. Die Beschreibung besteht dabei aus drei Bereichen. Der erste Teil einer Fehlerbeschreibung sollte die vorangegangenen Schritte erläutern. Der zweite Teil der Fehlerbeschreibung umfasst die genaue Beschreibung des Istzustandes. Der letzte Teil der Fehlerbeschreibung sollte möglichst genau den Sollzustand mit Begründung dessen sein. Eine Beschreibung des Fehlers in diesen drei Schritten hilft dem Bearbeiter bei der Fehlersuche und bei der Fehlerbehebung. Nachfolgend zusammengefasst sind die Bestandteile einer Fehlerbehebungsanfrage zum optimalen Informationsaustausch zwischen Programmierer und Anwender:

- Gespeichertes Projekt mit dem auftretenden Fehler ohne weitere Bearbeitung nach Fehlereintritt
- Fehlerbeschreibung bestehend aus drei Bereichen
  - Letzte Bearbeitungsschritte vor dem Fehlereintritt
  - Genaue Beschreibung des Istzustandes
  - Genaue Beschreibung des Sollzustandes

2.6 Integration einer Projektverwaltung

Eine wichtige neue Funktion für den produktiven Einsatz stellt die Projektverwaltung dar. Das Menü der Projektverwaltung ist in Abb. 2.6 zu sehen. Damit ist es möglich Projektstände mit allen Eingaben und Einstellungen in einer „.pzMove"-Datei zu speichern. Zuvor war es nur möglich einzelne Übertragungsfunktionen zu exportieren. Mit der Projektverwaltung werden nicht nur alle vorhandenen Übertragungsfunktionen und Einstellungen, sondern auch der Status der kompletten Oberfläche gespeichert. Somit kann nach einem Speichervorgang genau an dieser Stelle weiter gearbeitet-werden. Die Vorteile der vorangegangenen Umstrukturierung auf einen objektorientierten Ansatz zahlen sich hier aus. Die Projektverwaltung ist im Hauptfenster hinterlegt. Über die Getter und Setter des Identifiers und von AnalyDesign können so alle zu speichernden Variablen über die Getter ausgelesen und in die entsprechende Datei geschrieben werden. Beim Laden eines Projektes sind analog die Setter zur Übergabe der Variablen an die Objekte verantwortlich.

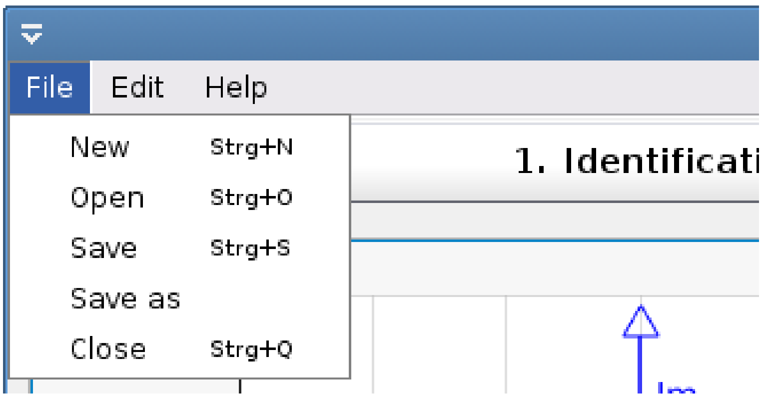


Abb. 2.6: Menü der Projektverwaltung

Die Speicherung der Oberfläche beschränkt sich dabei auf die Parameter „String", „Value", „Enable" und „Visible" der einzelnen graphischen Objekte. Die Graphen und Diagramme werden beim Laden eines Projektes neu erstellt. Mit den genannten vier Parametern ist jedoch garantiert, dass die Darstellung vor und nach dem Speichern identisch ist. Denn beim Laden eines Projekts werden im ersten Schritt diese vier Parameter an den entsprechenden Objekten geändert und anschließend die Update-

Funktionen der Oberfläche aufgerufen, sodass die Graphen und Diagramme neu geladen werden.

2.7 Erweiterung der Import-, Export- und Kopierfunktionen

Neben der Projektverwaltung sind zudem erweiterte Import-, Export- und Kopierfunktionen verfügbar. So ist es möglich jede Übertragungsfunktion des gesamten Projekts zu einer Anderen zu kopieren. Auch können die Übertragungsfunktionen im „tf"-Format in eine „mat"-Datei exportiert oder von einer entsprechenden Datei importiert werden. Das Menü ist in Abb. 2.7 dargestellt. Dadurch ist es möglich pzMove in jeder Regelungsaufgabe flexibel einzusetzen. Zu Beginn können die gegebenen Übertragungsfunktionen aus verschiedenen Systemen importiert und nach der Bearbeitung auch die gewünschten Funktionen exportiert werden. Die Importfunktion steht am Anfang der Bearbeitung einer Aufgabe. Im weiteren Bearbeitungsverlauf kann über die Projektverwaltung der aktuelle Bearbeitungsstand festgehalten und weitergegeben werden. Nach der Bearbeitung steht mit der Exportfunktion der abschließende Schritt im Einsatz von pzMove zur Verfügung.

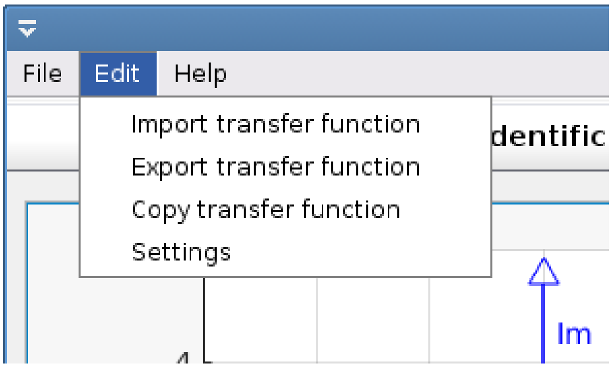


Abb. 2.7: Menü zum Importieren, Kopieren und Exportieren

2.8 Integration eines Quick-Start-Guides

Wie in Kap. 1.1 gezeigt, wird die Anwendung unter einem Interessenskonflikt programmiert. Um den Einstieg für Neuanwender zu erleichtern, steht in der Überarbeitung ein Quick-Start-Guide, zu sehen in Abb. 2.8, zur Verfügung. Dieser dient zur Anzeige eines einfachen Textdokuments, dessen Inhalt in kurzer, übersichtlicher Textform das Tool und dessen Anwendung auf Standardprobleme erklärt. Dadurch kann dem Anwender das Verhalten des Programms erklärt und vermeintliche Fehlfunktionen verhindert werden. Dies soll vor allem den Einstieg der Studenten erleichtern und so die Verbreitung im Lehrbereich fördern.

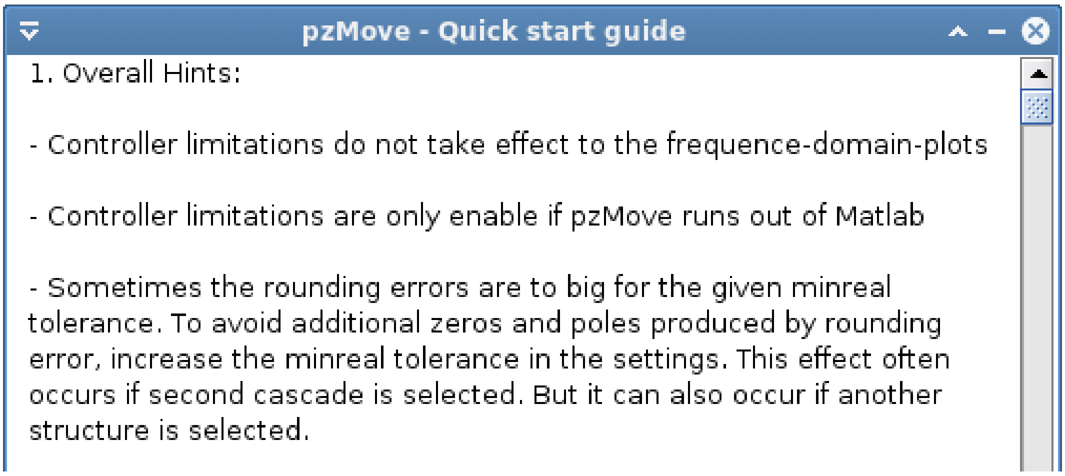


Abb. 2.8: Der neu integrierte Quick-Start-Guide

2.9 Verbesserung der Intuitivität

Die Intuitivität des Tools beruht vor allem auf der farblichen Trennung von Eingabe-, Ausgabe- und Administrativbereich. Im Identifier ist die GUI, wie in Abb. 2.9 zu sehen, in zwei Bereiche unterteilt. Der Eingabebereich umfasst das Laden und die Gewichtung der Messdaten, sowie die Definition der Vorgaben für die Berechnung der Übertragungsfunktion. Im Ausgabebereich ist die Übertragungsfunktion, sowie der Vergleich zwischen Messdaten und berechneter Übertragungsfunktion dargestellt. Es ist zudem möglich die berechnete Übertragungsfunktion zu ändern und deren Auswirkungen zu betrachten.

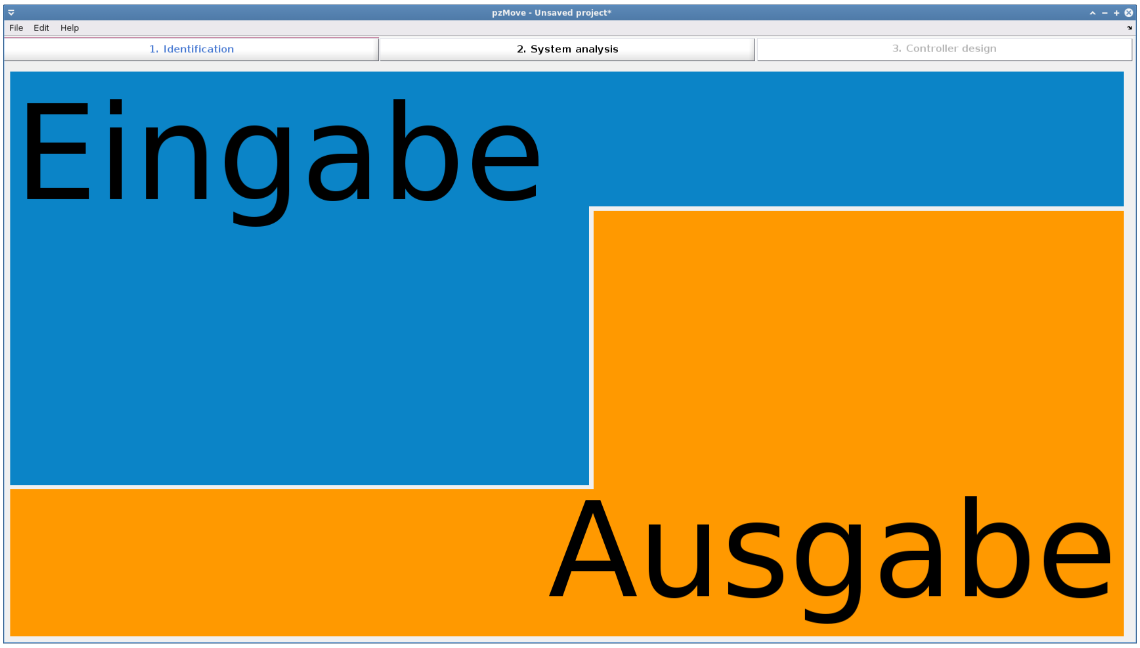


Abb. 2.9: Ein- und Ausgabebereiche des Identifiers

Der AnalyDesign-Teil ist, wie in Abb. 2.10 gezeigt, dreigeteilt. Im Eingabebereich können die verschiedenen Übertragungsfunktionen im Pole-Zero-Diagramm oder durch manuelle Eingabe in Textfelder eingegeben oder geändert werden. Der administrative Bereich ermöglicht die Auswahl verschiedener Regelungsstrukturen und die Auswahl einer Methode zur Berechnung der Reglerübertragungsfunktionen. Abhängig von der Auswahl der Ein- und Ausgangsvariablen sind im Ausgabebereich verschiedene zeitabhängige Diagramme, wie zum Beispiel die Sprungantwort oder die Impulsantwort, oder frequenzabhängige Diagramme, wie beispielsweise das Bode-Diagramm oder das Nyquist-Diagramm, darstellbar. Die Dreiteilung des AnalyDesign-Teils, die verschiedenen möglichen Regelungsstrukturen und die Möglichkeit zur Auswahl verschiedener Ein- und Ausgangsvariablen lassen nicht direkt erkennen, für welche Übertragungsfunktion die dargestellten Diagramme sind.

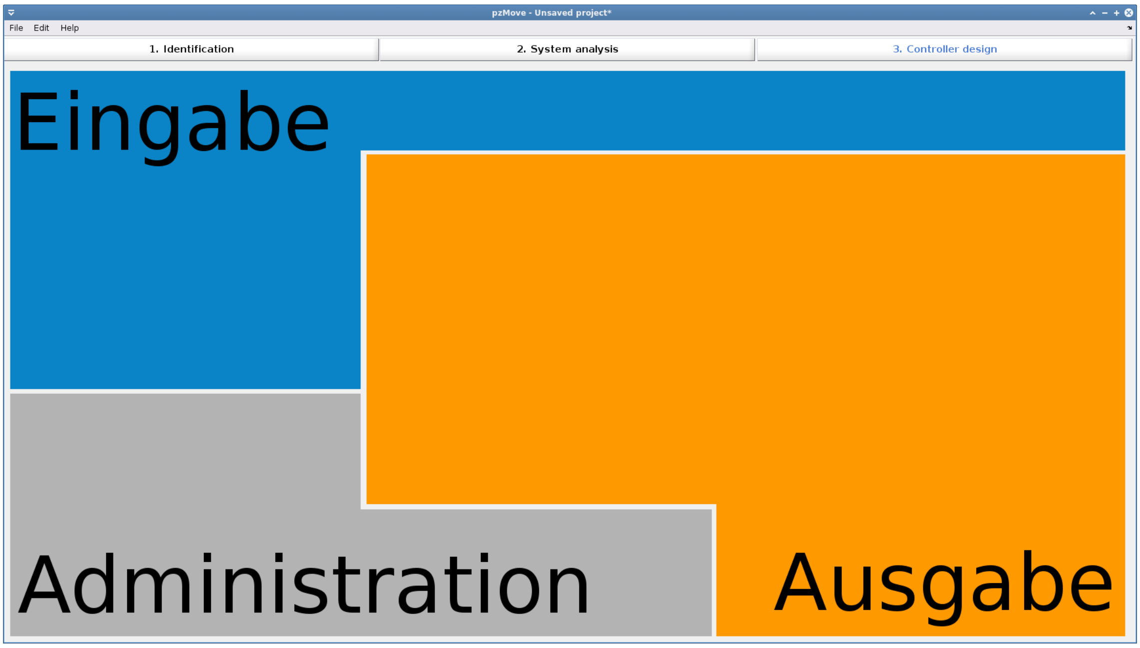


Abb. 2.10: Eingabe-, Administrations- und Ausgabebereich des AnalyDesign-Teils

Zur Beseitigung dieser Unklarheit wurde die Oberfläche im administrativen Bereich um einen dynamischen Signalplan erweitert. Mit dem Signalplan und den gewählten Ein- bzw. Ausgabegrößen ist eindeutig ersichtlich, welches System zur Berechnung der Diagramme Verwendung findet. Der Signalplan für den Standardregelkreis mit aktiver Störgrößenunterdrückung ist in Abb. 2.11 dargestellt. Der Signalplan wird aus mehreren Einzelbildern zusammengesetzt. Dies ist in Abb. 2.12 veranschaulicht. Dazu erfolgt eine Überlappung des Hintergrundbildes mit mehreren teiltransparenten Abbildungen, wodurch Teile des einfachen Regelkreises mit zusätzlichen Gliedern überdeckt werden. Ist die Funktion „hide if transfer function is equal 1“ aktiviert, werden nur regelkreiserweiternde Übertragungsfunktionen dargestellt, deren Übertragungsfunktion ungleich 1 ist. Das heißt, es werden alle Übertragungsfunktionen, die keinen Einfluss auf die Regelung haben, ausgeblendet. Unabhängig von der Übertragungsfunktion werden GR, GRC, GS, GSC, GM, GMC, GW und GX immer dargestellt.

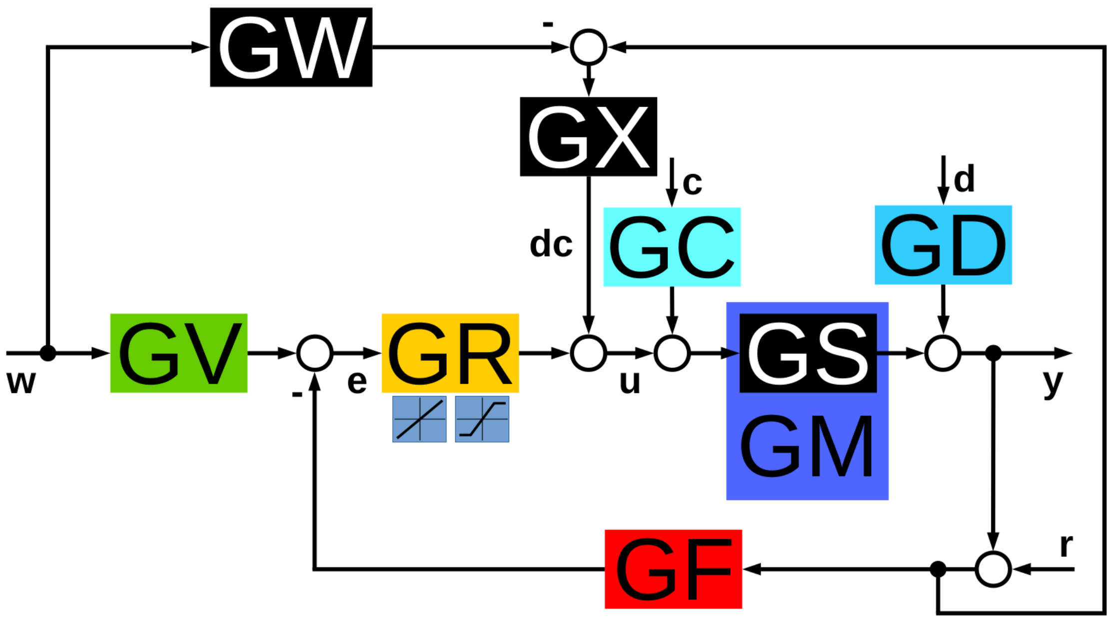


Abb. 2.11: Signalplan des Standardregelkreises mit aktiver Störgrößenunterdrückung

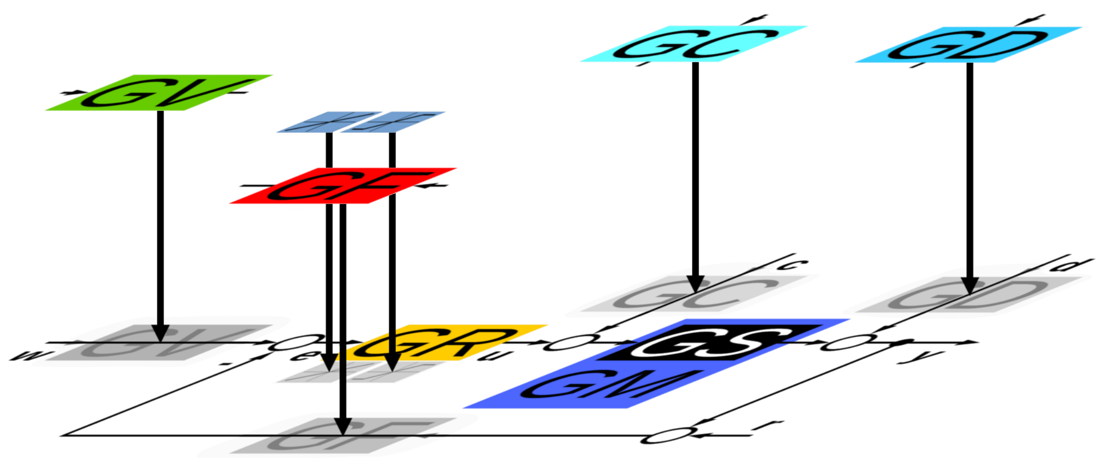


Abb. 2.12: Beispielhafter Aufbau des Signalplans des Standardregelkreises

3 Integration der Kaskadenregelung

Die Kaskadenregelung ist im Wesentlichen eine Verschachtelung von Standardregelkreisen. Die innere Kaskade bildet mit einer weiteren Strecke die zu regelnde Strecke für die nächst äußere Kaskade. Beispielhaft ist in Abb. 3.1 der Signalplan von pzMove für die Kaskadenregelung dargestellt. In dieser Abbildung ist eine Kaskadenregelung mit zwei Kaskaden als Regelungsstruktur vorgesehen. Es ist deutlich ersichtlich, dass sowohl die innere als auch die äußere Kaskade identisch mit einem Standardregelkreis sind. Fasst man die innere Kaskade, in dieser Darstellung grau hinterlegt, mit der äußeren Strecke „GSC“ zusammen, ergibt sich wieder der Standardregelkreis bestehend aus Vorfilter, Regler, Strecke und Rückführfilter. [1]

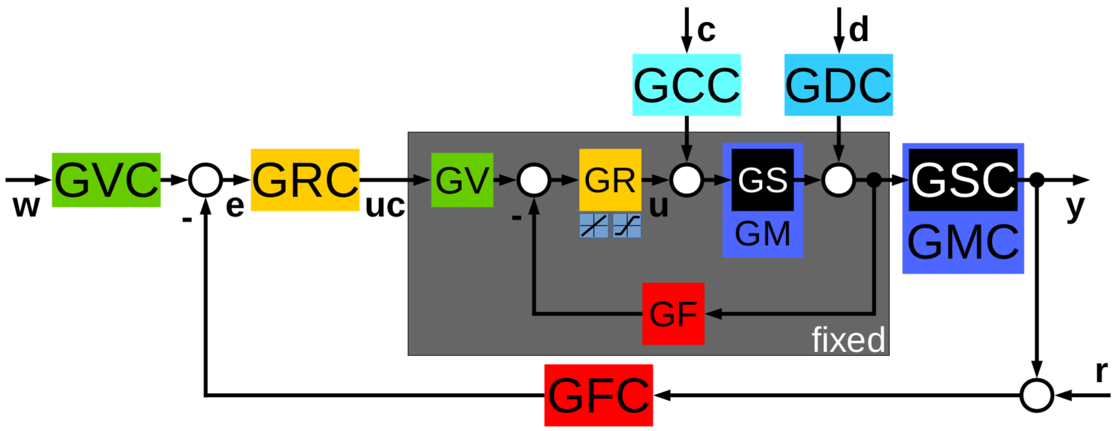


Abb. 3.1: Signalplan in pzMove für die Kaskadenregelung

3.1 Anwendungsmöglichkeiten der Kaskadenregelung

Der Signalplan der Kaskadenregelung wird standardmäßig ähnlich dem, in Abb. 3.2 gezeigten, Signalplan dargestellt. Das Problem bei der Erkennbarkeit der Anwendungsmöglichkeiten dieser Regelungsstruktur liegt in den beiden hintereinanderliegenden Regelstrecken.

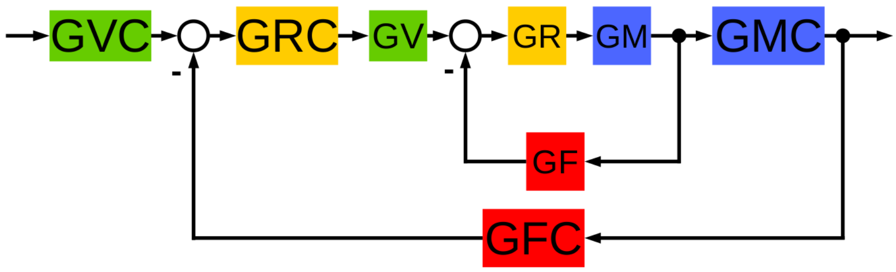


Abb. 3.2: Kaskadenregelung in Modelldarstellung

Möchte man die Einsatzmöglichkeiten dieser Regelungsstruktur erkennen, kann Abb. 3.3 helfen. In diesem Signalplan ist das zuammengefasste Gesamtsystem dargestellt.

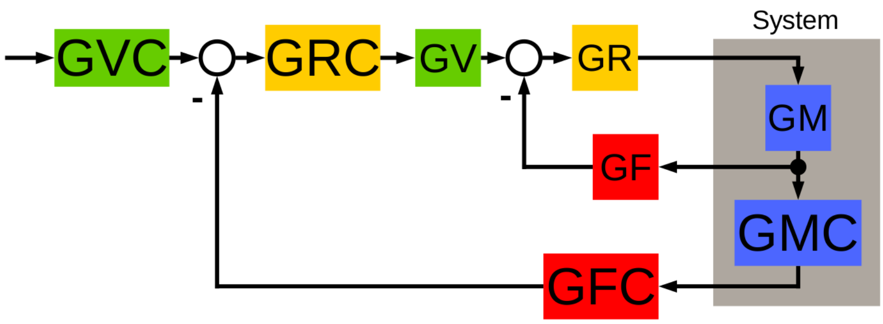


Abb. 3.3: Kaskadenregelung mit zusammengefasstem Gesamtsystem

Geht man noch einen Schritt weiter und wendet zudem die Black-Box-Darstellung an, ergibt sich Abb. 3.4. Bei der Betrachtung der Schnittstellen in der Black-Box-Darstellung fällt auf, dass die Kaskadenregelung immer Anwendung finden kann, wenn ein System eine Stellgröße und mehrere Rückführgrößen aufweist. Dabei ist für jede rückgeführte Größe ein eigener Regler notwendig. Zudem ist zwingend erforderlich, dass die nötige Ausregelgeschwindigkeit mit zunehmender Kaskadennummer abnimmt.

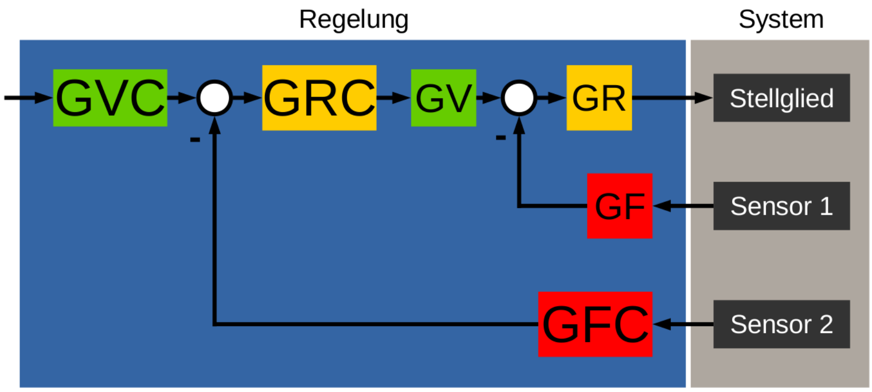


Abb. 3.4: Kaskadenregelung mit dem Gesamtsystem in Black-Box-Darstellung

Liegt ein derartiges System in der Praxis vor, kann mit der Herleitung der Modelle begonnen werden. Dabei sind zu Beginn die Rückführgrößen anhand der Regelgeschwindigkeiten zu sortieren und dadurch die genaue Regelstruktur fest vorzugeben. Anhand dieser Struktur ist eine zielstrebige Herleitung der Modelle möglich.

Die vorangegangene Betrachtung der Anwendungsmöglichkeiten beruht darauf, dass einem Anwender ein fixes System vorgegeben wird und er anhand diesem eine Regelung realisieren muss. Nachfolgend soll die Betrachtung der Anwendungsfälle von einer komplett anderen Sichtweise erfolgen. Ist ein variables System gegeben, bei dem Funktion und Funktionsstruktur zum Beispiel aus der Entwicklung einer Maschine vorgegeben sind und hat dieses System nur eine Stellgröße, kann sofort die Anwendung einer Kaskadenregelung in Betracht gezogen werden. Die weitere Problemstellung betrifft dabei vor allem die Auswahl der Sensoren und deren Platzierung zur Erfassung der Rückführgrößen. Die Auswahl und Position der Rückführgrößen sollte folgenden Hinweisen folgen:

- Es muss jede Systemgröße zurückgeführt werden, wenn sie durch das Stellglied beeinflussbar ist und in der Regelung zu berücksichtigen ist.

- Eine Rückführung einer Hilfsgröße ist nicht sinnvoll, wenn diese eindeutig aus einer anderen Rückführgröße berechenbar ist. Bei dem nachfolgenden Beispiel des Segways® ist die Position dessen nicht direkt aus dem Pendelwinkel berechenbar. Somit ist es nötig, die Motordrehposition als Hilfsgröße rückzuführen.

- Es ist auch möglich Systemgrößen zurückzuführen, wenn diese nicht mit einem Sensor erfassbar sind, vorrausgesetzt die Größe kann mit einem ausreichend gutem Modell aus den anderen Rückführgröße geschätzt werden.

Vor allem der letzte Hinweis stellt einen sehr interessanten Aspekt beim Einsatz einer Kaskadenregelung dar. Punkt zwei und drei der Hinweise mögen sich auf dem ersten Blick widersprechen. Bei genauerer Betrachtung liegt der Unterschied bei „eindeutig berechenbar“ und „geschätzt“. Ist keine genaue Berechnung der Rückführgröße möglich, jedoch ein Modell zur Schätzung aus einer anderen Rückführgröße berechenbar, kann die Rückführgröße mit einem gewissen Fehler berechnet werden. Dadurch kann eventuell eine ausreichend genaue Regelung dieser Größe erfolgen. Anwendungsfälle können sein:

- Die Rückführgröße kann mit keinem Sensor erfasst werden.

- Die Kosteneinsparung für den zusätzlichen Sensor überwiegt die fehlende Genauigkeit.

- Die Reduktion der Komplexität des technischen Systems durch den wegfallenden Sensor führt zu keiner Beeinträchtigung der Funktionsfähigkeit durch den Schätzfehler.

Betrachtet man den Signalplan einer Kaskadenregelung mit zwei Kaskaden und einer Rückführgrößenschätzung aus Abb. 3.5 fällt auf, dass das System eine Stellgröße und eine Rückführgröße hat und somit eigentlich als Standardregelkreis darstellbar ist. Der Vorteil der Anwendung der Kaskadenregelung in diesem Fall liegt im systematischen Auslegen der Einzelregler. Das Schritt-für-Schritt-Vorgehen der Kaskadenregelung mit Rückführgrößendefinition, Sortierung, Regelungsstrukturfestlegung und anschließender Reglerauslegung ermöglicht auch bei der Rückführgrößenschätzung einen einfachen Reglerentwurf unter systematischer Berücksichtigung der Einzelgrößen eines Systems.

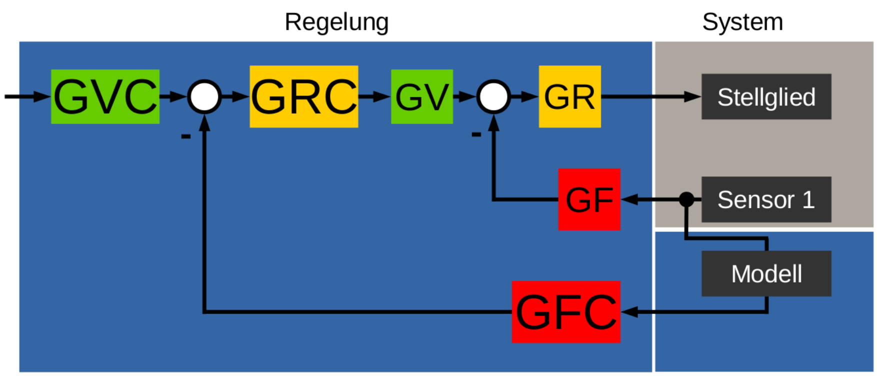


Abb. 3.5: Zweischleifige Kaskadenregelung mit Rückführgrößenschätzung

3.2 Vorgehen bei der Auslegung einer Kaskadenregelung

Da die Kaskadenregelung aus mehreren Standardregelkreisen zusammengesetzt ist, wird jeder Regler für sich, wie bei einem Standardregelkreis ausgelegt. Allgemein wird dabei folgendes Vorgehen in der einschlägigen Literatur herangezogen: [8, S. 580+581]

- Auslegen des Reglers der inneren Kaskade mit Augenmerk auf schnelle Kompensation von Störungen und gutes Führungsübertragungsverhalten.
- Zusammenfassung der inneren Kaskade mit der Strecke der äußeren Kaskade zur Gesamtstrecke der äußeren Kaskade.
- Auslegung des Reglers der äußeren Kaskade

Ist die innere Kaskade „schneller“ als die äußere Kaskade, wodurch die Regelabweichung der inneren Kaskade bei einer Stellgrößenvorgabe durch die äußere Kaskade vernachlässigbar klein ist, und können die auftretenden Störungen ohne Auswirkungen auf die äußere Kaskade von der Inneren ausgeregelt werden, ist die innere

Kaskade als „1“ anzunehmen und hat somit keinen Einfluss auf die Auslegung des Reglers der äußeren Kaskade.

Die Anwendung dieses Verfahrens mit Auslegen, Vereinfachen und weiterem Auslegen ist sehr einfach anwendbar und ermöglicht die Anwendung von einfachen Standardverfahren. Ein wesentlicher Aspekt fällt dabei außer Acht. Wie in Abschnitt 3.1 gezeigt, ist die Kaskadenregelung ein einfacher Regelkreis mit zusätzlicher Rückführung von Hilfsgrößen. Aus mehreren Messgrößen ergibt sich eine Stellgröße, wobei die Stellgröße der Regler der innersten Kaskade vorgibt. Vereinfacht man die innere Kaskade bei der Auslegung der äußeren Kaskade, kann die reale Stellgröße nicht mehr betrachtet werden. Dies kann dazu führen, dass erst bei der Umsetzung der Regelung in der Realität auffällt, dass die Stellwerte außerhalb der Grenzen des Stellglieds liegen.

3.3 Integration der Kaskadenregelung in pzMove

In pzMove ist die Kaskadenregelung für eine Regelung mit zwei Kaskaden und somit mit nur einer Hilfsgröße möglich. Als weitere Beschränkung sind nur zwei Stellen zur Störgrößenaufschaltung vorgesehen. Neben der Störung auf der realen Stellgröße ist zusätzlich eine Störung auf die erste Regelstrecke aufschaltbar. Eine Störung auf die Hilfsgröße kann nicht simuliert werden. Bei der Integration der Kaskadenregelung in pzMove sind zwei wesentliche Punkte berücksichtigt. Prinzipiell ist das in Abschnitt 3.2 gezeigte Vorgehen und eine Betrachtung der Stellgröße möglich. Zu Beginn der Auslegung einer Kaskadenregelung sind alle benötigten Übertragungsfunktionen, im Speziellen die Strecke und das Modell der inneren (GS und GM) und der äußeren (GSC und GMC) Kaskade zu definieren. Anschließend wechselt man von „System analysis“ in „Controller design“ und wählt im administrativen Bereich unter „Signal plan“ im Popupmenü „standard control loop“ für die innere Kaskade aus. Dadurch kann der Regler der inneren Kaskade ausgelegt werden. Stellt man unten rechts im Bereich „Output“ die Ausgangsgröße auf die Stellgröße „u“, sind die Auswirkungen einer Führungsgrößen- oder Störgrößenänderung auf die Stellgröße ersichtlich. Der zweite Schritt, die Vereinfachung, erfolgt in pzMove im Hintergrund und ist nicht vom Anwender durchzuführen. Zur Auslegung der äußeren Kaskade ist im Administrationsbereich die Regelungsstruktur von „standard control loop“ auf „second cascade“ umzustellen. In Abb. 3.6 ist beispielhaft die Auswahl im Standardprojekt dargestellt. Die innere Kaskade ist gesperrt und es sind nur noch Änderungen an der äußeren Kaskade möglich. Auch die Reglerauslegungsverfahren im grauen Bereich im Panel „Controller design“ beeinflussen nicht mehr den Regler GR, den Vorfilter GV und den Rückführfilter GF, sondern deren Äquivalente in der äußeren Kaskade GRC, GVC und GFC. Die eigentliche Reglerauslegung erfolgt in identischer Weise wie bei der inneren Kaskade. Auch in der zweiten Kaskade ist durch Auswahl der Stellgröße „u“ im Panel „Output“ diese auf die Auswirkungen von Führungs- und Störgrößenänderungen untersuchbar. Für den Anwender bringt somit die Umsetzung in pzMove zwei Vorteile: Zum einen entfällt der Schritt der Vereinfachung und zum anderen ist auch bei der äußeren Kaskade die Betrachtung der eigentlichen Stellgröße möglich.

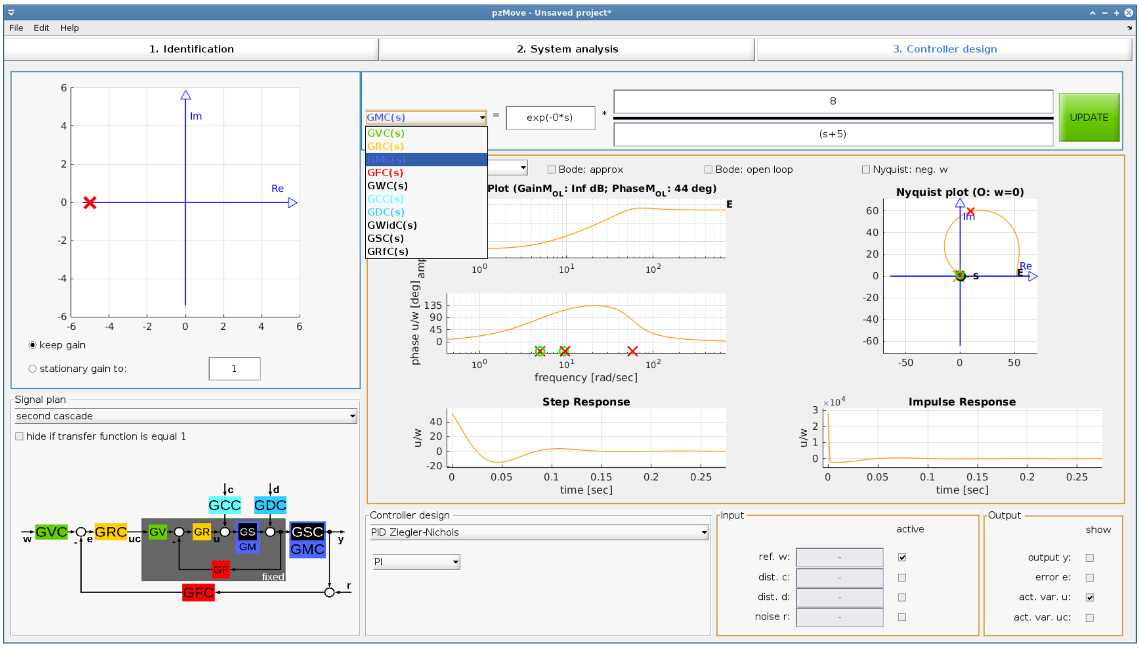


Abb. 3.6: Auslegung der äußeren Kaskade unter Betrachtung der realen Stellgröße

4 Integration des Loop-Shaping-Verfahrens

Das Loop-Shaping-Verfahren bezeichnet die Reglersynthese aus verschiedenen Korrekturgliedern. Anwendung findet das Nyquist-Kriterium [2, S. 189], welches in Gl. 4.1 dargestellt ist. W_+ bezeichnet dabei den im Gegenuhrzeigersinn, abzüglich den im Uhrzeigersinn, überstrichenen Winkel der Verbindungslinie zwischen der Nyquist-Kurve und dem kritischen Punkt P(-1,0) um diesen Punkt bei der Erhöhung der Frequenz von null bis plus Unendlich. r_0 stellt die Anzahl der Pole in der RHP und q die Anzahl der Pole auf der imaginären Achse dar. Anwendung findet dieses Kriterium im offenen Regelkreis. Das heißt, dass Gl. 4.1 für das Produkt aus Regler und Strecke zutreffen muss.

$$W_+ = r_0 \cdot \pi + q\frac{\pi}{2} \qquad \text{(Gl. 4.1)}$$

Das Nyquist-Kriterium dient bei der Anwendung des Loop-Shaping-Verfahrens zum Feststellen der Stabilität des Regelsystems. Des Weiteren werden in der einschlägigen Literatur auch diverse Korrekturglieder angegeben. Ein Anwendungsschema zum erfolgreichen Einsatz des Verfahrens ist nicht zu finden. Nachfolgend soll ein möglichst praktikables Vorgehen bei der Anwendung des Loop-Shaping-Verfahrens erarbeitet werden. In dieser Arbeit sind zwei Verfahrensmöglichkeiten dargestellt, die nachfolgend „Kriteriumsherleitung“ und „Schleifenoptimierung“ benannt sind. Ein Vorbericht [1] befasst sich bereits näher mit dem erweiterten Kreis der möglichen Korrekturglieder und dem Verfahren der Kriteriumsherleitung. Nachfolgend ist eine genauere Betrachtung der Anwendungsmöglichkeiten beider Vorgehensweisen dargestellt. Auch sind die Bedeutungen der einzelnen Korrekturglieder speziell für die beiden Verfahren aufgezeigt.

Die Angaben zum Nyquist-Diagramm beziehen sich immer auf den positiven Ast der Nyquist-Kurve. Der Anfangsbereich der Nyquist-Kurve kennzeichnet nachfolgend den Bereich der kleinen Frequenzen. Der Endbereich soll den Bereich der hohen Frequenzen bezeichnen.

4.1 Lehransatz: Kriteriumsherleitung

Die Kriteriumsherleitung bezeichnet ein striktes Vorgehen mit festem Verfahrensablauf und wenig Unsicherheiten. Dazu sind die geforderten Schritte nachfolgend beispielhaft an der Strecke aus Gl. 4.2 durchgeführt.

$$G(s) = \frac{1}{(s-1)(s-2)} \qquad \text{(Gl. 4.2)}$$

1. Berechnung der Pole der Strecke: Die Berechnung der Polstellen der Strecke erfolgt durch Nullsetzten des Nenners. Hier ergeben sich zwei Pole mit positivem Realteil bei P1(1,0) und bei P2(2,0). Zu beachten ist an dieser Stelle vor allem, dass eventuell eine Kürzung von Polen und Nullstellen möglich ist. Dies muss vor der Berechnung der Polstellen erfolgen. Hat die Strecke eine ungerade Anzahl an instabilen Polen, kann später mit den üblichen Korrekturgliedern kein stabiles Verhalten im geschlossenen Regelkreis erzeugt werden. Aus diesem Grund kann bei entsprechenden Strecken die Anzahl der instabilen Pole mit dem Korrekturglied „Zusätzlicher Pol in der RHP“ um eins erhöht werden.

2. Berechnung des Umschlingungswinkels anhand von Gl. 4.1: Das verwendete Beispiel hat zwei Pole in der rechten Halbebene, wodurch $r_0 = 2$ gilt, und keine Pole auf der imaginären Achse, weshalb $q = 0$ ist. Der berechnete Umschlingungswinkel um den Punkt P(-1,0) beträgt somit 2π oder 360 Grad gegen den Uhrzeigersinn.

3. Aufstellen der Kriterien für das Nyquist-Diagramm anhand des berechneten Umschlingungswinkels: Der End-Umschlingungswinkel beträgt immer 0 Grad, da

der kritische Punkt, auf den sich der Umschlingungswinkel bezieht, genau links vom Ursprung liegt und die Nyquist-Kurve aufgrund der abfallenden Amplitude für hohe Frequenzen immer im Ursprung endet. Damit die vollständige Umschlingung erfüllt wird, muss die Nyquist-Kurve entweder im Ursprung oder auf der positiven realen Achse beginnen. Der geschlossene Regelkreis soll stationär genau sein, weshalb die offene Kette integrierendes Verhalten haben muss. Dies ist der Fall, wenn die Start-Phase - 90 Grad beträgt. Dazu ist ein Pol im Ursprung nötig, der die Nyquist-Kurve im negativem, imaginären Unendlichen beginnen lässt. Der zusätzliche Pol im Ursprung erhöht die vom Nyquist-Kriterium geforderte Umschlingung um $0,5\pi$. Durch die Phasenabsenkung um 90 Grad wird diese Forderung erfüllt, weshalb nach dem Nyquist-Kriterium weiterhin Stabilität gegeben ist. Diese Bedingung ist in Abb. 4.1 mit K1 gekennzeichnet. Das zweite ableitbare Kriterium für die Umschlingung ist der Schnittpunkt der Kurve mit der realen Achse links vom kritischen Punkt, wie K2 zeigt. Sollte ein weiterer Schnittpunkt zwischen der realen Achse und der Nyquist-Kurve vorliegen, muss dieser rechts vom kritischen Punkt liegen. Das letzte Kriterium ist mit K3 gekennzeichnet.

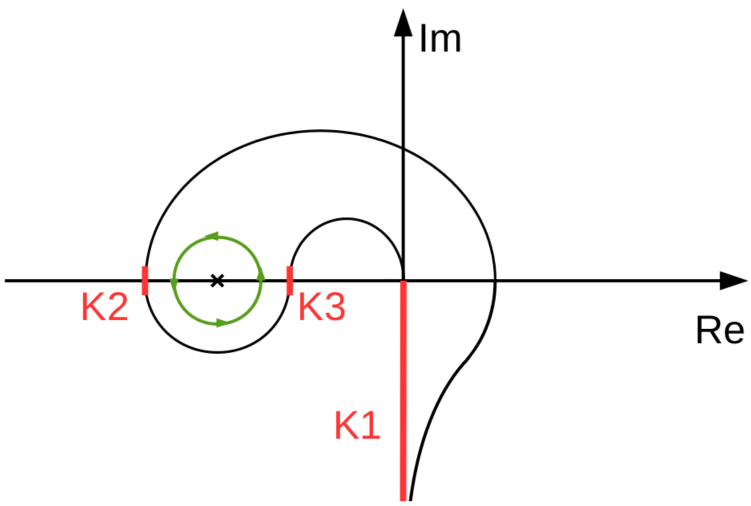


Abb. 4.1: Ableitung einfacher Kriterien für das Nyquist-Diagramm [1]

4. Übertrag der Kriterien des Nyquist-Diagramms in das Bode-Diagramm: Das erste Kriterium fordert eine Phase von - 90 Grad für sehr kleine Frequenzen. Aus dem zweiten Kriterium ist zu schließen, dass die Amplitude bei einer Phase von 180 Grad und einem positiven Gradienten über 1 und somit über 0 Dezibel liegen muss. Die dritte Bedingung führt zu einer Amplitude unter 0 Dezibel an dem Punkt, an dem der Phasenwinkel wieder unter 180 Grad fällt. In Abb. 4.2 sind die Kriterien in das Bode-Diagramm der Strecke eingezeichnet. Ein möglicher Verlauf der diese Kriterien erfüllt, ist in blau dargestellt.

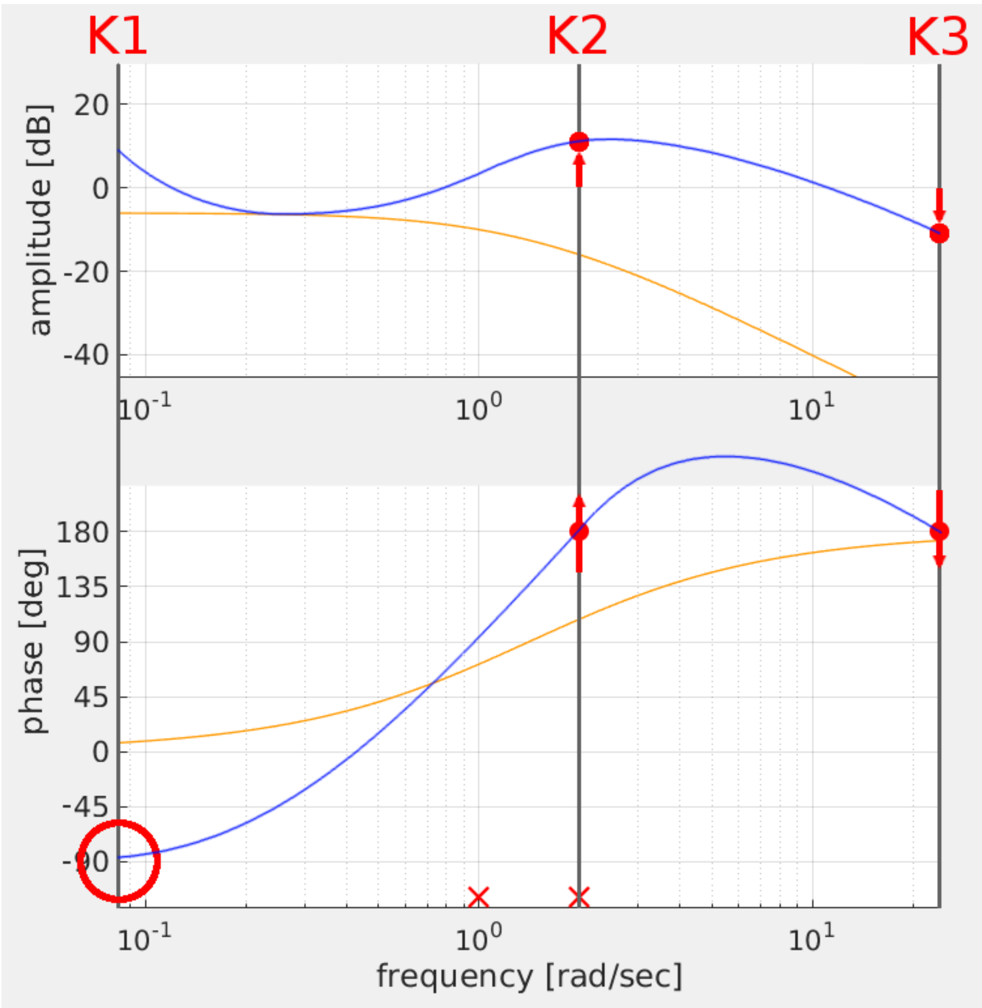


Abb. 4.2: Übernahme der Kriterien vom Nyquist- in das Bode-Diagramm [1]

5. Berechnung der geforderten Verläufe des Bode-Diagramms des Reglers: Das Bode-Diagramm des Reglers bildet die Differenz der geforderten Verläufe und der Verläufe der Strecke. Aufgrund der Darstellung der Amplitude in Dezibel ist dies genau die Fläche zwischen den beiden Kurven. Das berechnete Bode-Diagramm des Reglers ist in Abb. 4.3 dargestellt.

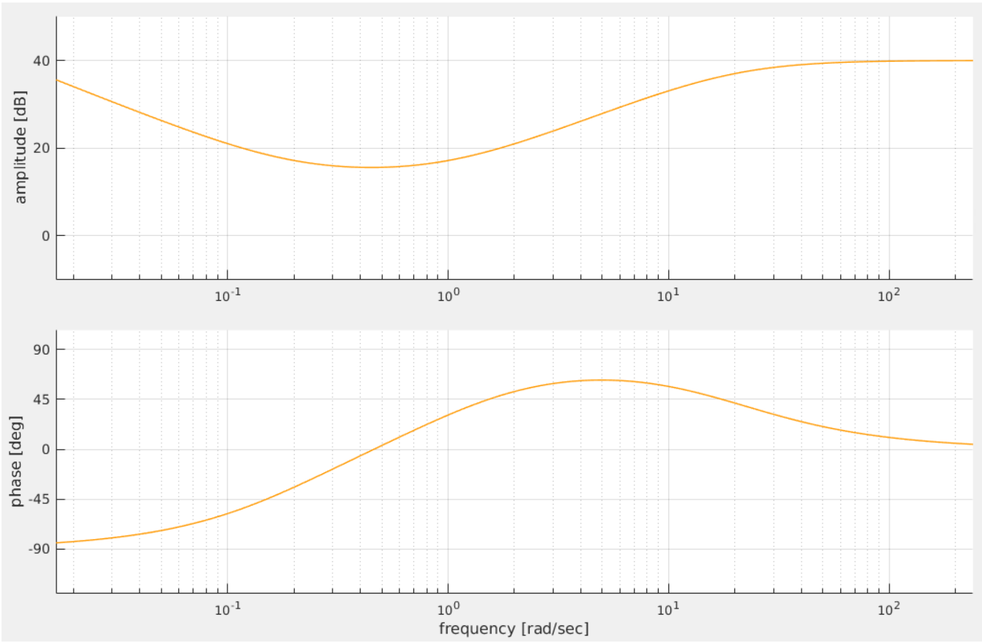


Abb. 4.3: Berechnetes Bode-Diagramm für einen stabilen Regler

6. Zusammenstellung der Korrekturglieder anhand des berechneten Bode-Diagramms: Betrachtet man das Bode-Diagramm des Reglers, sind zwei Hauptmerkmale ableitbar. Zum Einen muss der Regler die Phase im Bereich um 2 rad/s und zugleich die Amplitude rechts von dieser Grenze anheben. Das Lead-Glied aus Abschnitt 4.3.1 erzeugt genau dieses Verhalten. Die Phasenabsenkung für sehr kleine Frequenzen wird hingegen vom Lead-Glied nicht erfüllt. Somit ist ein zweites Korrekturglied anzuwenden. Das PI-Glied verursacht neben der Amplitudenanhebung auch eine Phasenabsenkung für niedrige Frequenzen. Wendet man ein Lead-Glied mit den Grenzfrequenzen bei 0,2 rad/s und 20 rad/s und zusätzlich ein PI-Glied mit der Eckfrequenz von 1 rad/s an, ergibt sich das Nyquist-Diagramm wie in Abb. 4.4 ersichtlich. Die Übertragungsfunktion des re-

sultierenden Reglers ist in Gl. 4.3 dargestellt. Mit einem Umschlingungswinkel von 470 Grad erfüllt diese Nyquist-Kurve das Nyquist-Kriterium, wodurch der geschlossene Regelkreis eingangs-ausgangs-stabil ist. Zusätzlich beginnt die Nyquist-Kurve bei -90 Grad Phase, wodurch stationäre Genauigkeit gegeben ist.

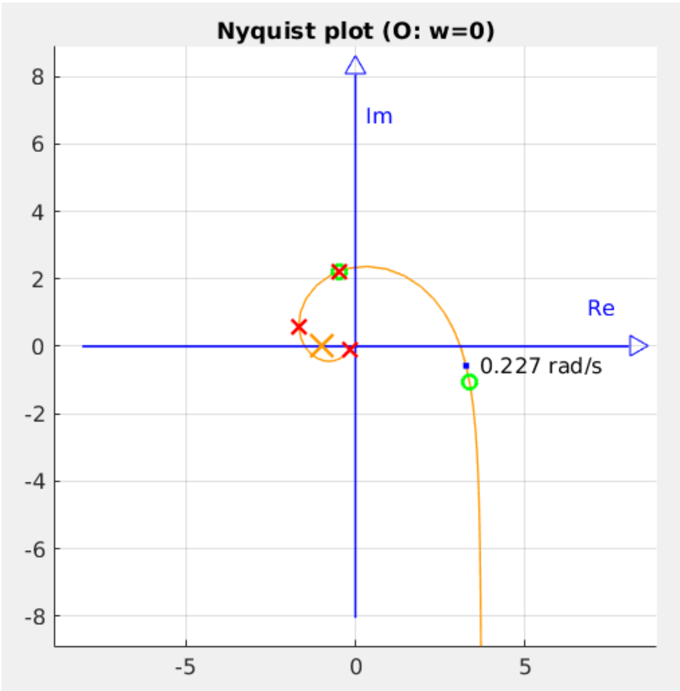


Abb. 4.4: Nyquist-Diagramm des offenen Regelkreises

$$R = \frac{\frac{s}{0,2} + 1}{\frac{s}{20} + 1} \cdot \left(1 + \frac{1}{s}\right) \qquad \text{(Gl. 4.3)}$$

Dieses Verfahren ist vor allem für die Lehre gut einsetzbar, da die Zusammenhänge zwischen Nyquist-Diagramm, Bode-Diagramm, Nyquist-Kriterium und den Zusammenbau verschiedener Korrekturglieder Schritt-für-Schritt dargelegt sind. Der Ansatz ist einfach anzuwenden, da jeder Schritt ohne Verständnis des Gesamtvorhabens durchführbar ist. Lediglich die Zusammenstellung der Korrekturglieder bedarf einer genauen Kenntnis der Bode-Diagramme der einzelnen Glieder. Hingegen ist auch bei diesem Schritt nur die Betrachtung der Bode-Diagramme der Korrekturglieder nötig. Die Anwendung des Ansatzes zur Kriteriumsherleitung hat zwei entscheidende Nachteile: Durch den strikten Ablauf und die Trennung der Aufgaben in einzelne, unabhängige Schritte, ist die Anwendung sehr zeitaufwändig. Als Ziel verfolgt die Kriteriumsherleitung vor allem die Erstellung eines stabilen Reglers, ohne dabei die späteren Regeleigenschaften in Betracht zu ziehen. Die Optimierung des Reglers bedarf der Anpassung der Parameter der Korrekturglieder. Dies ist aber mit diesem Verfahren kaum möglich.

4.2 Anwendungsansatz: Schleifenoptimierung

Die Schleifenoptimierung ist auf den ersten Blick komplexer in der Anwendung und erweckt den Eindruck einer „Spielerei“. Die effektive Anwendung dieses Verfahrens bedarf der Betrachtung der gesamten Auswirkungen der Korrekturglieder im Nyquist-Diagramm und etwas Erfahrung. Die nachfolgende Darstellung liefert einen Überblick über die Schleifenoptimierung anhand des Beispiels aus Gl. 4.2. Für das oben gezeigte Verfahren sind die Bode-Diagramme der Korrekturglieder von Bedeutung. Die Schleifenoptimierung betrachtet nur das Nyquist-Diagramm und die ungefähre Auswirkung der Korrekturglieder auf die Nyquist-Kurve. Der Ausdruck „ungefähr“ beschreibt das Problem des Verfahrens. Dazu sind jedoch in den Darstellungen der Korrekturglieder detaillierte Hinweise, speziell für die Anwendung der Glieder in der Schleifenoptimierung, gegeben. Auch hier gibt es ein festes Vorgehen, das in den ersten Schritten identisch mit dem Verfahren der Kriteriumsherleitung ist. Die ersten beiden Schritte sind in Abschnitt 4.1 genau beschrieben.

1. Berechnung der Pole der Strecke.

2. Berechnung des Umschlingungswinkels anhand von Gl. 4.1

3. Erzeugen der geforderten Umschlingungen um einen beliebigen Punkt auf der negativen, realen Achse: Bei der Schleifenoptimierung wird nur das Nyquist-Diagramm betrachtet. Die gegebene Kurve im Nyquist-Diagramm muss die vom Nyquist-Kriterium geforderten Umschlingungen erfüllen. Im gegebenen Fall

muss dazu die in Abb. 4.5 gezeigte Nyquist-Kurve verändert werden, um die geforderte Umschlingung zu erfüllen. Ein Teil der Kurve muss gegen den Uhrzeigersinn um den Ursprung gedreht werden, sodass die Kurve einen Punkt auf der negativen, realen Achse umschlingt. Grundsätzlich kommen zum Bilden der Umschlingungen nur das Lead- und das Lag-Glied zum Einsatz. Da die Kurve gegen den Uhrzeigersinn gedreht werden soll, ist auf ein Lead-Glied zurückzugreifen. Betrachtet man die Frequenzen im Nyquist-Diagramm, so ist festzustellen, dass die Nyquist-Kurve im Bereich zwischen 0,1 rad/s und etwa 10 rad/s am Deutlichsten ausgeprägt ist. Aufgrund dessen sind die Grenzfrequenzen mit diesen Werten zu wählen.

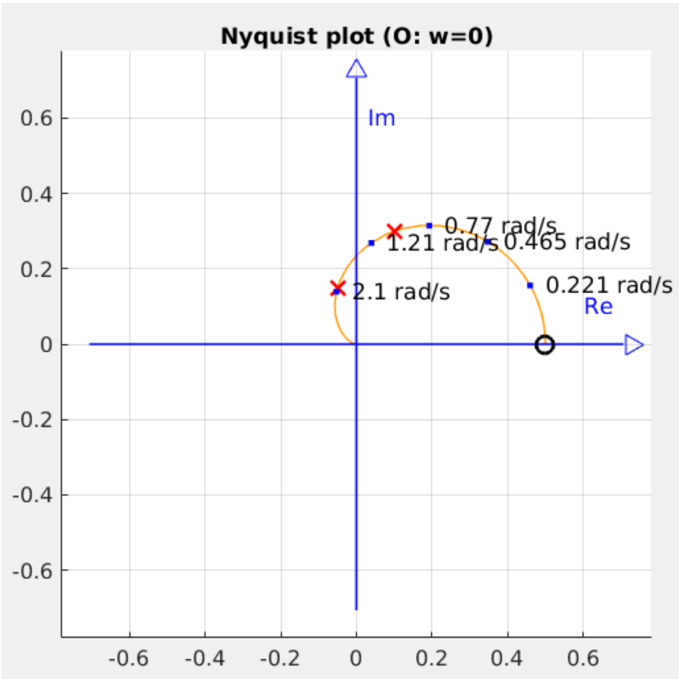


Abb. 4.5: Nyquist-Diagramm der instabilen Strecke

Fügt man ein Lead-Glied mit $\omega_{grenz\ 1} = 0{,}1$ rad/s und $\omega_{grenz\ 2} = 10$ rad/s ein, ergibt sich die Nyquist-Kurve wie in Abb. 4.6. In der zweiten Hälfte des Nyquist-Verlaufs wird eine „Blase“ gebildet und diese gegen den Uhrzeigersinn gedreht. So erhält man eine Schlaufe um den Punkt P(-2,0) auf der negativen, realen Achse. In diesem speziellen Fall ist auch der kritische Punkt bereits in der

Schlaufe, sodass die Stabilität bereits jetzt gegeben ist.

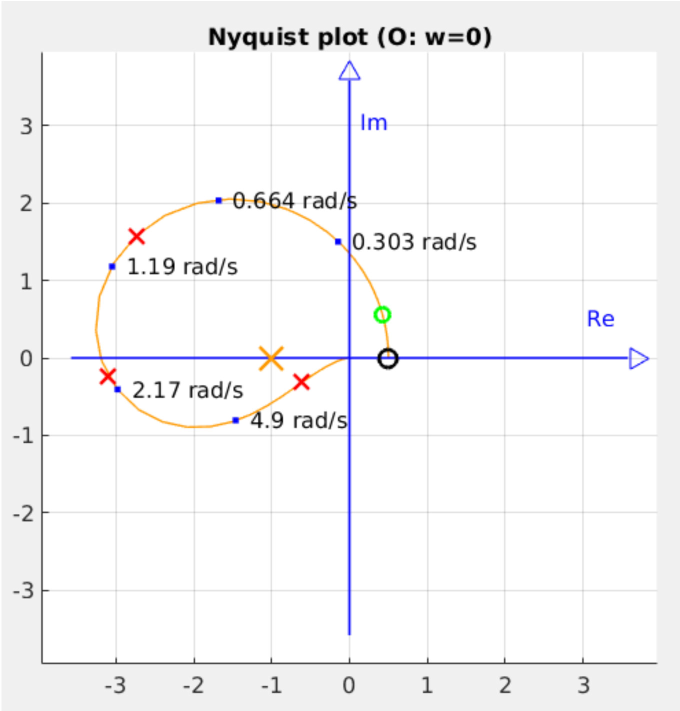


Abb. 4.6: Nyquist-Kurve des offenen Regelkreises mit Lead-Glied

4. Anpassung der Verstärkung: Wie im vorangegangenen Schritt erklärt, ist der geschlossene Regelkreis mit einem Lead-Glied bereits stabil. Durch eine P-Verstärkung kann die Kurve proportional zum Ursprung skaliert werden. In diesem Beispiel ist mit einer Verstärkung von 0,5 die Kurve so zu verkleinern, dass die Kurve einen maximalen Abstand zum kritischen Punkt erreicht. Die Nyquist-Kurve mit der zusätzlichen Verstärkung ist in Abb. 4.7 dargestellt. Bereits jetzt fällt auf, dass diese Optimierung mit der zusätzlichen Verstärkung mit dem Ansatz der Kriteriumsherleitung nicht möglich ist.

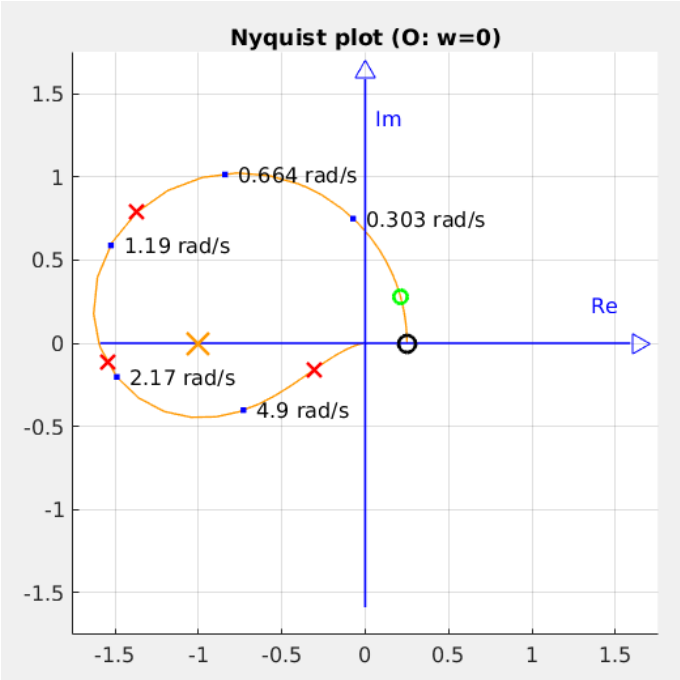


Abb. 4.7: Nyquist-Kurve des offenen Regelkreises mit zusätzlichem Verstärkungsglied

5. Einfügen des integrierenden Anteils: In den vorhergehenden Schritten wurde die Stabilität des geschlossenen Regelkreises erzeugt. In diesem Schritt sind ein oder mehrere PI-Glieder einzufügen. Damit der geschlossene Regelkreis stationär genau ist, muss die Kurve im negativen, imaginären Unendlichen beginnen. Ein PI-Glied dreht den Startpunkt der Nyquist-Kurve um 90 Grad im Uhrzeigersinn. Beginnt die Kurve auf der positiven, realen Achse, ist ein PI-Glied nötig. Liegt der Startpunkt auf der positiven, imginären Achse, sind zwei Korrekturglieder mit integrierendem Anteil nötig. Zu beachten ist hier, dass die obige Erklärung für Strecken gilt, die für sehr kleine Freqenzen eine Phase zwischen - 90 Grad und + 90 Grad haben. Die Anzahl der PI-Glieder lässt sich auch einfach berechnen. Dazu ist zu der Differenz aus der Anzahl der Nullstellen der Strecke bei $0 + 0j$ und der Anzahl der Pole der Strecke bei $0 + 0j$ 1 zu addieren. Die in Abb. 4.8 gezeigte Sprungantwort des geschlossenen Kreises zeigt, dass Stabilität gegeben ist, jedoch der Regelkreis nicht stationär genau ist. In diesem

Beispiel liegt der Startpunkt auf der positiven, realen Achse, wodurch genau ein PI-Glied nötig ist.

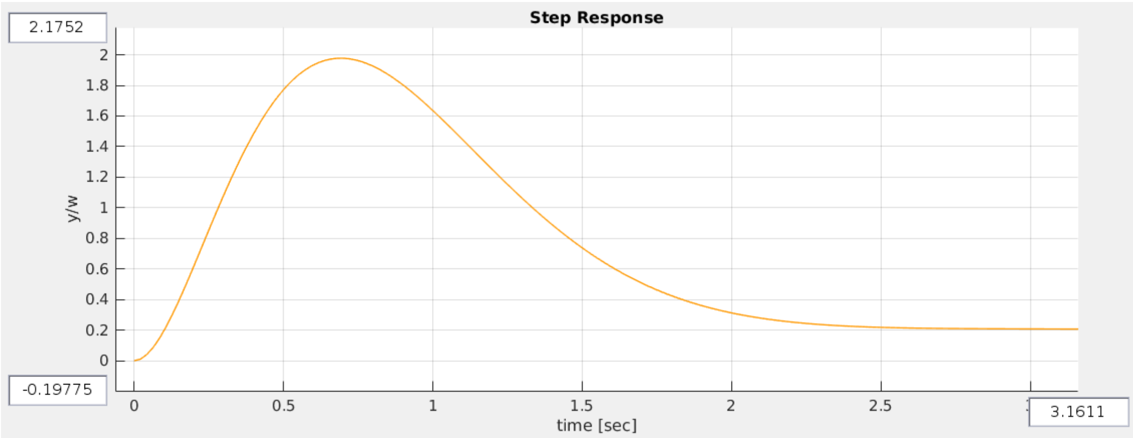


Abb. 4.8: Sprungantwort des stabilen Regelkreises

Da das PI-Glied den Anfangsbereich der Nyquist-Kurve, wie in Abb. 4.22 gezeigt, um 90 Grad im Uhrzeigersinn um den Ursprung dreht, darf die Eckfrequenz ω des PI-Glieds, wie in der zweiten Darstellung in Gl. 4.12 gezeigt, nicht zu hoch gewählt werden. Einfach erklärt, zieht das PI-Glied die Kurve nach rechts, wodurch bei zu hoch gewählter Frequenz die Umschlingung und somit die Stabilität aufgehoben wird. In diesem Beispiel ist die Grenzfrequenz auf maximal $\omega = 2$ rad/s zu setzen. Das resultierende Nyquist-Diagramm ist in Abb. 4.9 gezeigt.

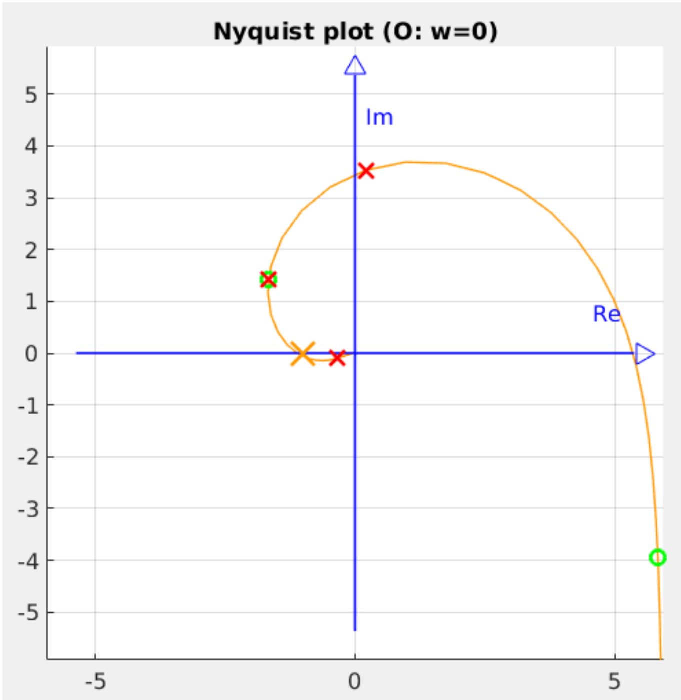


Abb. 4.9: Nyquist-Kurve des offenen Regelkreises mit zusätzlichem PI-Glied

6. Optimierung der Verstärkung: Wie in Abb. 4.9 gezeigt, ist der Abstand der Nyquist-Kurve zum kritischen Punkt durch das PI-Glied sehr klein. Durch eine Anhebung der Verstärkung von 0,5 auf 0,8 kann der Abstand, wie in Abb. 4.10 dargestellt, erneut maximiert werden. In Abb. 4.11 ist deutlich erkennbar, dass durch das zusätzliche PI-Glied und die angepasste Verstärkung die Sprungantwort stationär genau und die Anstiegszeit deutlich reduziert ist. Der sich ergebende Regler aus der Schleifenoptimierung ist in Gl. 4.4 dargestellt.

$$R = \frac{\frac{s}{0,1} + 1}{\frac{s}{10} + 1} \cdot 0,8 \cdot \left(1 + \frac{2}{s}\right) \qquad \text{(Gl. 4.4)}$$

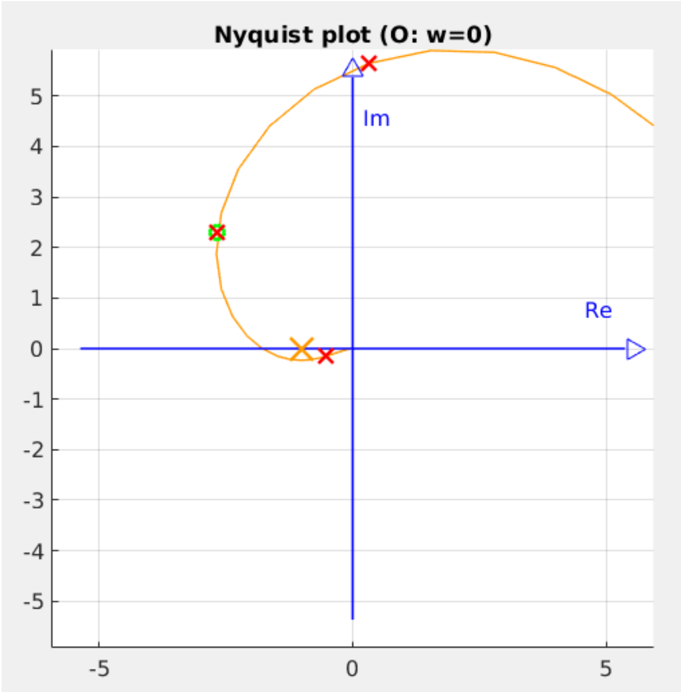


Abb. 4.10: Nyquist-Kurve des offenen Regelkreises mit angepasster Verstärkung

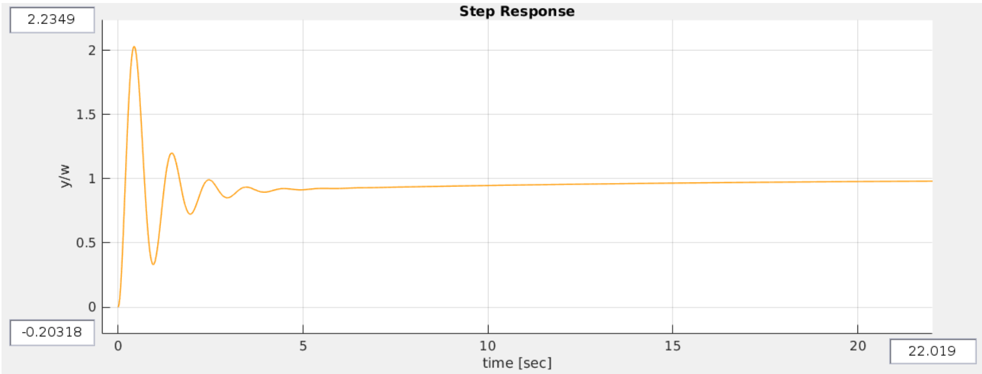


Abb. 4.11: Sprungantwort des geschlossenen Regelkreises

Eine entsprechende Reglerentwurfsumgebung muss die Nyquist-Kurve nach jedem Schritt erneut berechnen und ein Zusammenfügen beliebiger Korrekturglieder ermöglichen. Da die Festlegung der Korrekturglieder meist anhand der Eck- und Grenzfrequenzen erfolgt, müssen im Nyquist-Diagramm mehrere Anhaltspunkte für die Frequenzen auf der Kurve gegeben sein. Es ist deutlich zu erkennen, dass die

Schleifenoptimierung für den Anwender den deutlich einfacheren Ansatz darstellt, da die Auswahl der Korrekturglieder klarer ist. Die Anwendung der verschiedenen Korrekturglieder ist auf die Schritte der Schleifenoptimierung aufgeteilt. Nachfolgende Liste stellt noch einmal die Schritte mit den zugehörigen Korrekturgliedern, die in diesen Schritten zum Einsatz kommen können, dar:

1. Berechnung der Pole der Strecke: eventuell ein zusätzlicher Pol in der RHP
2. Berechnung des Umschlingungswinkels anhand von Gl. 4.1: kein Korrekturglied
3. Erzeugen der geforderten Umschlingungen um einen beliebigen Punkt auf der negativen, realen Achse: ein oder mehrere Lead- und / oder Lag-Glieder
4. Anpassung der Verstärkung: ein P-Glied
5. Einfügen des integrierenden Anteils: ein oder mehrere PI-Glieder
6. Optimierung der Verstärkung: Anpassung des bereits eingefügten P-Glieds

4.2.1 Optimierung unter Betrachtung der Regelgüte

Das oben aufgeführte Ablaufschema der Schleifenoptimierung dient zum Erstellen eines stabilen Reglers. Im letzten Schritt ist durch eine Anpassung der Verstärkung eine geringfügige Optimierung des Reglers möglich. Ziel dieses Schemas ist eine Schritt-für-Schritt-Anleitung für die Schleifenoptimierung darzustellen. Die weitere Optimierung des Reglers folgt keinem festen Ablauf, sondern beruht auf der Weiterführung bestimmter Anregungspunkte. Mit diesen Hinweisen und einer genauen Kenntnis der Korrekturglieder kann ein sehr gutes Regelergebnis erzielt werden. Ausgangspunkt für die Optimierung stellt ein stabiler Regler, wie in Gl. 4.4 dar.

Das erste Ziel der Optimierung ist eine Reduktion der Überschwinger, sowie eine Verkürzung der Abklingzeit. Dazu ist der Abstand der Nyquist-Kurve zum kritischen Punkt möglichst groß zu gestalten. Der Bereich der Kurve links vom kritischen Punkt kann über den Verstärkungsfaktor des P-Glieds vergrößert werden. Das Problem stellt der Kurven-Bereich zwischen kritischem Punkt und Ursprung dar. Das verwendete

Lead-Glied dreht den Bereich zwischen den Grenzfrequenzen gegen den Uhrzeigersinn. Wählt man die höhere Grenzfrequenz sehr hoch, ist der gedrehte Frequenzbereich breiter und Kurvenbereiche, die dem kritischen Punkt nahe liegen, werden gegen den Uhrzeigersinn weiter vom kritischen Punkt weggedreht. Da das Lead-Glied die Amplitude für hohe Frequenzen anhebt, kann eine sehr hohe zweite Grenzfrequenz zu Problemen führen. Dadurch ist es eventuell nicht möglich die höhere Frequenz beliebig zu variieren. In gezeigtem Beispiel ist ω_2 des Lead-Glieds auf 1000 rad/s zu erhöhen.

Eine weitere Möglichkeit zum Vergrößern des Abstandes zwischen Nyquist-Kurve und kritischem Punkt stellt die Kombination aus zusätzlichem Lead-Glied und Erhöhung des Verstärkungsfaktors der PI-Glieder dar. Durch das zusätzliche Lead-Glied, das mit den gleichen Frequenzen von dem bereits vorhandenen Lead-Glied eingestellt wird, erzeugt man eine Schlaufe, die bis unter den Ursprung gegen den Uhrzeigersinn gedreht ist. Durch Erhöhen des Verstärkungsfaktor des PI-Glieds wird die „Überdrehung“ im niedrigfrequenten Bereich zurückgedreht. Zudem ist die Schlaufe durch die Vergrößerung der Amplitude im Bereich der niedrigen Frequenzen durch das PI-Glied größer. Durch das zusätzliche Lead-Glied ist eine „Überdrehung“ möglich, die durch den hohen Verstärkungsfaktor im PI-Glied vergrößert zurückgedreht wird. So erhält man eine sehr große Schlaufe, die zusätzlich im hochfrequenten Bereich weiter gegen den Uhrzeigersinn gedreht ist. Dieses Vorgehen ermöglicht eine enorme Reduktion der Überschwingweite und der Abklingzeit. Es hat jedoch auch einen entscheidenden, geometrischen Vorteil. Ist der Bereich der Nyquist-Kurve nahe am Ursprung, unabhängig von dessen Form, sehr klein und verläuft die Kurve fast geradlinig vom Ursprung weg am kritischen Punkt vorbei, kann der Verstärkungsfaktor des P-Glieds fast beliebig erhöht werden, ohne dass sich die Nyquist-Kurve dem kritischen Punkt weiter nähert. Dies ermöglicht eine sehr große Variation des Verstärkungsfaktors k des P-Glieds. Da das Verstärkungsglied das Stellsignal proportional skaliert, wird dadurch auch der integrierende Anteil erhöht. Soll möglichst schnell stationäre Genaugikeit erreicht werden, ist der Verstärkungsfaktor an der oberen Grenze des möglichen Variationsbereichs zu wählen. In diesem Beispiel ist das bei $k = 0,0008$. Das Ergebnis dieses Optimierungsschrittes ist in Abb. 4.12 dargestellt. Nachfolgend eine Zusammenfassung des ersten Optimierungsschrittes.

1. Erhöhen der höheren Grenzfrequenz des bestehenden Lead-Glieds von 10 auf 1000 rad/s.

2. Zusätzliches Lead-Glied mit ebenfalls $\omega_1 = 0,1$ rad/s und $\omega_2 = 1000$ rad/s im Kombination mit einer Erhöhung der Verstärkungsfaktors des PI-Glieds auf $ki = 4000$.

3. Anpassung des P-Glieds auf $k = 0,0008$.

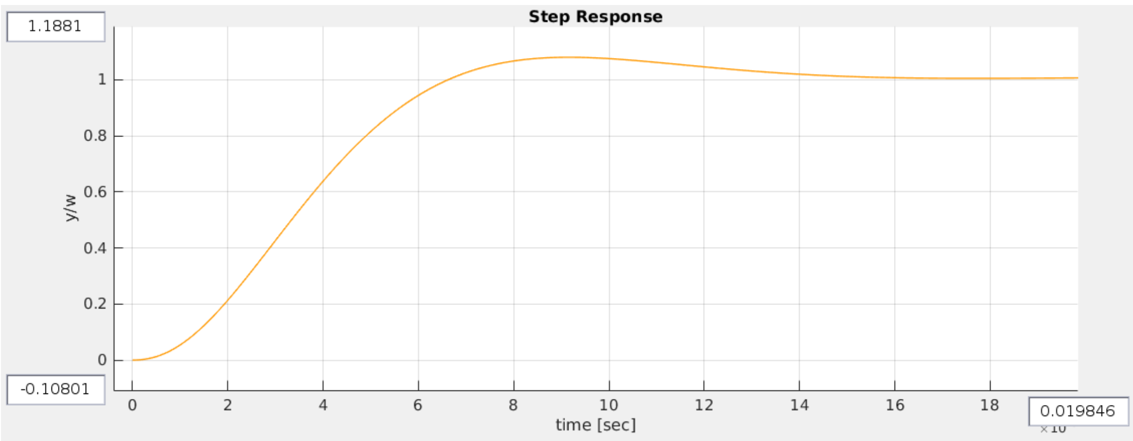


Abb. 4.12: Sprungantwort des geschlossenen Regelkreises mit optimiertem Regler

4.2.2 Optimierung unter Betrachtung der Stellgröße

Durch die vorangegangene Optimierung konnte die Überschwingweite von 100 % auf 8 % verkleinert und die Anstiegszeit von 0,131 s auf 0,00426 s verkürzt werden. Bei diesem Vorgehen ist jedoch zu berücksichtigen, dass die Erhöhung der Reglergüte mit einer Vergrößerung der Stellsignale einhergeht. Im gezeigten Beispiel ist der Maximalwert bei einer Sprungantwort durch die Optimierung um das 1500-fache gestiegen. Meist ist bei realen Anwendungen durch die mechanische Struktur oder das verbaute Stellglied die Stellgröße beschränkt. Deshalb ist im nächsten Schritt die Anstiegszeit und damit verbunden der Maximalwert der Stellgröße zu berücksichtigen. Dazu sind zu Beginn die Maximalwerte der Vorgabegröße, der Änderung der Vorgabegröße und möglicher Störungen festzulegen. Anhand des optimierten Reglers können diese Bedingungen in pzMove simuliert und so die Maximalwerte der auftretenden Stellgröße mit den Einschränkungen aus der Mechanik verglichen werden. Sind die Stellwerte zu hoch, ist der Regler weiter dahingehend zu optimieren. Durch eine Variation der differenzierenden, proportionalen und integrierenden Anteile im Regler, ist eine relativ einfache Optimierung des Reglers hinsichtlich Stellgröße möglich. Die Reduktion der Stellgröße beruht dabei auf zwei Vorgehensweisen. Zum einen kann dies durch eine Erhöhung der Anstiegszeit unter Beibehalt der grundsätzlichen Überschwing- und Abklingeigenschaften erfolgen. Zum anderen ist durch eine geringe Reduktion der Regelgüte bereits eine erhebliche Abnahme der Werte der Stellgröße zu erreichen. Die Erhöhung der Anstiegszeit erfolgt durch Vergrößerung des integrierenden Anteils und Reduktion des proportionalen und differenzierenden Anteils. Ist dieses Vorgehen ausgeschöpft, sind die Ansprüche an die Regelgüte zu senken. Dabei ist der differenzierende und der proportionale Anteil zu verringern, ohne dass dabei eine weitere Erhöhung des integrierenden Teils erfolgt.

Für die Variation der verschiedenen Anteile ist es nötig zu wissen, zu welchen Anteilen die verschiedenen Korrekturglieder beitragen. Der Pol in der rechten Halbebene hat keinen entscheidenen Einfluss und wird deshalb in diesem Optimierungsschritt nicht weiter betrachtet. Das P-Glied skaliert das gesamte Stellsignal und kann deshalb in erster Linie nicht zu einer Variation des Einflusses der verschiedenen Anteile beitragen. Zur Vergrößerung der integrierenden Komponente ist der Verstärkungsfaktor k_i der PI-Glieder zu erhöhen. Die Variation des differenzierenden Teils des Reglers erfolgt anhand der Lead- und Lag-Glieder. Dazu ist die Struktur dieser beiden Glieder in Gl. 4.5 etwas umgestellt dargestellt. Betrachtet man den zweiten Teil in Klammern, fällt auf, dass dies ein PD-Glied darstellt. Soll also der differenzierende Anteil eines Reglers vergrößert werden, erfolgt dies durch Verkleinerung der

Frequenz ω_x der Lead- oder Lag-Glieder.

$$G(s)_{Lead/Lag} = \frac{\frac{s}{\omega_x}+1}{\frac{s}{\omega_y}+1} = \frac{1}{\frac{s}{\omega_y}+1} \cdot \left(\frac{s}{\omega_x}+1\right) \qquad \text{(Gl. 4.5)}$$

Nachfolgend sind die allgemeinen Variationsmöglichkeiten für den zweiten Optimierungsschritt zusammengefasst aufgelistet:

1. Erhöhen des differenzierenden Regleranteils durch Verkleinerung der Grenzfrequenz ω_x der Lead- oder Lag-Glieder.

2. Erhöhen des integrierenden Regleranteils durch Vergrößerung des Verstärkungsfaktors k_i der PI-Glieder.

3. Erhöhen des proportionalen Regleranteils durch Reduktion der Größe der differenzierenden und integrierenden Anteile, kombiniert mit einer Erhöhung des Verstärkungsfaktors des P-Glieds.

Die Variation der Einstellwerte der Korrekturglieder erfolgt dabei in kleinen Schritten unter Betrachtung einer zeitabhängigen Darstellungsweise des geschlossenen Regelkreises, wie zum Beispiel der Sprungantwort. Dadurch ist es möglich die Optimierung vorzunehmen, ohne dabei das Nyquist-Diagramm unter Beobachtung halten zu müssen, da eventuell auftretendes, nicht-stabiles Verhalten bereits in den entsprechenden Diagrammen erkennbar ist.

In dem bestehenden Beispiel kann die Frequenz ω_1 der Lied-Glieder um den Faktor 20 auf $\omega_1 = 2$ rad/s erhöht werden. Damit wieder 8 % Überschwingweite erreicht werden, ist k_i des PI-Glieds auf 385000 zu erhöhen. Die Anstiegszeit der Sprungantwort steigt dabei um das 4,6-fache auf 0,0196 s. Mit diesem Vorgehen ist es möglich, die Werte der Stellgröße um den Faktor 4,3 zu senken. Eine zusätzliche Senkung der Maximalwerte der Stellgröße ist durch weitere Reduktion des differenzierenden Anteils möglich. Dabei ist die niedrigere Grenzfrequenz der Lead-Glieder weiter auf 3,05 rad/s zu erhöhen. Dadurch steigt die Überschwingweite auf 20 %, jedoch sinkt der maximale Stellwert bei einer Sprungantwort weiter um das 2,3-fache. Somit ist in diesem zweiten Optimierungsschritt im gezeigten Beispiel eine Reduktion der Stellgrößenwerte um den Faktor 10 möglich, indem die Anstiegszeit um das 4,6-fache und die Überschwingweite um das 2,5-fache erhöht werden. Die Übertragungsfunktion des stabilen und zweifach optimierten Reglers ist in Gl. 4.6 gezeigt. Die Sprungantwort des geschlossenen Regelsystems ist in Abb. 4.13 dargestellt.

$$G_R(s) = 0{,}0008 \cdot \frac{\frac{s}{3{,}05}+1}{\frac{s}{1000}+1} \cdot \left(1+\frac{385000}{s}\right) \cdot \frac{\frac{s}{3{,}05}+1}{\frac{s}{1000}+1} \tag{Gl. 4.6}$$

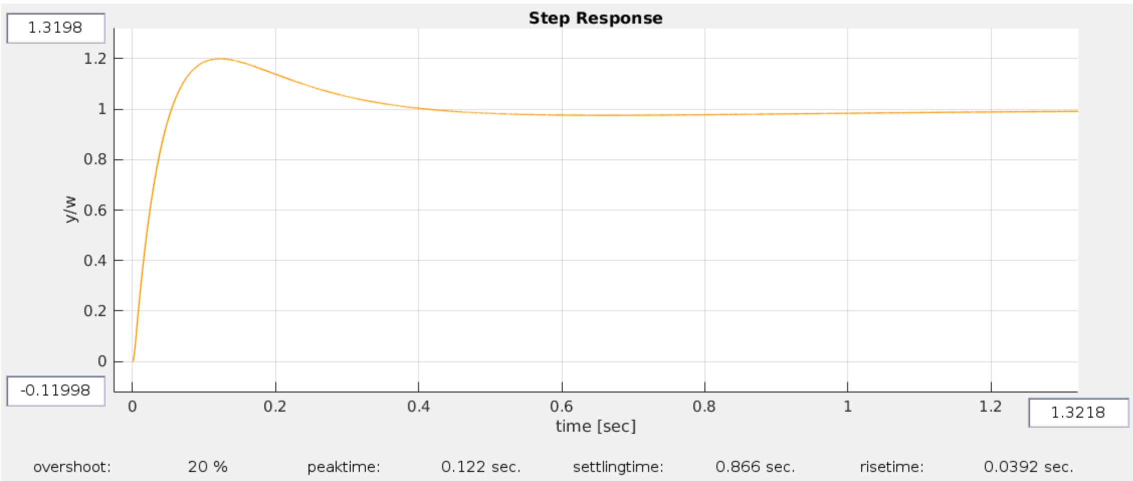


Abb. 4.13: Sprungantwort des geschlossenen Regelkreises mit zweifach optimiertem Regler

4.3 Korrekturglieder

Die Korrekturglieder sind teilweise dem Vorbericht [1, S. 33-37] entnommen, jedoch ist die nachfolgende Darstellung der Korrekturglieder speziell in Hinblick auf die Kriteriumsherleitung und die Schleifenoptimierung ausgearbeitet. Für die Veranschaulichung der Auswirkungen der Korrekturglieder im Nyquist-Diagramm sind diese beispielhaft an zwei Demo-Strecken angewandt. Die erste Demo-Strecke aus Gl. 4.7 ist aufgrund der identischen Vorzeichen bei allen Komponenten im Nenner stabil.

Die Übertragungsfunktion der zweiten Strecke ist in Gl. 4.8 abgebildet und unterscheidet sich nur im Vorzeichen von „s“ von der ersten Demo-Strecke. Die zweite Demo-Strecke ist durch das geänderte Vorzeichen instabil. Die Bode-Diagramme beider Strecken sind in Abb. 4.14 dargestellt. Sie unterscheiden sich nur im Vorzeichen im Phasen-Verlauf. Die Beschreibung der Korrekturglieder erfolgt in drei Schritten. Im ersten Teil sind deren Verläufe im Bode-Diagramm beschrieben. Diese Verläufe sind vor allem für die Reglerbildung bei der Anwendung der Kriteriumsherleitung nötig. Weiter sind für jedes Korrekturglied zwei Merkregeln für die Anwendung in der Schleifenoptimierung angegeben. Diese Merkregeln werden im dritten Teil anhand der beiden Demo-Strecken graphisch veranschaulicht, wobei die Grenzfrequenzen der Glieder dabei auf oder um den Knick der Amplitudenverläufe der Demo-Strecken gelegt sind. Mit den beiden Merkregeln und der beispielhaften Anwendung der Glieder an den beiden Strecken ist eine Art „Merkzettel“ gegeben, mit dem die richtige Anwendung der Korrekturglieder in der Schleifenoptimierung zum Erzeugen eines stabilen Reglers oder im ersten Optimierungsschritt vereinfacht wird.

$$S_1 = \frac{1}{s^2 + s + 1} \tag{Gl. 4.7}$$

$$S_2 = \frac{1}{s^2 - s + 1} \tag{Gl. 4.8}$$

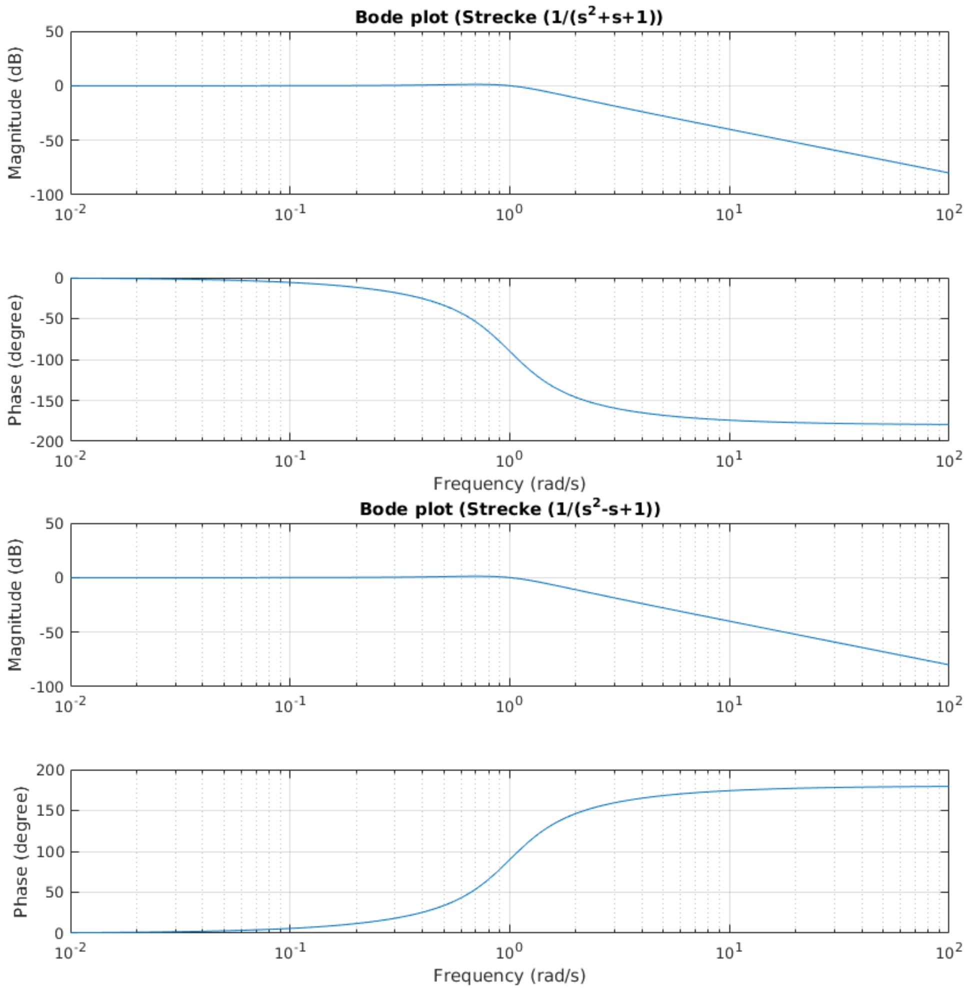


Abb. 4.14: Bode-Diagramme einer stabilen und einer instabilen Strecke

4.3.1 Das Lead-Glied

$$Lead = \frac{\frac{s}{\omega_{grenz\ 1}} + 1}{\frac{s}{\omega_{grenz\ 2}} + 1} \qquad \text{mit:} \qquad \omega_{grenz\ 1} < \omega_{grenz\ 2} \qquad \text{(Gl. 4.9)}$$

Das Lead-Glied, auch als phasenanhebendes Korrekturglied bezeichnet, ist das erste der beiden Glieder, mit dem Umschlingungen erzeugbar sind. Wie in Abb. 4.15 zu sehen, hebt dieses Glied die Phase zwischen den beiden Grenzfrequenzen. Die Amplitude wird für hohe Frequenzen angehoben. Mit zunehmendem Abstand der Grenzfrequenzen nähert sich die Phaseanhebung 90 Grad an und die Amplitudenanhebung steigt.

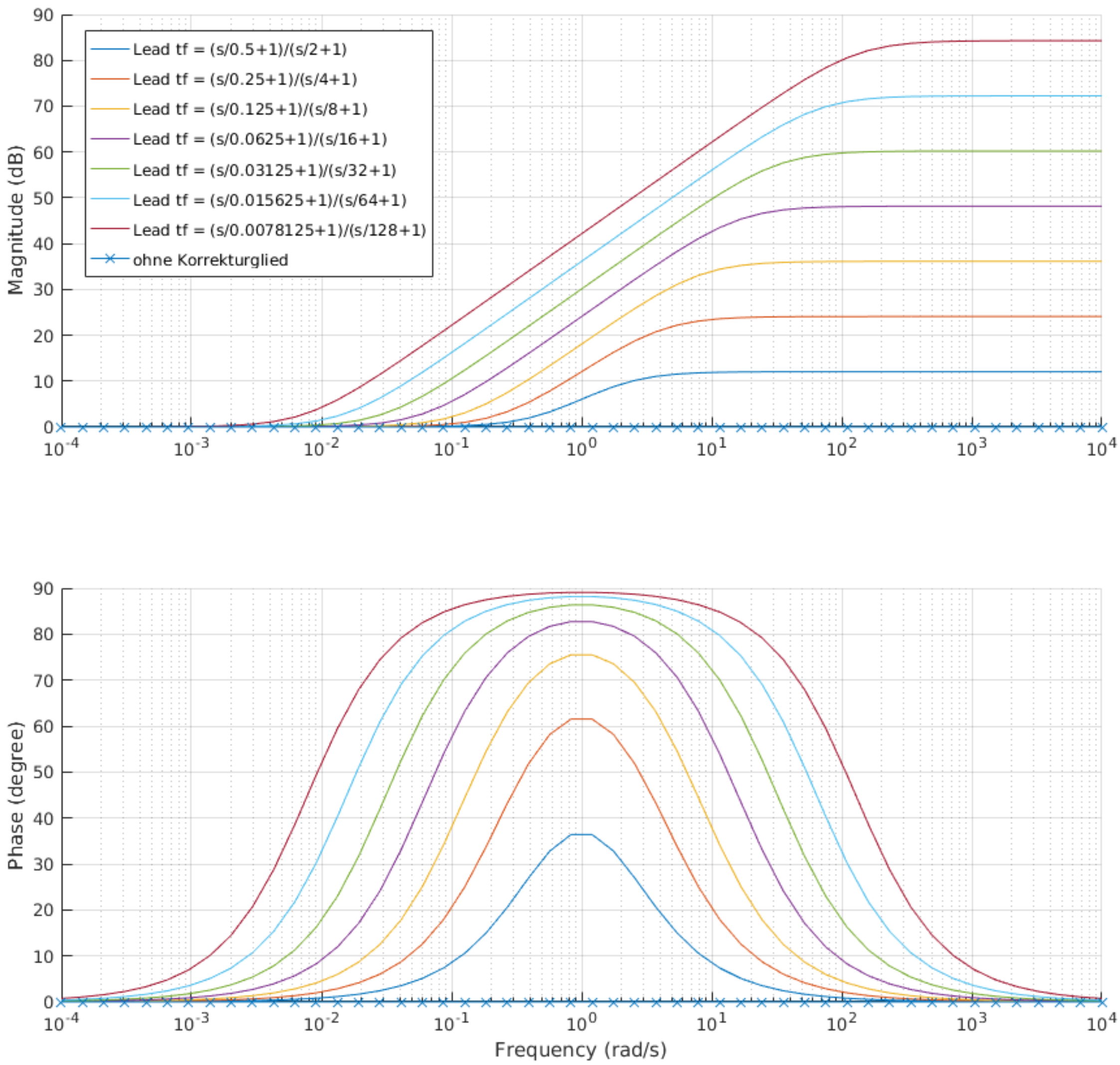


Abb. 4.15: Auswirkungen des Lead-Glieds im Bode-Diagramm

Die für die Schleifenoptimierung wichtigen Anwendungspunkte lauten wie folgt:

1. Der Endbereich wird vergrößert.
2. Der mittlere Bereich wird gegen den Uhrzeigersinn um den Ursprung gedreht.

Das Lead-Glied bildet eine erweiterte und gegen den Uhrzeigersinn gedrehte „Blase“. Deshalb kommt dieses Glied vor allem zum Einsatz, wenn mindestens eine Umschlingung gefordert ist und die Kurve gegen den Uhrzeigersinn nach links „gebeugt“ werden muss.

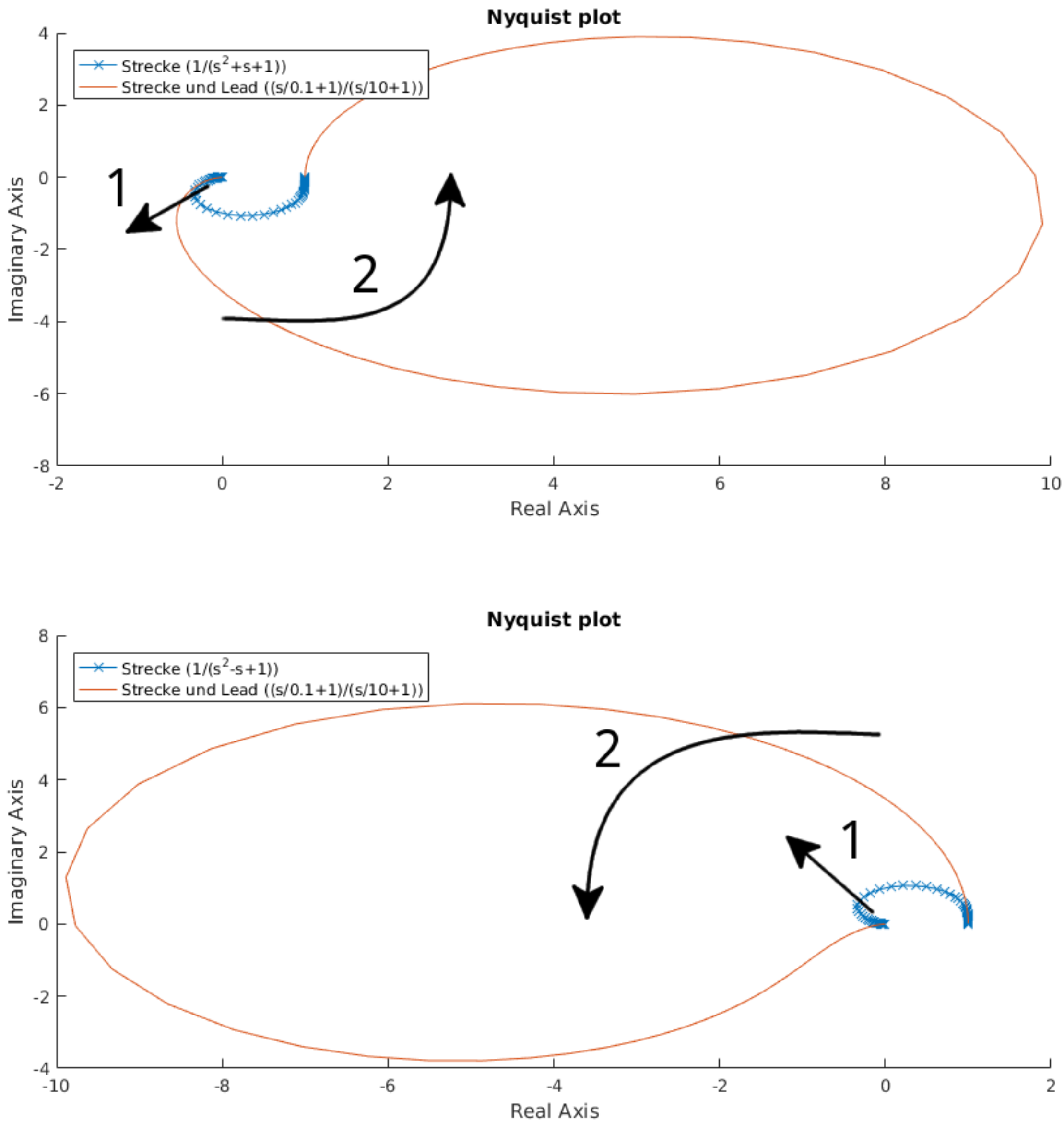


Abb. 4.16: Auswirkungen des Lead-Glieds im Nyquist-Diagramm

4.3.2 Das Lag-Glied

$$Lag = \frac{\frac{s}{\omega_{grenz\ 2}} + 1}{\frac{s}{\omega_{grenz\ 1}} + 1} \qquad \text{mit:} \qquad \omega_{grenz\ 1} < \omega_{grenz\ 2} \qquad \text{(Gl. 4.10)}$$

Das Lag-Glied, auch als phasenabsenkendes Glied bezeichnet, senkt die Phase zwischen den Grenzfrequenzen um bis zu 90 Grad. Die Amplitude für hohe Frequenzen wird ebenfalls reduziert. Der Unterschied in den Übertragungsfunktionen von Lead- und Lag-Glied besteht lediglich in den vertauschten Grenzfrequenzen, wie in Gl. 4.10 zu erkennen ist. Analog zum Lead-Glied nimmt die Phasen- und Amplitudenabsenkung mit steigendem Abstand zwischen den Grenzfrequenzen zu.

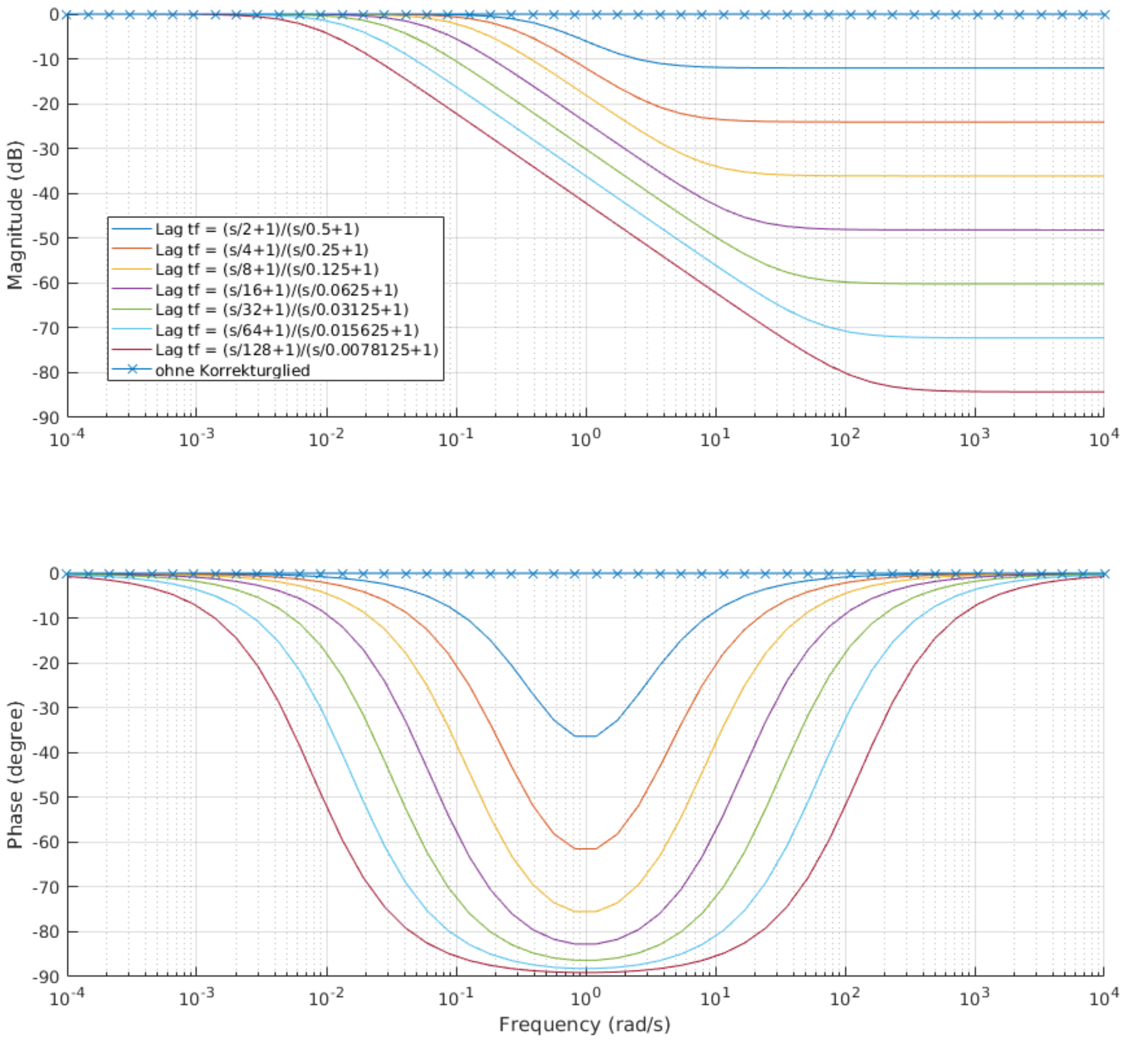
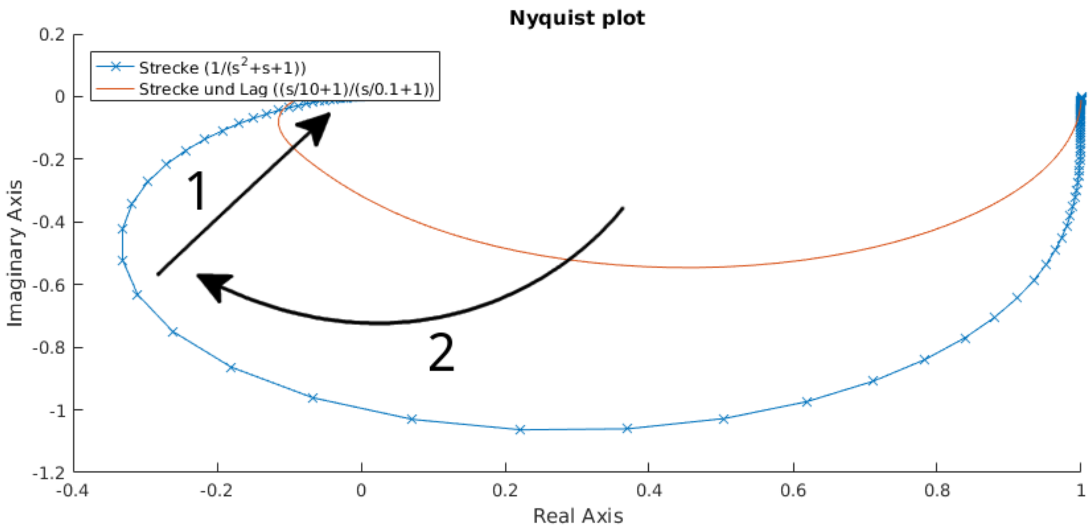


Abb. 4.17: Auswirkungen des Lag-Glieds im Bode-Diagramm

Die für die Schleifenoptimierung wichtigen Anwendungspunkte lauten wie folgt:

1. Der Endbereich wird verkleinert.
2. Der mittlere Bereich wird im Uhrzeigersinn um den Ursprung gedreht.

Dieses Glied dient vor allem zum Verhindern einer Umschlingung oder zum Verkleinern des End-Bereichs, falls dieser bei der Skalierung der Nyquist-Kurve stört.



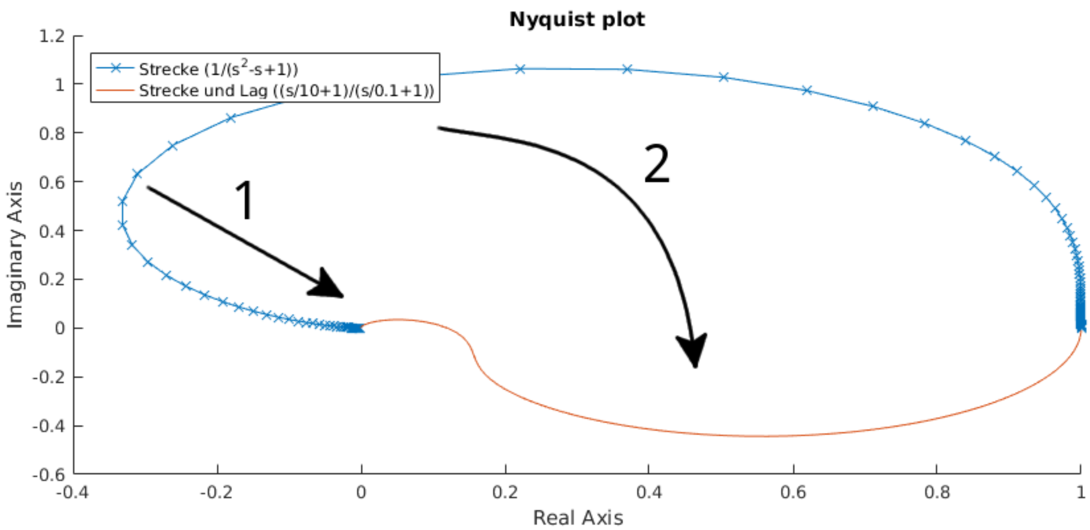


Abb. 4.18: Auswirkungen des Lag-Glieds im Nyquist-Diagramm

4.3.3 Das P-Glied

$$P = k_p \tag{Gl. 4.11}$$

Das P-Glied ist das einfachste der Korrekturglieder. Die Phase ist kostant 0 und verändert somit den Phasengang des offenen Regelkreises nicht. Der Amplitudengang entspricht konstant dem Wert k_p.

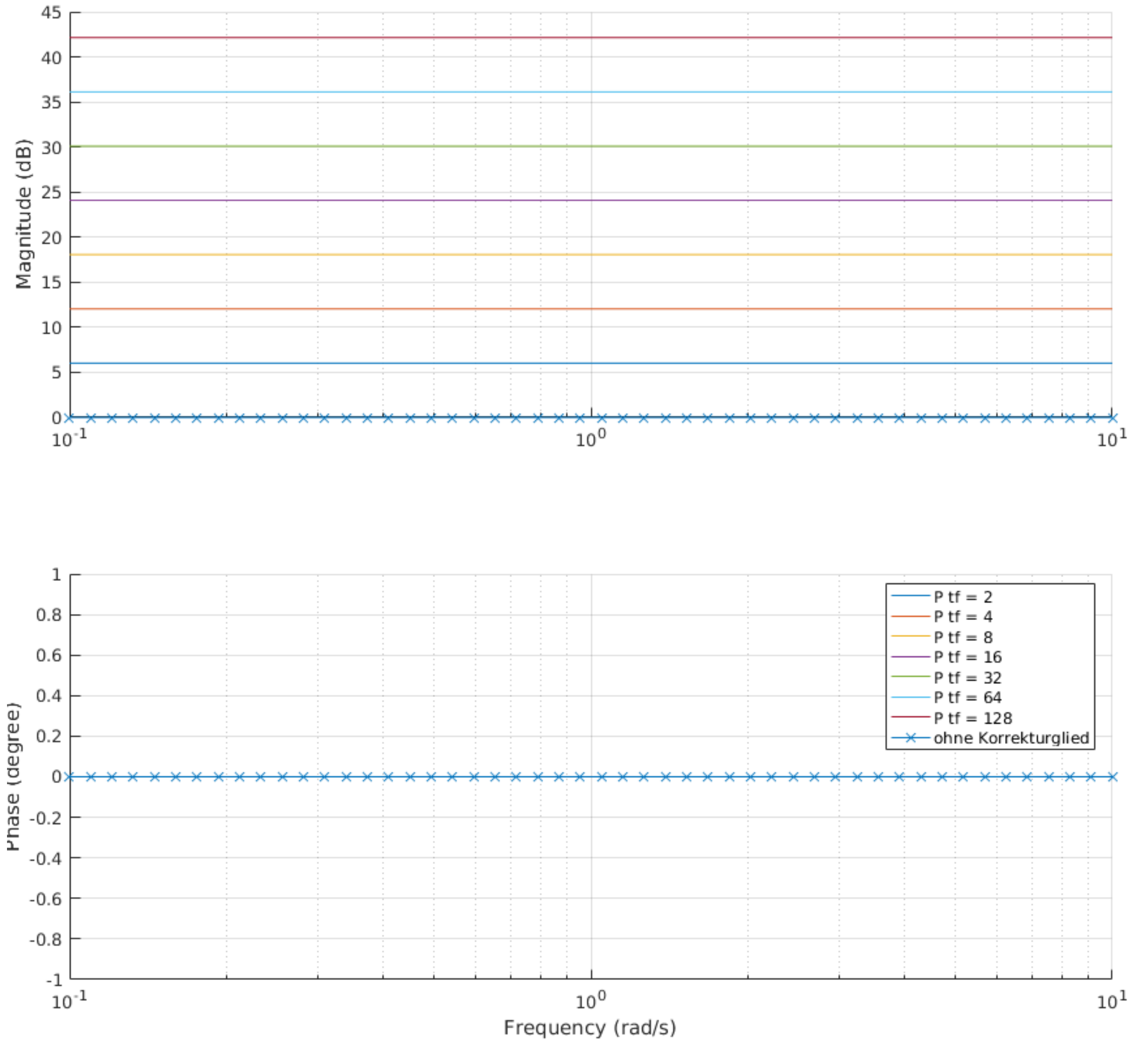


Abb. 4.19: Auswirkungen des P-Glieds im Bode-Diagramm

Der für die Schleifenoptimierung wichtige Anwendungspunkt lautet wie folgt:

1. Die gesamte Nyquist-Kurve wird proportional zum Ursprung skaliert.

Durch die Skalierung der Kurve wird das P-Glied vor allem angewendet, um die Umschlingungen auf den kritischen Punkt P(-1,0) zu verschieben und den Abstand zwischen diesem Punkt und der Nyquist-Kurve einzustellen. Je größer der Abstand, desto größer ist die Stabilität. Dadurch wird zu großes Überschwingen verhindert und die Abklingzeit nach einer Anregung verkürzt.

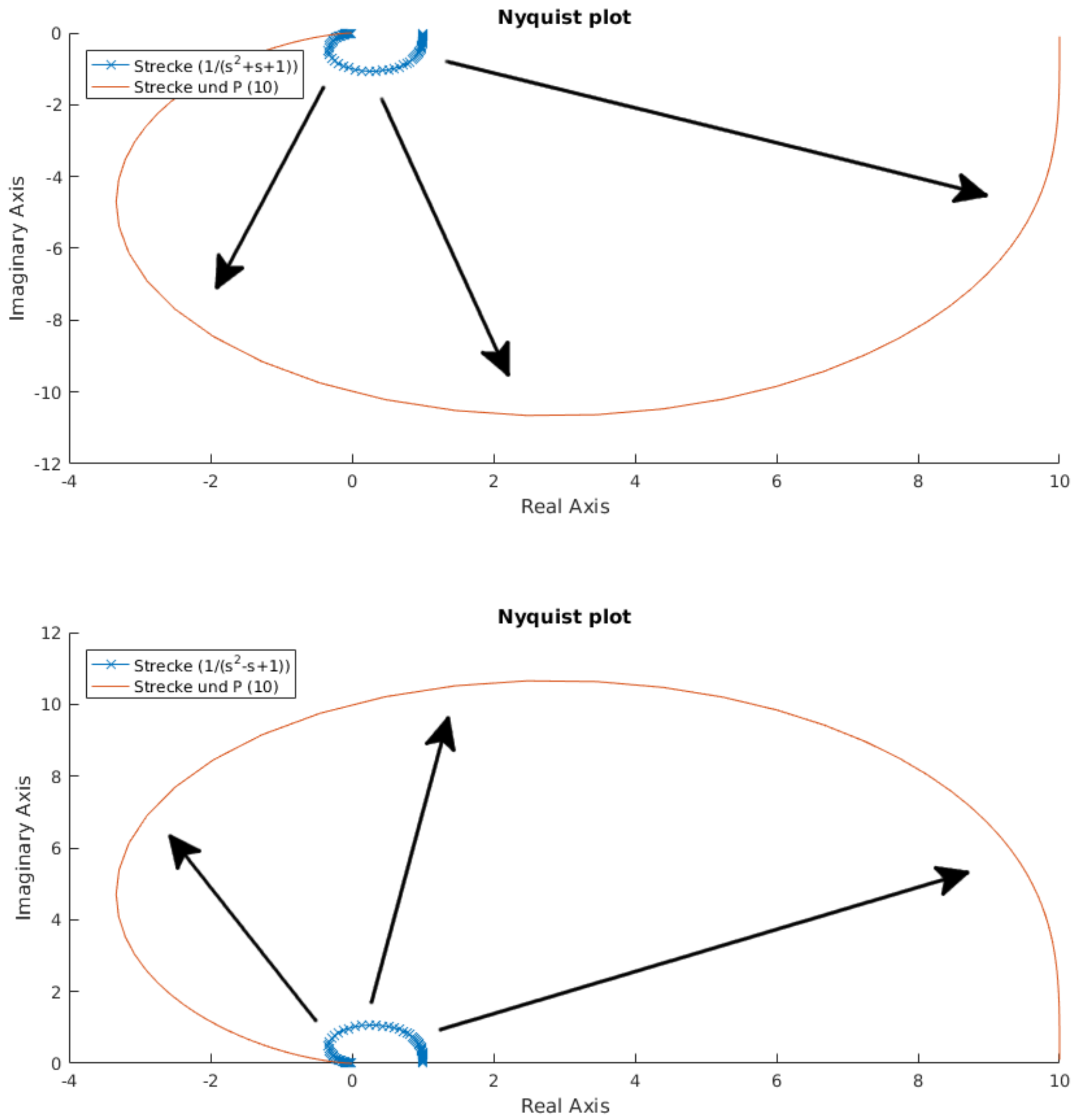


Abb. 4.20: Auswirkungen des P-Glieds im Nyquist-Diagramm

4.3.4 Das PI-Glied

$$PI = 1 + \frac{k_i}{s} \qquad\qquad PI = 1 + \frac{\omega_e}{s} \tag{Gl. 4.12}$$

Für stationäre Genauigkeit im geschlossenen Regelkreis muss ein integrierender Anteil in der offenen Kette vorhanden sein, der die statische Abweichung aufsummiert und so die bleibende Differenz ausregelt. Stationäre Genauigkeit wird durch ein oder mehrere PI-Glieder erreicht. Dazu muss die Phase der offenen Kette für sehr kleine Frequenzen kleiner oder gleich - 90 Grad sein. Die Übertragungsfunktion des PI-Glieds ist in Gl. 4.12 in zwei verschiedenen Versionen dargestellt. Die erste Darstellung mit dem Faktor k_i verdeutlicht, dass durch ein größeres k_i der integrierende Anteil vergrößert wird. Das PI-Glied hat eine Nullstelle bei $-k_i + 0j$. Der Betrag der Nullstelle stellt die Eckfrequenz in rad/s im Bodediagramm dar [4, S. 236]. Dadurch ist der Verstärkungsfaktor über die Interpretation als Eckfrequenz in der Schleifenoptimierung einfach bestimmbar.

Das PI-Glied fügt einen zusätzlichen Pol im Ursprung ein, der im Nyquist-Kriterium eine zusätzliche Umschlingung von 90 Grad fordert. Da die Amplitude durch das Glied für sehr kleine Frequenzen ins Unendliche angehoben und die Phase in diesem Bereich um 90 Grad abgesenkt wird, ist genau diese zusätzliche Umschlingung gegeben. Somit sind beliebig viele PI-Glieder einfügbar, sofern diese nicht die Umschlingung gefährden. Zu beachten ist jedoch, dass sich der zusätzliche Pol mit einer eventuell vorhandenen Nullstelle der Strecke kürzen lässt. Tritt dieser Fall auf, wird keine zusätzliche Umschlingung durch das PI-Glied gefordert.

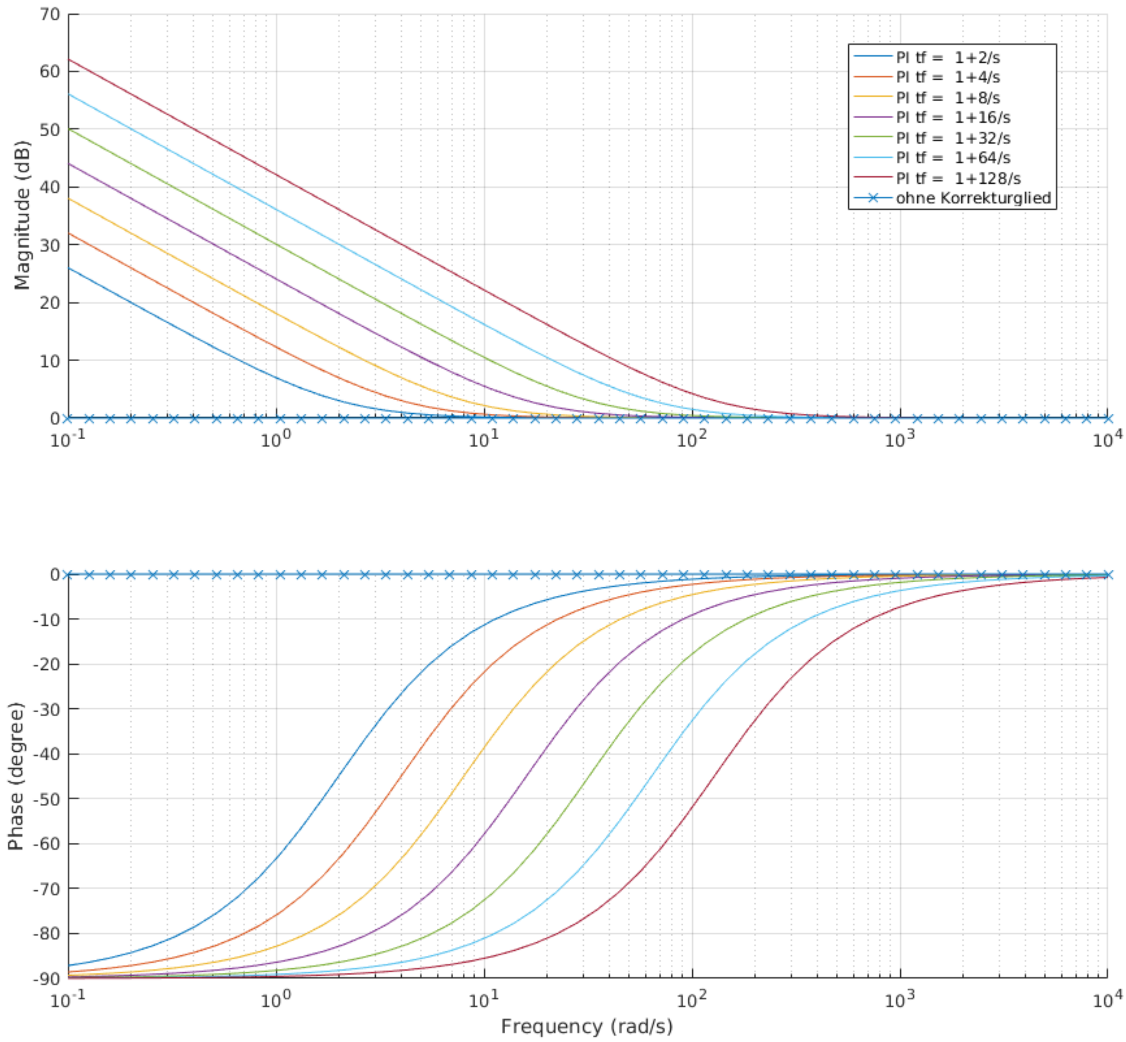


Abb. 4.21: Auswirkungen des PI-Glieds im Bode-Diagramm

Die für die Schleifenoptimierung wichtigen Anwendungspunkte lauten wie folgt:

1. Der Anfangsbereich wird um 90 Grad im Uhrzeigersinn gedreht.

2. Zudem wird der Anfangsbereich vergrößert.

Durch die abgesenkte Phasenlage wird dieses Glied ausschließlich für die Erzeugung der stationären Genauigkeit eingesetzt. In Kombination mit einem zusätzlichen Lead-Glied und einem erhöhten Verstärkungsfaktor k_i kann dieses Glied auch zur Optimierung des Reglers beitragen.

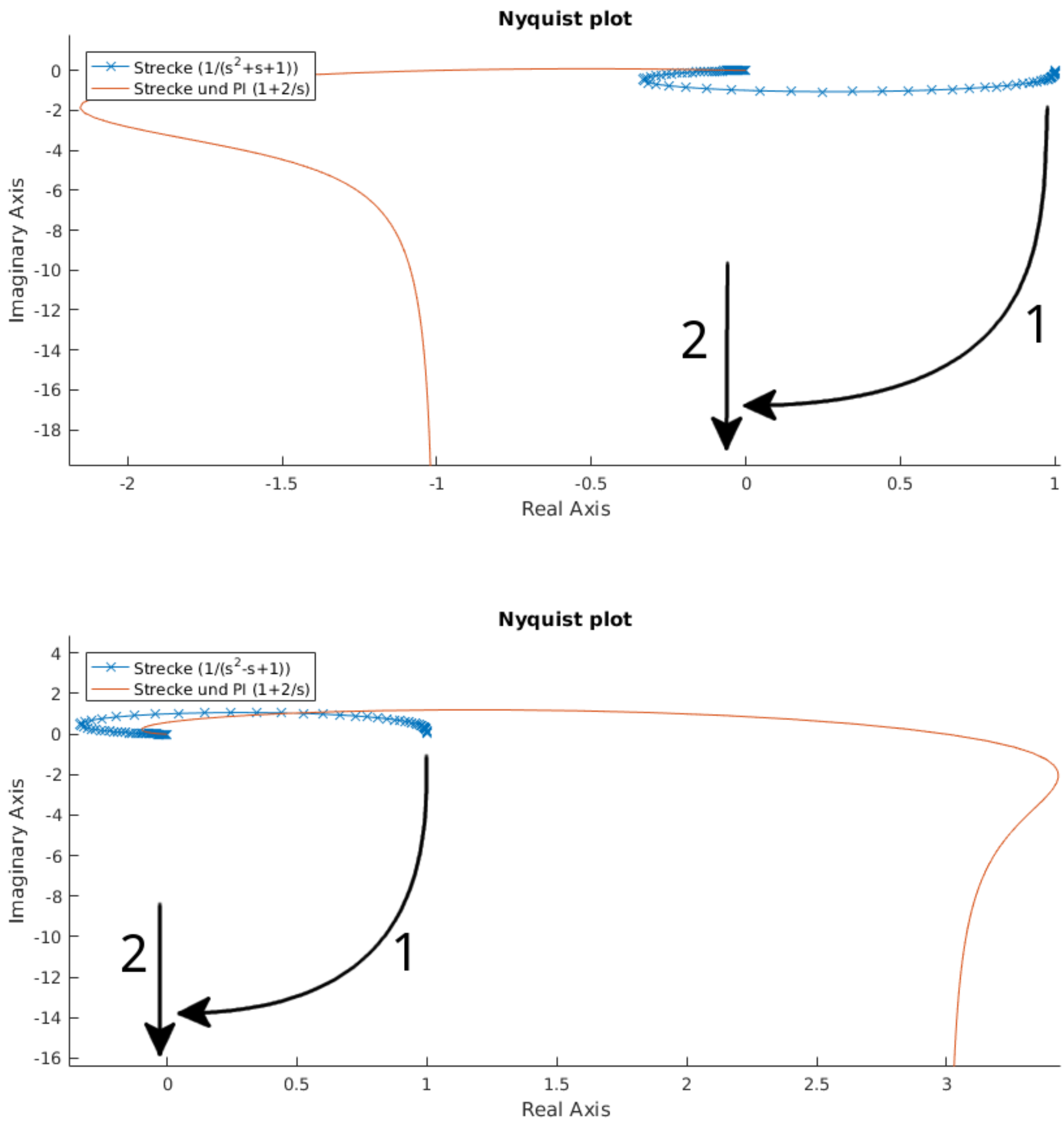


Abb. 4.22: Auswirkungen des PI-Glieds im Nyquist-Diagramm

4.3.5 Zusätzlicher Pol in der RHP

$$Pol_{RHP} = \frac{1}{1 - \frac{s}{\omega}} \qquad \text{(Gl. 4.13)}$$

Der Pol in der rechten Halbebene soll hier als Korrekturglied aufgefasst werden. Der geschlossene Regelkreis sollte proportionales oder stationär genaues Verhalten aufweisen. Dies ist gegeben, wenn die offene Kette proportionales oder integrierendes

Verhalten hat. Dies wiederum ist gegeben, wenn die offene Kette in gekürzter Form keine Nullstelle bei $0 + 0j$ hat. Hat die Strecke derartige Nullstellen, werden diese im Verlauf der Reglerauslegung durch PI-Glieder gekürzt. Dadurch ist die Phase für sehr kleine Frequenzen 0 Grad oder negativ. Hat die Strecke eine ungerade Anzahl an instabilen Polen, fordert das Nyquist-Kriterium keine ganzzahlige Anzahl an Umschlingungen. Eine halbe Umschlingung würde bedeuten, dass die Phase der offenen Kette für sehr kleine Frequenzen somit bei -180 Grad liegen müsste. Da kein Korrekturglied vorhanden ist, das die Phase für sehr kleine Frequenzen absenkt, ohne dabei über das Nyquist-Kriterium die geforderte Umschlingung zu erhöhen, ist dies nicht erreichbar. Damit kann eine instabile Strecke nur stabilisiert werden, wenn diese eine gerade Anzahl an instabilen Polen hat. Bei Strecken mit einer ungeraden Anzahl an instabilen Polen ist deshalb ein zusätzlicher instabiler Pol einzufügen. Durch den instabilen Pol im Regler, ergibt sich eine abdriftende Stellgröße. Der Regelkreis ist dann zwar eingangs-ausgangs-stabil, jedoch muss die Stellgröße über beispielsweise eine zweite Kaskade mit eigenem Regler zurückgeführt und so auch innere Stabilität erzeugt werden. Im Bode-Diagramm in Abb. 4.23 ist zu sehen, dass der instabile Pol für hohe Frequenzen die Amplitude senkt und die Phase um 90 Grad anhebt.

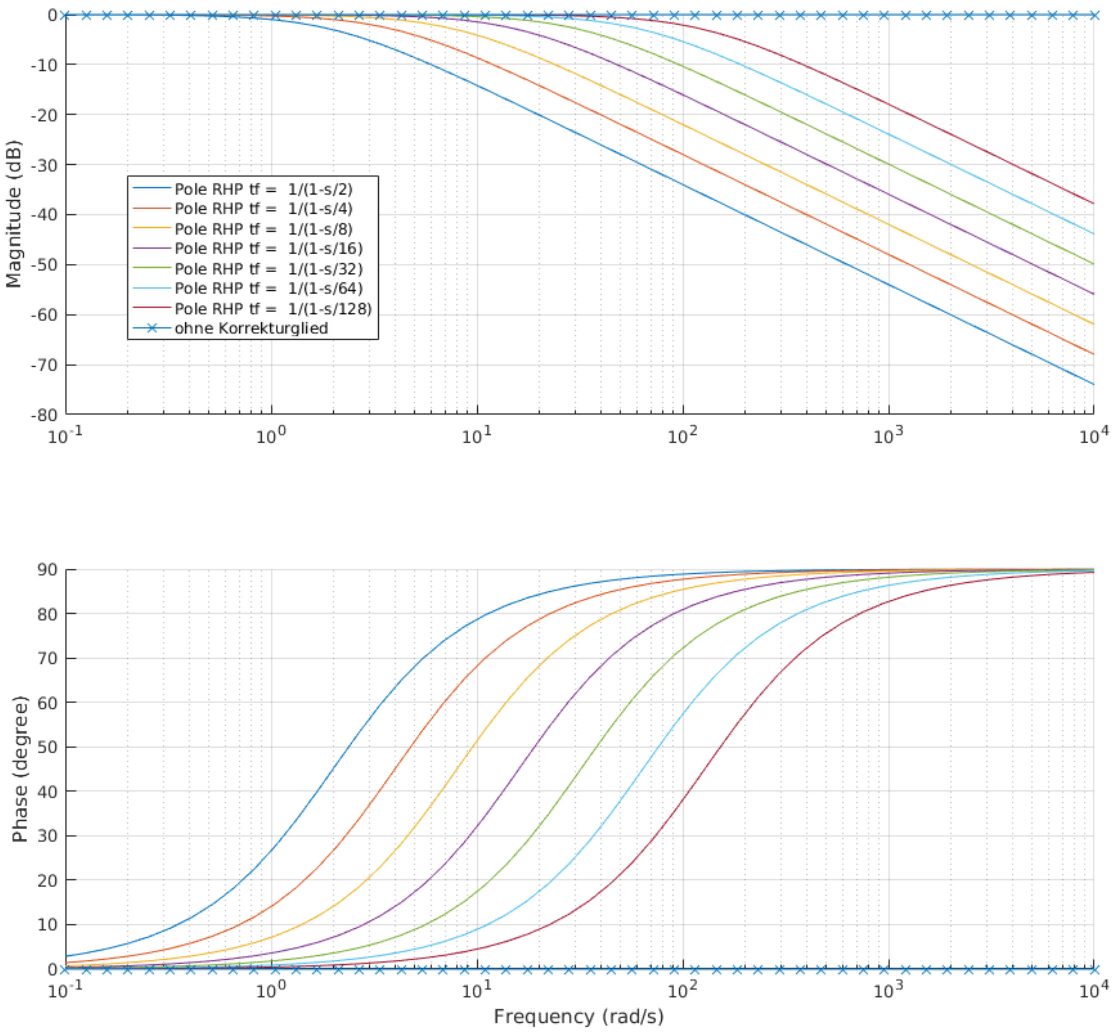


Abb. 4.23: Auswirkungen eines zusätzlichen Pols in der rechten Halbebene im Bode-Diagramm

Die für die Schleifenoptimierung wichtigen Anwendungspunkte lauten wie folgt:

1. Der Endbereich wird um 90 Grad gegen den Uhrzeigersinn gedreht.
2. Zudem wird der Endbereich verkleinert.

Durch seine Eigenschaften wird der zusätzliche Pol in der rechten Halbebene nur angewendet, wenn nach dem Nyquist-Kriterium keine ganzzahlige Anzahl an Umschlingungen gefordert wird. Deshalb ist dieses Korrekturglied nur maximal ein Mal in einem entsprechenden Regler einzusetzen. Hilfreicher Nebeneffekt ist, dass bei einer instabilen Strecke, wie in Abb. 4.24 dargestellt, die Nyquist-Kurve durch den zusätzlichen Pol bereits gegen den Uhrzeigersinn gedreht wird und damit die geforderten Umschlingungen leichter erreichbar sind.

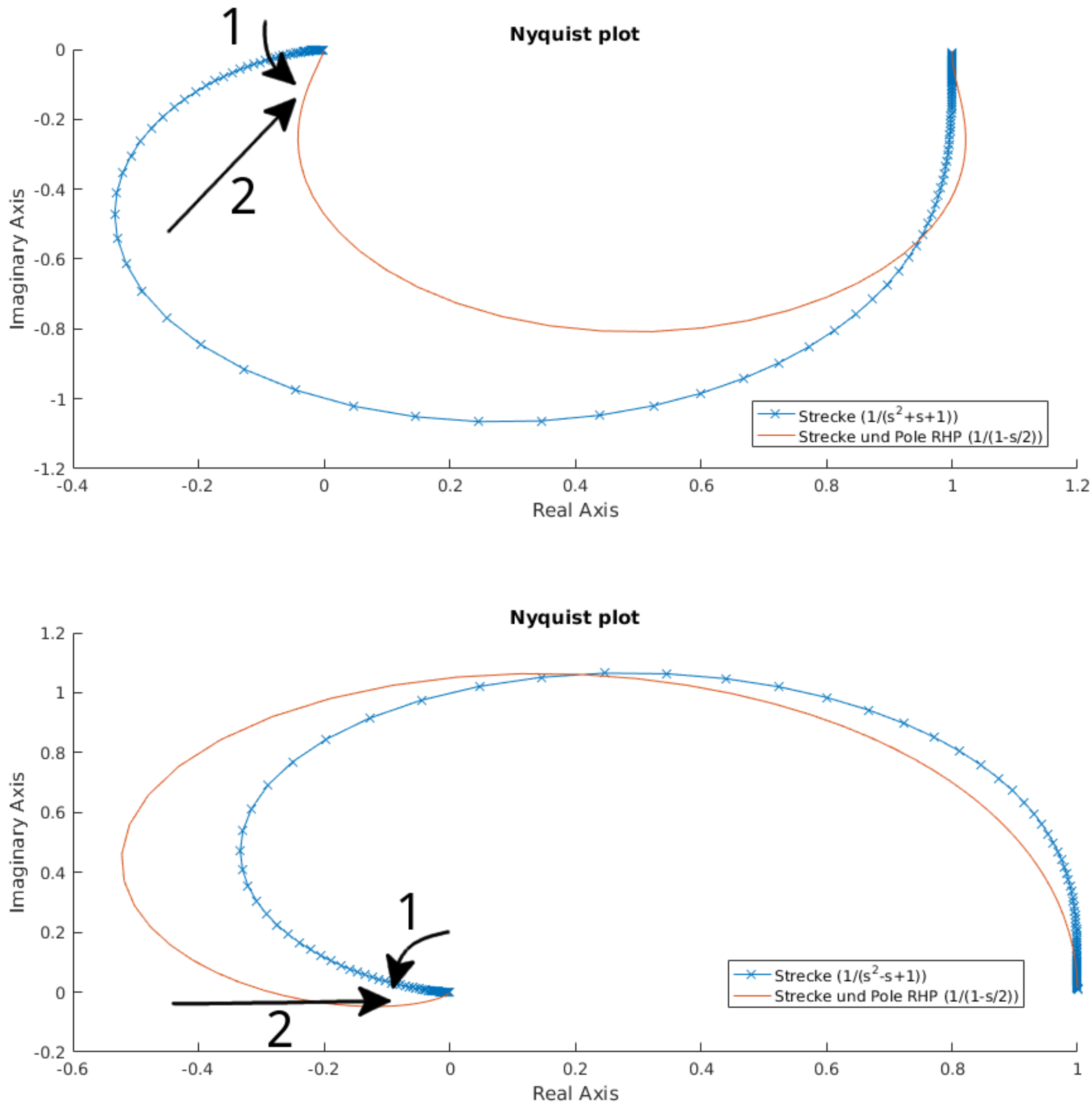


Abb. 4.24: Auswirkungen eines zusätzlichen Pols in der rechten Halbebene im Nyquist-Diagramm

4.4 Integration des Loop-Shaping-Verfahrens in pzMove

Die Reglerentwurfsumgebung pzMove ist für die Anwendung der Schleifenoptimierung vorbereitet. Dazu ist es möglich eine beliebige Anzahl der oben aufgeführten Korrekturglieder zu einem Regler zusammenzuführen und die Auswirkungen der einzelnen Glieder zu betrachten. Neben dem, in der Mitte von Abb. 4.25 dargestellten, Popupmenü finden sich drei Buttons. Mit dem „+“-Button öffnet sich ein Auswahlmenü, mit dem ein neues Korrekturglied hinzugefügt werden kann. Mit dem „–“-Button entfernt man das aktuell im Popupmenü ausgewählte Korrekturglied. Der „dis/en“-Button ermöglicht das aktivieren/deaktivieren der Korrekturglieder. Möchte man zum Beispiel nur die Auswirkungen eines einzelnen Korrekturglieds betrachten, kann durch Auswahl des entsprechenden Glieds im Popupmenü und durch Betätigen der Disable-Funktion dieses Glied deaktiviert werden. Durch das Aktivieren und Deaktivieren eines Glieds, ist im Nyquist-Diagramm direkt dessen Auswirkung erkennbar. Alle deaktivierten Korrekturglieder sind im Popupmenü grau dargestellt, wodurch auf einen Blick alle aktiven Glieder erkennbar sind. Unter dem Popupmenü liegen die Eingabefelder, die das Verändern der Variablen der einzelnen Glieder ermöglichen. Dazu muss das entsprechende Korrekturglied im Popupmenü ausgewählt sein.

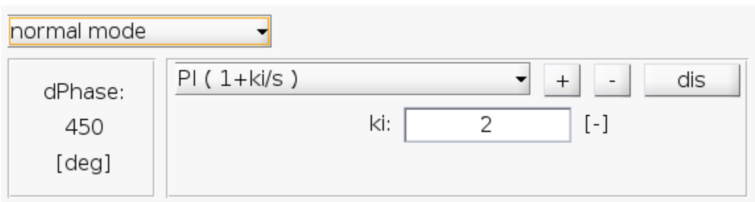


Abb. 4.25: Eingabebereich zur Reglerentwicklung nach dem Loop-Shaping-Verfahren in pzMove

Auf der linken Seite ist der, nach dem Nyquist-Kriterium geforderte, Umschlingungswinkel dPhase in Grad dargestellt. Somit muss der Anwender zur Kontrolle der Stabilität lediglich diesen Wert ablesen und mit der Umschlingung im Nyquist-Diagramm vergleichen. Da aufgrund der PI-Glieder die Amplitude sehr groß werden kann, ist das Nyquist-Diagramm eventuell zu klein skaliert. Bewegt man die Maus über das Nyquist-Diagramm, kann mit dem Mausrad die Darstellung um den Ursprung skaliert und so das Diagramm genauer betrachtet werden.

Über dem Popupmenü für die Korrekturglieder ist ein weiteres Popupmenü mit den Auswahlmöglichkeiten „design mode“ und „normal mode“ verfügbar. Ist dort „design mode“ ausgewählt, sind alle Diagramme für den offenen Regelkreis dargestellt, weshalb alle zeitabhängigen Diagramme nicht auswählbar sind. Im normalen Modus sind die Diagramme für den geschlossenen Regelkreis dargestellt.

Die Auslegung eines Reglers nach dem Loop-Shaping-Verfahren mit pzMove ist für die Schleifenoptimierung ausgelegt. Wählt man den Entwicklungsmodus aus, wird direkt das Nyquist-Diagramm dargestellt. Nun ist es möglich anhand des Diagramms einen stabilen Regler aus den verschiedenen Korrekturgliedern zu erzeugen. Speziell für dieses Verfahren sind im Nyquist-Diagramm fünf Punkte mit den entsprechenden Frequenzen gekennzeichnet, welche bei der Festlegung der Grenzfrequenzen der Korrekturglieder behilflich sind. Ist die Reglerentwicklung abgeschlossen, kann in den normalen Modus gewechselt werden. Dabei wird eine Diagrammübersicht, bestehend aus Bode-Diagramm, Nyquist-Diagramm, Sprungantwort und Impulsantwort, dargestellt. In dieser Übersicht kann die Regelgüte anhand der Sprungantwort und der Impulsantwort bestimmt werden. Möchte man den Regler weiter optimieren, ist wieder in den „design mode“ zu wechseln, woraufhin sofort wieder das Nyquist-Diagramm des offenen Regelkreises dargestellt wird. Durch die automatische Auswahl der Diagramme beim Wechsel der Modi ist ohne Umschalten der Reglerentwurf unter Betrachtung der Stabilität im Nyquist-Diagramm des offenen Regelkreises und der Regelgüte aus der Sprungantwort und der Impulsantwort vornehmbar. Die Übertragungsfunktion des fertigen Reglers kann anschließend exportiert oder durch Auswahl von GR im oberen Fensterbereich dargestellt werden.

5 Anwendung am Beispiel des LEGO® MINDSTORMS® EV3-Roboters

Abb. 5.1: LEGO® MINDSTORMS® EV3-Roboter [2]

Als Anwendungsbeispiel für die Kaskadenregelung und des Loop-Shaping-Verfahrens unter Verwendung von pzMove dient der „Gyroboy“. Der in Abb. 5.1 dargestellte Roboter baut auf die LEGO® MINDSTORMS®-Technik auf und bietet so vor allem für den späteren Einsatz in einem Praktikum für Studenten entscheidende Vorteile gegenüber einem selbst erstellten Aufbau. Der Steuerungsstein ist direkt aus MATLAB® Simulink heraus programmierbar. Zudem ist ein Set mit 349,99 € [6] sehr günstig. Ein weiterer Vorteil dieser Applikation ist die Mobilität des Systems. Der LEGO® MINDSTORMS® EV3-Roboter ist einfach zu transportieren und für die Programmierung ist nur ein Rechner mit installiertem MATLAB® Simulink und einer WLAN-Verbindung zum LEGO® MINDSTORMS® EV3-Roboter nötig. Nach der Auslegung der Regler für diese Anwendung, können diese aus pzMove exportiert und in Matlab importiert werden. Anschließend sind sie durch einen entsprechenden Block in MATLAB® Simulink verwendbar. Sind die Kosten für mehrere LEGO® MINDSTORMS® EV3-Roboter zu hoch, können die Studenten ihre Regler in Kleingruppen erstellen und anschließend über einen USB-Stick oder über das Netzwerk am Programmierrechner zur Verfügung stellen und am realen Objekt testen.

Der Segway® stellt im Wesentlichen das bekannte Problem des inversen Pendels dar. Der Unterschied zwischen dem inversen Pendel und dem Segway®-Roboter ist das fehlende Motormoment beim einfachen inversen Pendel. Für die Reglerauslegung ist ein Modell vom LEGO® MINDSTORMS® EV3-Roboter nötig. Dies ist entweder durch eine Messung mit anschließender Identifikation mit pzMove oder über eine mathematische Herleitung zu erstellen. Da der Segway® ein instabiles System darstellt, ist für eine Messung nur ein sehr kleiner Zeitbereich verfügbar, der zu sehr großen Messungenauigkeiten und somit zu einem sehr ungenauen Modell führt. Die mathematische Herleitung der Modelle für den Segway® ist bereits im Vorbericht [1, S. 1-24] ausführlich dargestellt und nachfolgend kurz zusammengefasst aufgeführt.

Die angestrebte Struktur der Regelung stellt eine Kaskadenregelung dar. Die innere Kaskade regelt den Pendelwinkel anhand des Gyrosensors und den Motoren. Diese Kaskade hält somit den Segway® aufrecht. Die äußere Kaskade regelt die Position des LEGO® MINDSTORMS® EV3-Roboters durch Vorgabe des Pendelwinkels für den inneren Regelkreis anhand der Encoder der Motoren. Die Herleitung des mathematischen Modells erfolgt deshalb anhand zwei zielgebender Zusammenhänge. Für die Auslegung der inneren Kaskade muss der Zusammenhang zwischen Motor-ansteuerungsgrad und Pendelwinkel bekannt sein. Für die Auslegung des äußeren Reglers muss der Motorwinkel abhängig vom Pendelwinkel gegeben sein.

5.1 Modellbildung

Lässt man die Möglichkeit außer acht, dass sich der Roboter durch die zwei getrennten Motoren um seine Achse drehen kann, ist es möglich, die Problemstellung auf eine zweidimensionale Betrachtung zu beschränken. Zudem sind die Räder und der Körper des LEGO® MINDSTORMS® EV3-Roboters vereinfacht als Starrkörper zu betrachten. Stellt man das Kräfte- und Momentengleichgewicht eines beliebigen Starrkörpers aus Abb. 5.2 auf, erhält man Gl. 5.1 bis Gl. 5.3.

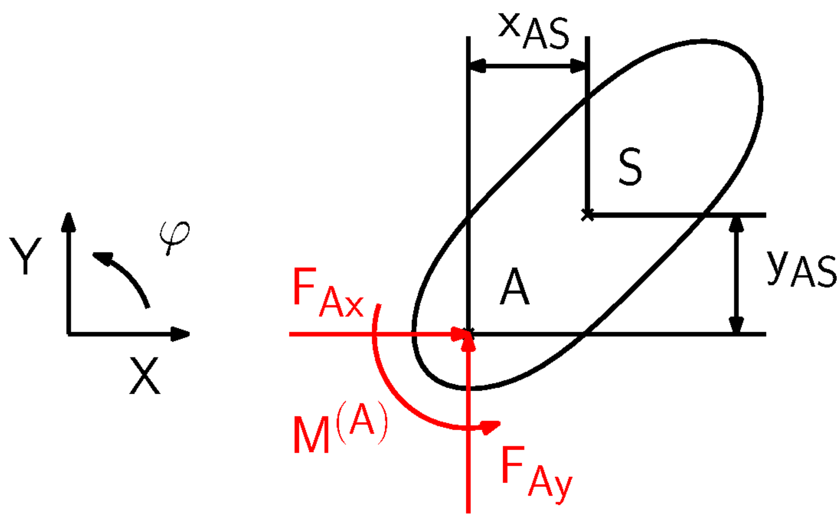


Abb. 5.2: Starrkörperbewegung [1, S. 1]

$$m \cdot a_{Ax} - m \cdot \ddot{\varphi} \cdot y_{AS} - m \cdot \dot{\varphi}^2 \cdot x_{AS} = F_{Ax} \quad \text{(Gl. 5.1)}$$

$$m \cdot a_{Ay} - m \cdot \ddot{\varphi} \cdot x_{AS} - m \cdot \dot{\varphi}^2 \cdot y_{AS} = F_{Ay} \quad \text{(Gl. 5.2)}$$

$$J^{(A)} \cdot \ddot{\varphi} - m \cdot a_{Ax} \cdot y_{AS} + m \cdot a_{Ay} \cdot x_{AS} = M^{(A)} \quad \text{(Gl. 5.3)}$$

In Abb. 5.2 ist ein Freischnitt dargestellt. Damit die, durch die vorangegangenen Gleichungen, berechneten Kräfte und Momente nachfolgend bei der Anwendung auf das

Rad und dem Pendelkörper zusammen mit den äußeren Kräften sinnvoll darstellbar sind, wird auf eine Darstellung der Freischnitte für die beiden Starrkörper verzichtet, wodurch die in Abb. 5.2 eingezeichneten Kräfte und Momente nachfolgend entgegengerichtet eingetragen sind. Zu den Gleichungen Gl. 5.1 bis Gl. 5.3 aus der Starrkörperbewegung sind somit für das Rad und den Pendelkörper zudem zwei weitere Kräftegleichgewichte und ein weiteres Momentengleichgewicht aufzustellen.

Auf das Rad in Abb. 5.3 wirken neben der Erdbeschleunigung im Schwerpunkt auch jeweils eine Kraft in X- und Y-Richtung und ein Moment über die Motorwelle. Die beiden zusätzlichen Kräftegleichgewichte und das Momentengleichgewicht sind in Gl. 5.4 bis Gl. 5.6 aufgestellt.

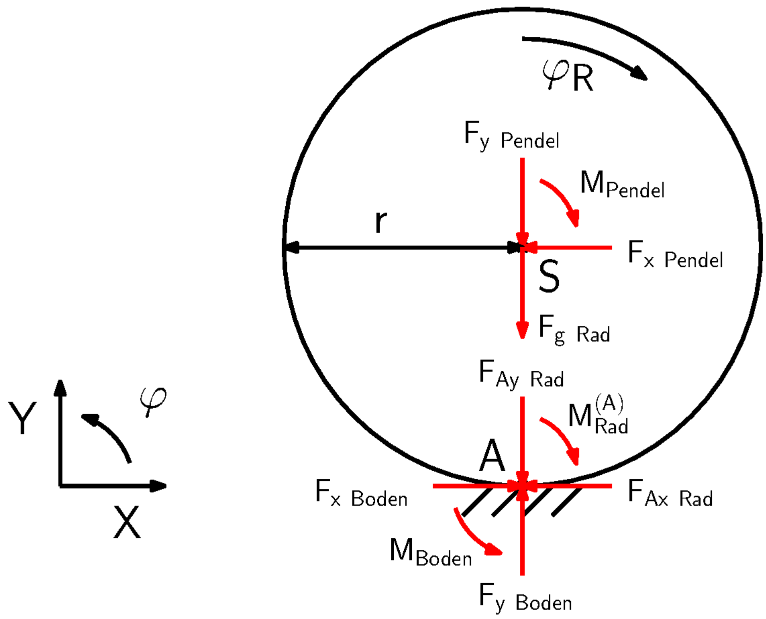


Abb. 5.3: Kräftegleichgewicht am Rad [1, S. 2]

$$F_{x\ Boden} = F_{Ax\ Rad} + F_{x\ Pendel} \tag{Gl. 5.4}$$

$$F_{y\ Boden} = F_{Ay\ Rad} + F_{y\ Pendel} + F_{g\ Rad} \tag{Gl. 5.5}$$

$$M_{Boden} + F_{x\ Pendel} \cdot r = M_{Rad}^{(A)} + M_{Pendel} \tag{Gl. 5.6}$$

$$m_R \cdot a_{Ax\ Rad} + m_R \cdot \ddot{\varphi}_R \cdot y_{AS\ Rad} - m_R \cdot \dot{\varphi}_R^2 \cdot x_{AS\ Rad} = F_{Ax\ Rad} \tag{Gl. 5.7}$$

$$m_R \cdot a_{Ay\ Rad} + m_R \cdot \ddot{\varphi}_R \cdot x_{AS\ Rad} - m_R \cdot \dot{\varphi}_R^2 \cdot y_{AS\ Rad} = F_{Ay\ Rad} \tag{Gl. 5.8}$$

$$-J_R^{(A)} \cdot \ddot{\varphi}_R - m_R \cdot a_{Ax\ Rad} \cdot y_{AS\ Rad} + m_R \cdot a_{Ay\ Rad} \cdot x_{AS\ Rad} = M_{Rad}^{(A)} \quad \text{(Gl. 5.9)}$$

Zur Vereinfachung der Zusammenhänge für das Roboter-Rad sind folgende Annahmen und Ersetzungen anzuwenden.

- Vernachlässigung der Terme höherer Ordnung.

$$\dot{\varphi}_R^2 = 0 \quad \text{(Gl. 5.10)}$$

- Aufgrund des Momentanpols des Rades im Kontaktpunkt zwischen Rad und Boden ist der Schwerpunkt immer genau über diesem Kontaktpunkt.

$$x_{AS\ Rad} = 0 \quad \text{(Gl. 5.11)}$$

$$y_{AS\ Rad} = r \quad \text{(Gl. 5.12)}$$

- Vernachlässigung der Rollreibung zwischen Rad und Boden.

$$M_{Boden} = 0 \quad \text{(Gl. 5.13)}$$

- Der Boden ist stationär, eben und waagrecht.

$$a_{Ax\ Rad} = a_{Ay\ Rad} = 0 \quad \text{(Gl. 5.14)}$$

- Die Schwerkraft des Rades kann durch das Produkt aus Masse und Erdbeschleunigung ersetzt werden.

$$F_{g\ Rad} = m_R \cdot g \quad \text{(Gl. 5.15)}$$

Setzt man zudem die Zusammenhänge aus der Starrkörperbewegung in die Kräfte- und Momentengleichgewichte ein, ergeben sich Gl. 5.16 bis Gl. 5.18.

$$F_{x\ Boden} = m_R \cdot \ddot{\varphi}_R \cdot r + F_{x\ Pendel} \quad \text{(Gl. 5.16)}$$

$$F_{y\ Boden} = F_{y\ Pendel} + m_R \cdot g \quad \text{(Gl. 5.17)}$$

$$M_{Pendel} = F_{x\ Pendel} \cdot r + J_R^{(A)} \cdot \ddot{\varphi}_R \quad \text{(Gl. 5.18)}$$

Selbiges Vorgehen ist in identischer Weise auf das Pendel anzuwenden. Dieses ist in Abb. 5.4 dargestellt und hat nur die Schwerkraft als einzige äußere Kraft. Die zusätzlichen Gleichgewichte sind in Gl. 5.19 bis Gl. 5.21 gegeben.

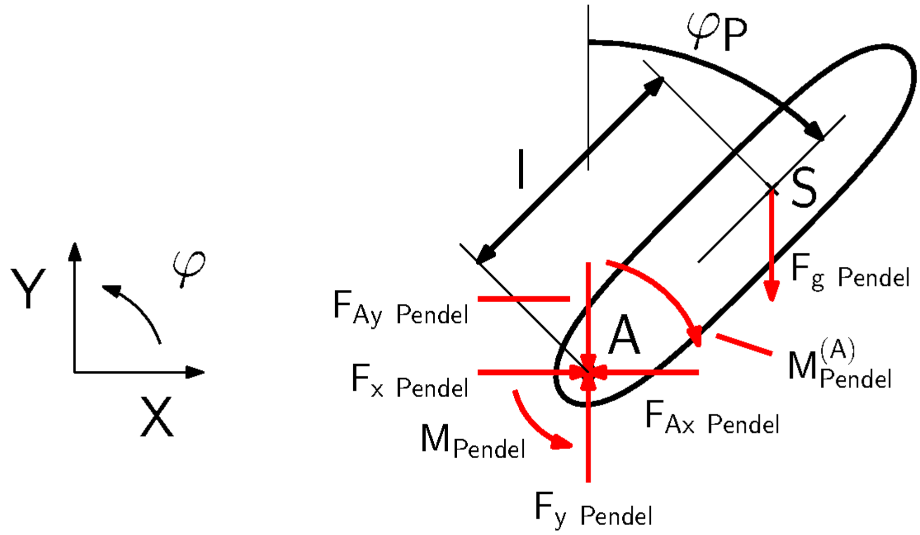


Abb. 5.4: Kräftegleichgewicht am Pendel [1, S. 4]

$$F_{x\ Pendel} = F_{Ax\ Pendel} \tag{Gl. 5.19}$$

$$F_{y\ Pendel} = F_{Ay\ Pendel} + F_{g\ Pendel} \tag{Gl. 5.20}$$

$$M_{Pendel} = M^{(A)}_{Pendel} + sin(\varphi_P) \cdot l \cdot F_{g\ Pendel} \tag{Gl. 5.21}$$

$$m_P \cdot a_{Ax\ Pendel} + m_P \cdot \ddot{\varphi}_P \cdot y_{AS\ Pendel} - m_P \cdot \dot{\varphi}_P^2 \cdot x_{AS\ Pendel} = F_{Ax\ Pendel} \tag{Gl. 5.22}$$

$$m_P \cdot a_{Ay\ Pendel} + m_P \cdot \ddot{\varphi}_P \cdot x_{AS\ Pendel} - m_P \cdot \dot{\varphi}_P^2 \cdot y_{AS\ Pendel} = F_{Ay\ Pendel} \tag{Gl. 5.23}$$

$$-J_P^{(A)} \cdot \ddot{\varphi}_P - m_P \cdot a_{Ax\ Pendel} \cdot y_{AS\ Pendel} + m_P \cdot a_{Ay\ Pendel} \cdot x_{AS\ Pendel} = M^{(A)}_{Pendel} \tag{Gl. 5.24}$$

Bei der Zusammenfassung der Gleichungen des Pendels sind auch, wie beim Rad, bestimmte Vereinfachungen und Ersetzungen zu tätigen. Diese sind in nachfolgender Aufstellung gezeigt:

- Vernachlässigung der Terme höherer Ordnung.

$$\dot{\varphi}_P^2 = 0 \tag{Gl. 5.25}$$

- Der Boden ist stationär, eben und waagrecht.

$$a_{Ay\ Pendel} = 0 \tag{Gl. 5.26}$$

- Ersetzen mehrerer Unbekannter durch entsprechende geometrische Zusammenhänge.

$$x_{AS\ Pendel} = sin(\varphi_P) \cdot l \tag{Gl. 5.27}$$

$$y_{AS\ Pendel} = cos(\varphi_P) \cdot l \tag{Gl. 5.28}$$

$$a_{Ax\ Pendel} = \ddot{\varphi}_R \cdot r \tag{Gl. 5.29}$$

- Die Schwerkraft des Pendels kann durch das Produkt aus Masse und Erdbeschleunigung ersetzt werden.

$$F_{g\ Pendel} = m_P \cdot g \tag{Gl. 5.30}$$

- Anwenden der Kleinwinkelnäherung.

$$sin(\varphi_P) = \varphi_P \tag{Gl. 5.31}$$

$$cos(\varphi_P) = 1 \tag{Gl. 5.32}$$

Durch die gegebenen Vereinfachungen und durch Einsetzen der Zusammenhänge aus der Starrkörperbewegung in die Kräfte- und Momentengleichgewichte ergeben sich wie beim Rad folgende drei Gleichungen.

$$F_{x\ Pendel} = m_P \cdot \ddot{\varphi}_R \cdot r + m_P \cdot \ddot{\varphi}_P \cdot l \tag{Gl. 5.33}$$

$$F_{y\ Pendel} = m_P \cdot \ddot{\varphi}_P \cdot \varphi_P \cdot l + m_P \cdot g \tag{Gl. 5.34}$$

$$M_{Pendel} = -J_P^{(A)} \cdot \ddot{\varphi}_P - m_P \cdot \ddot{\varphi}_R \cdot r \cdot l + \varphi_P \cdot l \cdot m_P \cdot g \tag{Gl. 5.35}$$

Setzt man Gl. 5.18 und Gl. 5.35 gleich, ersetzt $F_{x\ Pendel}$ durch Gl. 5.33 und führt eine Laplace-Transformation durch, erhält man abschließend den Zusammenhang aus Gl. 5.36 mit den in Gl. 5.37 bis Gl. 5.39 dargestellten Vereinfachungen.

$$\frac{\varphi_R}{\varphi_P} = \frac{A \cdot s^2 + B}{C \cdot s^2} \tag{Gl. 5.36}$$

$$A = -m_P \cdot l \cdot r - J_P^{(A)} \qquad \text{(Gl. 5.37)}$$

$$B = m_P \cdot l \cdot g \qquad \text{(Gl. 5.38)}$$

$$C = m_P \cdot r^2 + J_R^{(A)} + m_P \cdot r \cdot l \qquad \text{(Gl. 5.39)}$$

Für die Berechnung der Faktoren sind mehrere unbekannte Systemvariablen nötig. Die Massen der Räder und des Pendels sind mit einer Waage zu bestimmen. Aufgrund der zweidimesionalen Betrachtung ist die Masse beider Räder zu addieren. Die Räder wiegen 66 g, das Pendel 754 g. Der Radius der Räder beträgt 28 mm. Dabei wird die Kompression der Elastomer-Lauffläche vernachlässigt. Etwas aufwändiger ist die Bestimmung des Schwerpunktabstandes l des Pendels. Mit einem einfachen Messaufbau mit einer Küchenwaage kann dieser jedoch sehr genau bestimmt werden. Der Aufbau ist in Abb. 5.5 dargestellt. Stellt man das Momentengleichgewicht an der Motorwelle wie in Gl. 5.40 auf, kann durch Umformen, wie in Gl. 5.41, der Schwerpunktabstand berechnet werden.

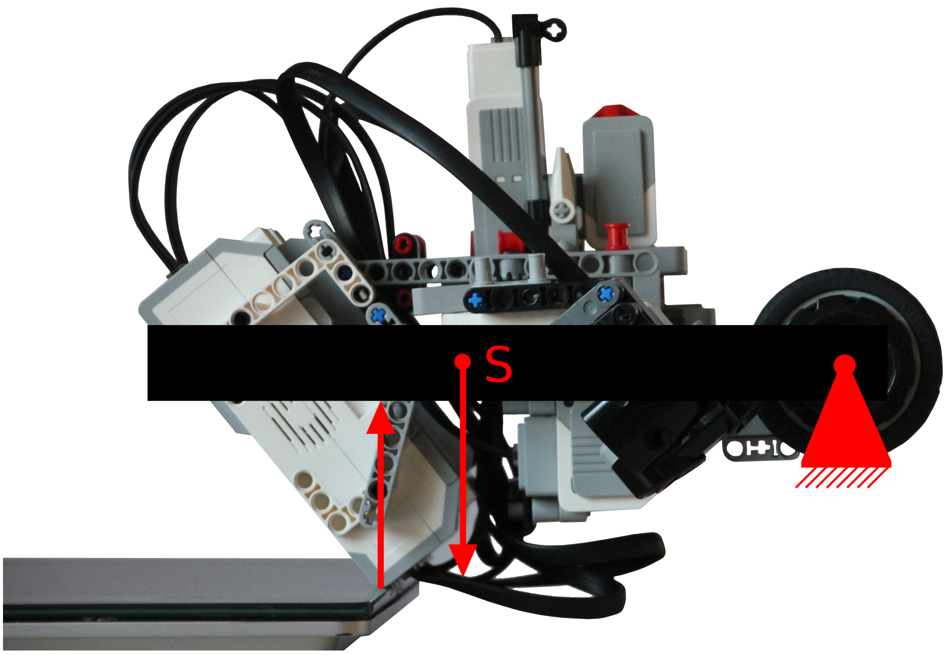


Abb. 5.5: Bestimmung des Schwerpunktabstandes l des Pendels [1, S. 9]

$$M = 0 = m_P \cdot g \cdot l - m_{Waage} \cdot g \cdot l_{Waage} \qquad \text{(Gl. 5.40)}$$

$$l = \frac{m_{Waage}}{m_p} \cdot l_S = \frac{0,614}{0,754} \cdot 132,5 = 107,9\ [mm] \qquad \text{(Gl. 5.41)}$$

Die Räder sind vereinfacht als Vollzylinder anzunehmen, da durch weitere kleinere Räder im Inneren der Felge diese fast komplett ausgefüllt sind. Das Trägheitsmoment der Räder $J_R^{(A)}$ bezieht sich auf den Kontaktpunkt am Boden, weshalb zusätzlich der steinersche Anteil berücksichtigt werden muss. Die Berechnung erfolgt laut Gl. 5.42 [3, S. M2 + M3].

$$J_R^{(A)} = \frac{1}{2} \cdot m_R \cdot r^2 + m_R \cdot r^2 = \frac{3}{2} \cdot 0,066 \cdot 0,028^2 = 7,762 \cdot 10^{-5}\ [Nms^2] \qquad \text{(Gl. 5.42)}$$

Die Bestimmung des Trägheitsmoments des Pendelkörpers erfolgt anhand des berechneten, mathematischen Modells aus Gl. 5.36. Lässt man den Roboter von seiner senkrechten Position aus umfallen, ohne dass die Motoren angesteuert werden, ist der Radwinkel gleich dem Pendelwinkel. Ersetzt man den Radwinkel durch den Pendelwinkel und formt die Gleichung um, erhält man Gl. 5.43, welche eine homogene, lineare Differentialgleichung zweiter Ordnung darstellt.

$$\varphi_P \cdot s^2 + \frac{B}{A - C} \cdot \varphi_P = 0 \qquad \text{(Gl. 5.43)}$$

Die allgemeine Lösung der Differentialgleichung ist in Gl. 5.44 [3, S. J5] dargestellt.

$$\varphi_P = C_1 \cdot sin(\omega \cdot t) + C_2 \cdot cos(\omega t) \qquad \text{mit:} \qquad \omega = \sqrt{\frac{B}{A - C}} \qquad \text{(Gl. 5.44)}$$

Führt man eine Messung des Pendelwinkels beim Fall des Roboters aus einer fast aufrechten Position durch, erhält man eine Kurve, aus der zwei Messwerte zu entnehmen sind. Die verwendeten Messwerte [1, S. 12] sind in Tab. 5.1 dargestellt. Die dritte nötige Bedingung zur Berechnung des Trägheitsmoments des Pendels ist die Anfangsbedingung. Da der Roboter aus dem Stillstand fallen gelassen wird, ist die Pendelwinkelgeschwindigkeit zum Zeitpunkt $t = 0$ ebenfalls null. Setzt man die beiden Messwerte in die allgemeine Lösung und die Anfangsbedingung in die erste Ableitung der allgemeinen Lösung ein, erhält man ein Gleichungssystem mit drei Gleichungen und drei unbekannte Variablen. Mit diesem Gleichungssystem ergibt sich ω zu $\pm\ 7,1385\ i$.

Tab. 5.1: Messstellen aus der Pendelwinkelmessung

Messstelle	Zeit [s]	Pendelwinkel [rad]
1	0,19	0,2678
2	0,332	0,6981

Setzt man ω in den entsprechenden Ausdruck aus Gl. 5.44 ein und löst das System mit eingesetzten Faktoren nach dem Trägheitsmoment des Pendels auf, erhält man Gl. 5.45, mit der das Trägheitsmoment berechnet werden kann.

$$\begin{aligned} J_P^{(A)} &= \frac{-m_P \cdot l \cdot g}{\omega^2} - 2 \cdot m_P \cdot l \cdot r - m_P \cdot r^2 - J_R^{(A)} = ... \\ ... &= \frac{-0,754 \cdot 0,1079 \cdot 9,81}{(\pm\, 7,1385\ i)^2} - 2 \cdot 0,754 \cdot 0,1079 \cdot 0,028 - ... \\ & ... - 0,754 \cdot 0,028^2 - 7,762 \cdot 10^{-5} = 1,044 \cdot 10^{-2}\ [kgm^2] \end{aligned} \quad \text{(Gl. 5.45)}$$

Bei dieser Herleitung ist der Radwinkel integriert, welcher für die Regelung der Position ausschlaggebend ist. Der Radwinkel bezeichnet den Winkel der Rades ausgehend von seiner Anfangsposition bezogen auf den Boden. Da dieser Winkel nicht erfassbar ist, wird der Radwinkel durch den Motorwinkel ersetzt. Der Motorwinkel ist der Winkel des Rades ausgehend von der Anfangsposition bezogen auf das Pendel. Dies hat den Nachteil, dass die Positionsregelung des Segways® durch geringe Abweichungen im Pendelwinkel von der aufrechten Position etwas ungenauer ist, jedoch ist der Motorwinkel über die Encoder in den Motoren einfach messbar. Die Beziehung zwischen Motor-, Pendel- und Radwinkel ist in Gl. 5.46 dargestellt.

$$\varphi_M = \varphi_R - \varphi_P \quad \text{(Gl. 5.46)}$$

Für die Ansteuerung der Motoren ist die Geschwindigkeit in Prozent anzugeben. Gesucht ist der Zusammenhang zwischen Ansteuerungsgrad und Pendelwinkel. Aus der obigen Herleitung in Verbindung mit Gl. 5.46 ist der Pendelwinkel abhängig von der Motordrehzahl berechenbar. Somit fehlt für die Berechnung der ersten gesuchten Übertragungsfunktion die Verbindung zwischen Ansteuerungsgrad und Drehzahl. Die verbauten Motoren sind Gleichstrommotoren. Die entsprechenden physikalischen Zusammenhänge sind in Gl. 5.47 [7, S. 20], Gl. 5.48 [7, S. 20] und Gl. 5.49 [7, S. 20] dargestellt.

$$U_{Anker} = R_{Anker} \cdot I_{Anker} + L_{Anker} \cdot \frac{dI_{Anker}}{dt} + U_{induziert} \quad \text{(Gl. 5.47)}$$

$$U_{induziert} = c \cdot \phi \cdot \omega \quad \text{(Gl. 5.48)}$$

$$M_{innen} = c \cdot \phi \cdot I_{Anker} = M_{Last} + M_{Schlepp} + J_{Anker} \cdot \dot{\omega} \quad \text{(Gl. 5.49)}$$

Verwendet man diese Gleichungen zur Berechnung der Ankerspannung und eliminiert den Ankerstrom erhält man den Zusammenhang aus Gl. 5.50. Das Lastmoment kann mit dem obigen mathematischen Modell aus dem Motorwinkel berechnet werden. Ersetzt man das Lastmoment in Gl. 5.50 durch den entsprechenden mathematischen Zusammenhang des Segways® und vernachlässigt das Schleppmoment, kann abhängig von der Ankerspannung die Motordrehzahl unter Berücksichtigung des Lastmoments simuliert werden. Lässt man die Motoren am frei und aufrecht aufgestellten, realen Roboter für 0,09 Sekunden mit 100 Prozent Ansteuerung nach vorne drehen und ändert dann die Ansteuerung schlagartig auf - 100 Prozent, können die Motoren den Roboter gerade noch wieder aufrichten. Dieser Fall stellt somit einen maximalen Belastungsfall für die Motoren dar. Simuliert man diese Belastung mit dem Modell, mit Berücksichtigung des Lastmoments, und einem weiteren Modell, ohne Berücksichtigung des Lastmoments, kann der Fehler auf unter 0,8 Prozent eingeschränkt werden. Durch die Integration eines idealen LEGO® MINDSTORMS® EV3-Roboter-Modells kann die Formel mit dem integrierten Lastmoment nicht für die weitere Auslegung verwendet werden. Da der Fehler bei der Vernachlässigung des Lastmoments im Maximalfall nur 0,8 Prozent beträgt, ist dieses vernachlässigbar.

$$\begin{aligned} U_{Anker} = {} & \frac{L_{Anker} \cdot J_{Anker}}{c \cdot \phi} \cdot \ddot{\omega} + \frac{R_{Anker} \cdot J_{Anker}}{c \cdot \phi} \cdot \dot{\omega} + c \cdot \phi \cdot \omega + \ \ldots \\ & \ldots \ + \left(\frac{R_{Anker} \cdot (M_{Last} + M_{Schlepp}) + L_{Anker} \cdot \dot{M}_{Last}}{c \cdot \phi} \right) \end{aligned} \quad \text{(Gl. 5.50)}$$

Geht man davon aus, dass die Ankerspannung direkt proportional zum Ansteuerungsgrad ist, vernachlässigt das Schlepp- und das Lastmoment und führt eine Laplace-Transformation durch, erhält man das Motormodell aus Gl. 5.51. Aufgrund der schweren Bestimmbarkeit der Ankerinduktivität und des Trägheitsmoments des Ankers soll hier auf eine genaue Bestimmung der Systemvariablen verzichtet werden.

$$\frac{\dot{\varphi}_M}{S_{Motor}} = \frac{1}{\frac{L^*_{Anker} \cdot J^*_{Anker}}{c^* \cdot \phi^*} \cdot s^2 + \frac{R^*_{Anker} \cdot J^*_{Anker}}{c^* \cdot \phi^*} \cdot s + c^* \cdot \phi^*} \tag{Gl. 5.51}$$

Zudem ist der Aufwand für die Berücksichtigung des Getriebeverhaltens zu groß. Deshalb wird der Motor mit 100 Prozent angesteuert und die Drehzahl aufgezeichnet. Mit pzMove und dem Identifier kann daraus ein Modell für das Übertragungsverhalten des Motors berechnet werden. Verwendet man vereinfacht die PT2-Form aus Gl. 5.51 erhält man die Übertragungsfunktion aus Gl. 5.52. Der Vergleich zwischen Messung und Simulation mit der identifizierten Übertragungsfunktion ist in Abb. 5.6 dargestellt.

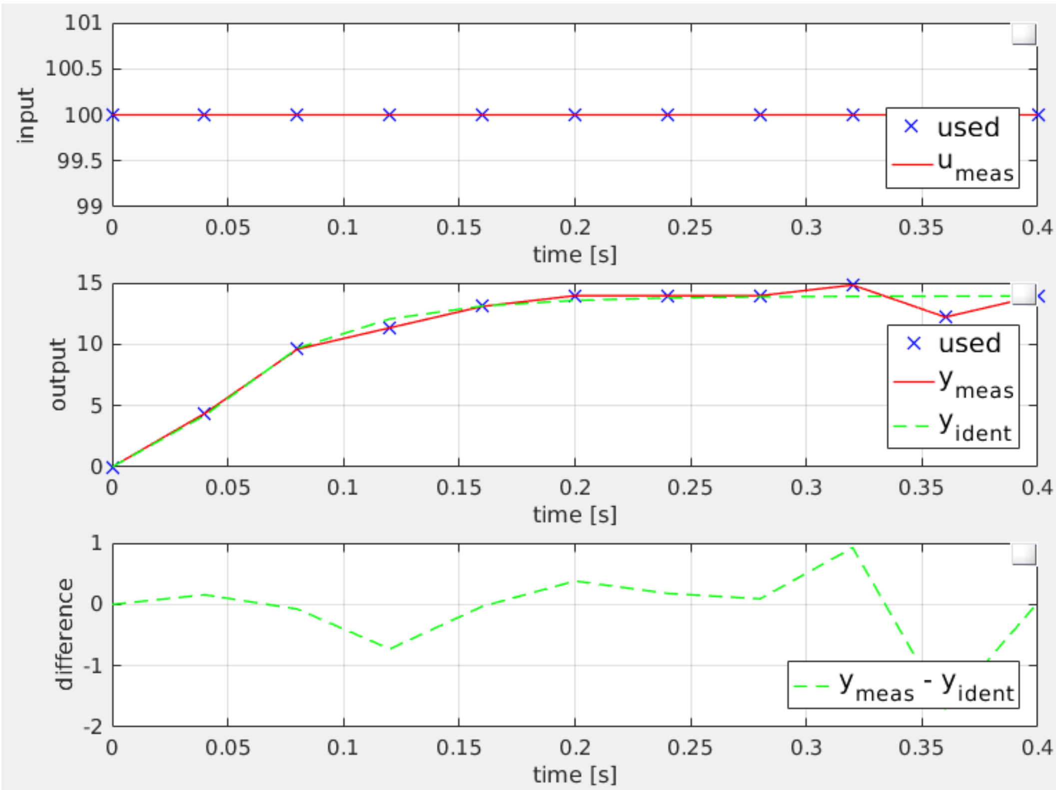


Abb. 5.6: Vergleich zwischen gemessener Motordrehzahl und Simulation mit der identifizierten Übertragungsfunktion

$$\frac{\dot{\varphi}_M}{S_{Motor}} = \frac{621,51}{s^2 + 236,75 \cdot s + 4458,4} \tag{Gl. 5.52}$$

An diesem Punkt sind alle nötigen Zusammenhänge bekannt, sodass die gesuchten Übertragungsfunktionen berechenbar sind. Für die erste Kaskade ist die

Übertragungsfunktion zur Berechnung des Pendelwinkels aus der Motoransteuerung nötig. Stellt man die gesuchte Übertragungsfunktion auf, erweitert diese, wie in Gl. 5.53, mit $\dot{\varphi}_M/\dot{\varphi}_M$ und setzt Gl. 5.46 ein, kann man diese Gleichung so umformen, dass Gl. 5.36 und Gl. 5.52 direkt eingesetzt werden können. Durch Einsetzen der entsprechenden Werte und Umformen erhält man die gewünschte Funktion wie in Gl. 5.54.

$$G_{Sys1}(s) = \frac{\varphi_P}{S} = \frac{\dot{\varphi}_M}{S} \cdot \frac{\varphi_P}{\dot{\varphi}_M} = \frac{\dot{\varphi}_M}{S} \cdot \frac{\varphi_P}{\varphi_M \cdot s} = \frac{\dot{\varphi}_M}{S} \cdot \frac{\varphi_P}{(\varphi_R - \varphi_P) \cdot s} = \frac{\dot{\varphi}_M}{S} \cdot \frac{1}{(\frac{\varphi_R}{\varphi_P} - 1) \cdot s} \tag{Gl. 5.53}$$

$$G_{Sys1}(s) = \frac{\varphi_P}{S} = \frac{-1,831 \cdot s}{0,01566 \cdot s^4 + 3,709 \cdot s^3 + 69,04 \cdot s^2 - 189 \cdot s - 3558} \tag{Gl. 5.54}$$

Für die zweite Kaskade ist die Übertragungsfunktion zum Berechnen des Motorwinkels aus dem Pendelwinkel gesucht. Auch hier ist durch Aufstellen der Übertragungsfunktion und Einsetzen von Gl. 5.46 eine Form erreichbar, in der der Zusammenhang aus Gl. 5.36 direkt eingesetzt werden kann. Mit eingesetzten Werten ergibt sich die gesuchte Übertragungsfunktion wie in Gl. 5.56.

$$G_{Sys2}(s) = \frac{\varphi_M}{\varphi_P} = \frac{\varphi_R - \varphi_P}{\varphi_P} = \frac{\varphi_R}{\varphi_P} - 1 \tag{Gl. 5.55}$$

$$G_{Sys2}(s) = \frac{\varphi_M}{\varphi_P} = \frac{-0,01566 \cdot s^2 + 0,7981}{0,002947 \cdot s^2} \tag{Gl. 5.56}$$

5.2 Auslegung des inneren Reglers

Da dieses Beispiel zur Demonstration der Anwendung der Kaskadenregelung und des Loop-Shaping-Verfahrens mit pzMove dienen soll, müssen die Übertragungsfunktionen in pzMove entsprechend vordefiniert werden. Als Modell und als Strecke ist für die erste Kaskade die Übertragungsfunktion aus Gl. 5.54 nötig. Alle anderen Übertragungsfunktionen sind mit 1 vorzubelegen.

$$GR = GF = GV = GC = GD = 1 \tag{Gl. 5.57}$$

$$GM = GS = G_{Sys1}(s) \tag{Gl. 5.58}$$

Zur Anwendung kommt die Schleifenoptimierung aus Abschnitt 4.2. Dazu wählt man im Bereich „Controller design“ im gleichnamigen Panel im Popup für die Methode der Reglerauslegung „loopshaping“ aus. Nachfolgend wird auf die genaue Beschreibung der einzelnen Schritte in pzMove verzichtet, da die Umsetzung des allgemein beschriebenen Vorgehens anhand der Beschreibung in Abschnitt 4.4 in pzMove eindeutig ist. Die ersten beiden Schritte umfassen die Berechnung des geforderten Umschlingungswinkels. Dies übernimmt pzMove für den Anwender. Bei diesem Beispiel ist ein Umschlingungswinkel von 180 Grad gefordert, da die Strecke einen instabilen Pol hat und keine Pole auf der imaginäre Achse. Deshalb ist ein zusätzlicher Pol in der rechten Halbebene nötig. Das Nyquist-Diagramm der Strecke, der inneren Kaskade, ist in Abb. 5.7 dargestellt. Betrachtet man die Auswirkungen eines zusätzlichen instabilen Pols aus Abb. 4.24, so fällt auf, dass der Pol bei einer instabilen Strecke den Endbereich der Kurve gegen den Uhrzeigersinn dreht und so positiv zur geforderten Umschlingung beiträgt.

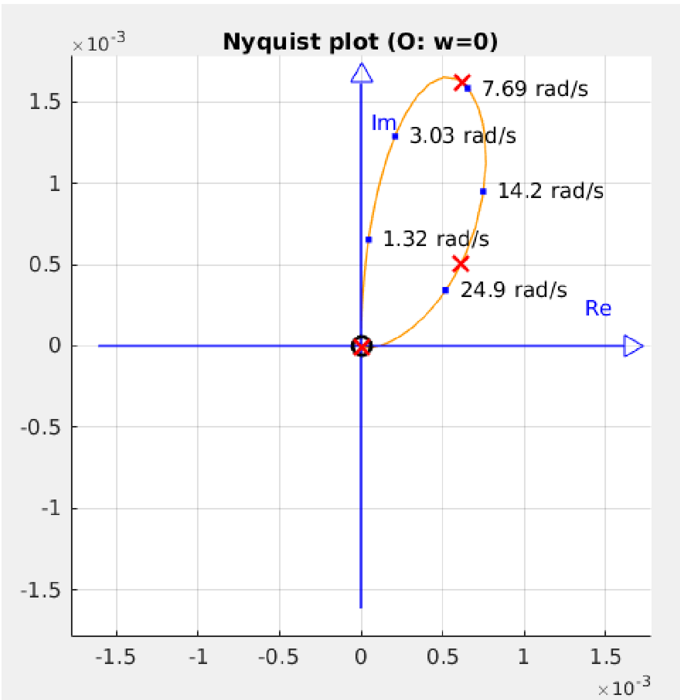


Abb. 5.7: Nyquist-Diagramm der ersten Strecke ohne Regler

Fügt man den Pol bei einer Frequenz von 1 rad/s ein, erhält man bereits eine Nyquist-

Kurve, die deutlich weiter in Richtung der negativen, realen Achse gebogen ist. Diese ist in Abb. 5.8 dargestellt. Die Kurve ist im höheren Frequenzbereich „links“ vom Kurvenbereich der niedrigeren Frequenz, wodurch die Umschlingung bei weiterem Ausbau der Schlaufe wie gefordert gegen den Uhrzeigersinn erfolgt.

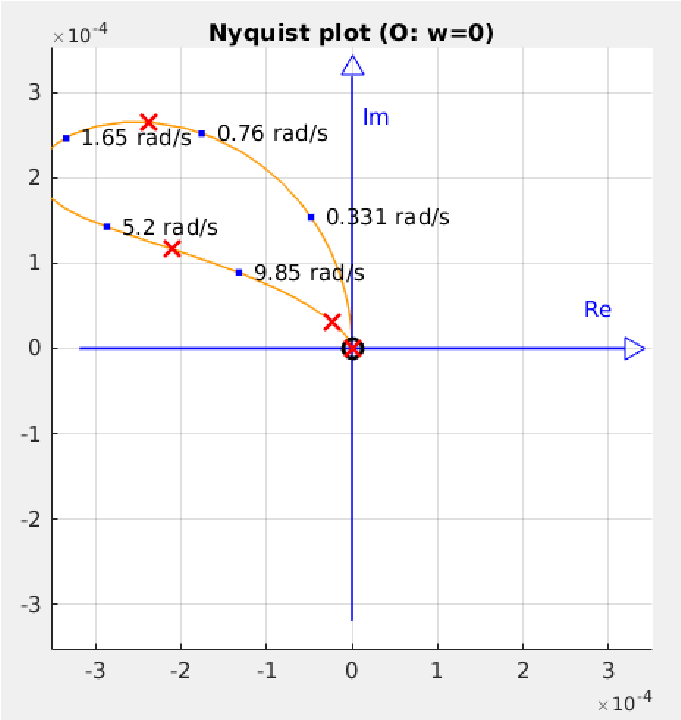


Abb. 5.8: Nyquist-Diagramm der offenen, inneren Kaskade mit zusätzlichem instabilen Pol

Damit die Kurve noch weiter gegen den Uhrzeigersinn ausgebaut werden kann, ist ein Lead-Glied erforderlich. Bei diesem Korrekturglied ist der Bereich zwischen den Grenzfrequenzen gegen den Uhrzeigersinn gedreht, wodurch in diesem Fall der vordere Kurvenbereich dazwischen liegen muss. So können die Grenzfrequenzen des Lead-Glieds bereits aus Abb. 5.8 etwa auf 2 rad/s und 10 rad/s festgelegt werden. Damit die Drehung und somit die Umschlingung größer ausfällt, ist die höhere Grenzfrequenz weiter auf etwa 100 rad/s zu vergrößern. Das Ergebnis ist in Abb. 5.9 erkennbar.

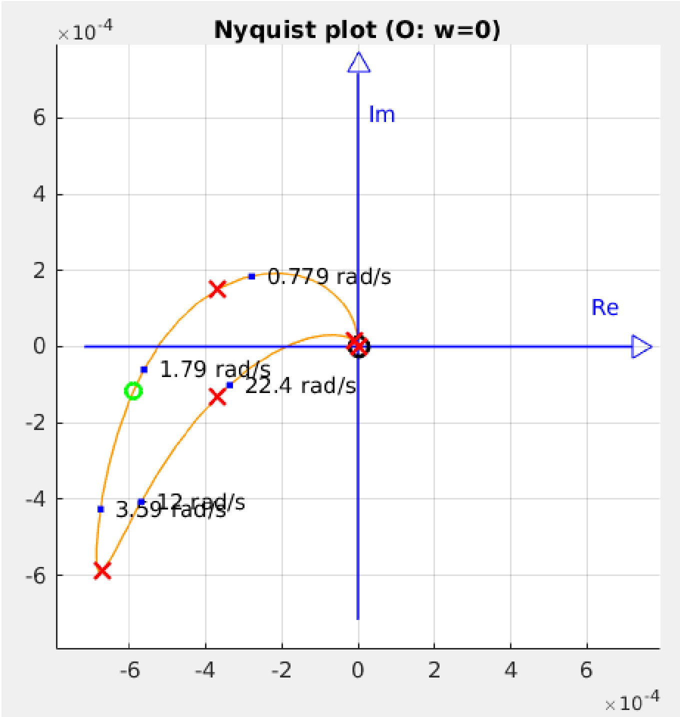


Abb. 5.9: Nyquist-Diagramm der offenen, inneren Kaskade mit zusätzlichem Lead-Glied

Die Nyquist-Kurve umschlingt in etwa den Punkt P(-0,00038,0). Im weiteren Vorgehen ist die Kurve mit einem P-Glied mit der Verstärkung 1/0,00038=2632 zu skalieren. Dadurch umschlingt die Kurve, wie in Abb. 5.10 zu sehen, den kritischen Punkt, wodurch Stabilität für den geschlossenen Regelkreis gegeben ist.

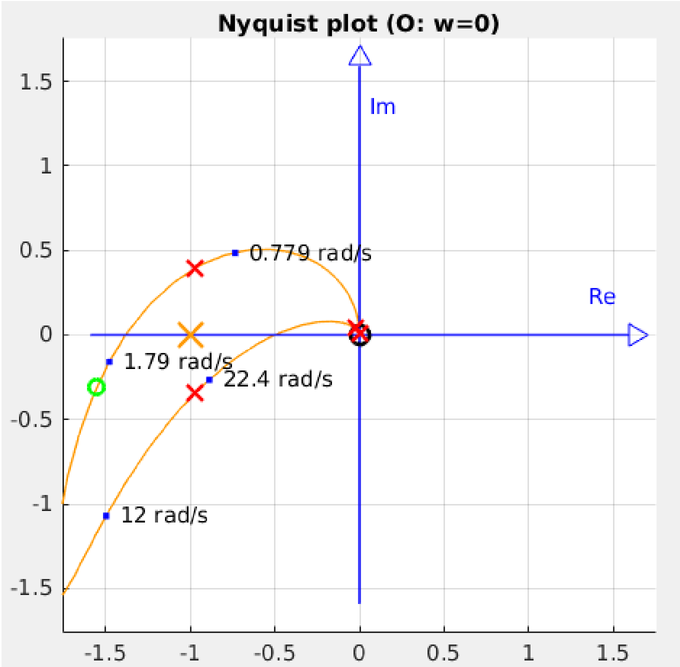


Abb. 5.10: Nyquist-Diagramm der offenen, inneren Kaskade mit zusätzlichem P-Glied

Die Nyquist-Kurve beginnt noch im Ursprung, wodurch keine stationäre Genauigkeit gegeben ist. Durch Einfügen von zwei weiteren PI-Gliedern mit einem Verstärkungsfaktor von 2, beginnt die Kurve im negativen, imaginären Unendlichen und erfüllt dadurch die stationäre Genauigkeit. Auch Stabilität ist weiterhin gegeben. Durch den instabilen Pol der Strecke, den instabilen Pol des Reglers und die zwei PI-Glieder fordert das Nyquist-Kriterium 1,5 Umschlingungen. Da die Strecke jedoch eine Nullstelle im Ursprung hat, wird ein Pol im Ursprung, den die PI-Glieder erzeugen, gekürzt. Deshalb sind vom Nyquist-Kriterium nicht 1,5 sondern nur 1,25 Umschlingungen gefordert. Damit die Stabilität abschließend noch maximiert werden kann, ist die Verstärkung des P-Glieds noch auf 1350 anzupassen. Das Nyquist-Diagramm der offenen, ersten Kaskade ist in Abb. 5.11 dargestellt.

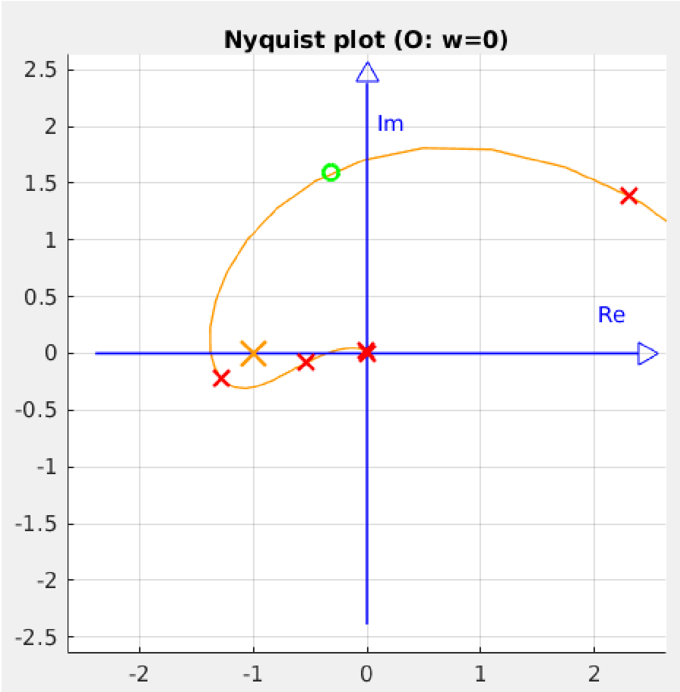


Abb. 5.11: Nyquist-Diagramm der offenen, inneren Kaskade mit zwei zusätzlichen PI-Gliedern

Das Verhalten des geschlossenen Regelkreises ist in Abb. 5.12 simuliert. Die Sprungantwort zeigt deutliches Überschwingen, aber auch schnelles Abklingen der Schwingung und stationäre Genauigkeit.

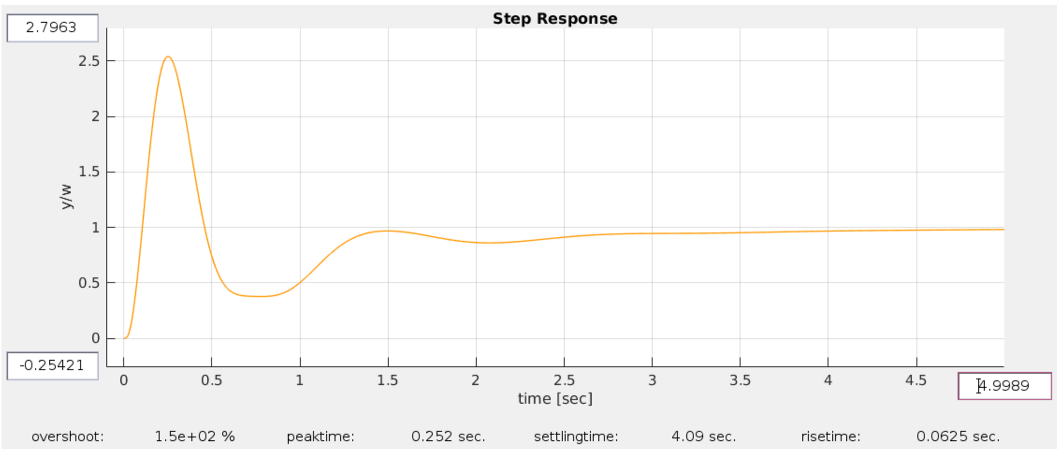


Abb. 5.12: Sprungantwort des geschlossenen Regelkreises der ersten Kaskade

Im ersten Optimierungsschritt ist der Abstand der Nyquist-Kurve zum kritischen Punkt zu erhöhen. Steigert man die obere Grenzfrequenz des Lead-Glieds auf 200, ist der Endbereich weiter gegen den Uhrzeigersinn gedreht und somit vor allem der Abstand in diesem Bereich vergrößert. Durch ein weiteres Lead-Glied, mit selbigen Grenzfrequenzen, in Verbindung mit der Erhöhung der Verstärkungsfaktoren der PI-Glieder auf $k_i = 12$ ist die Schlaufe weiter zu vergrößern. Eine abschließende Reduktion der Verstärkung des P-Glieds auf 200 sorgt für einen maximalen Abstand zwischen der Nyquist-Kurve und dem kritischen Punkt. Das Ergebnis ist in Abb. 5.13 zu sehen. Die daraus resultierende Sprungantwort ist in Abb. 5.14 gegeben.

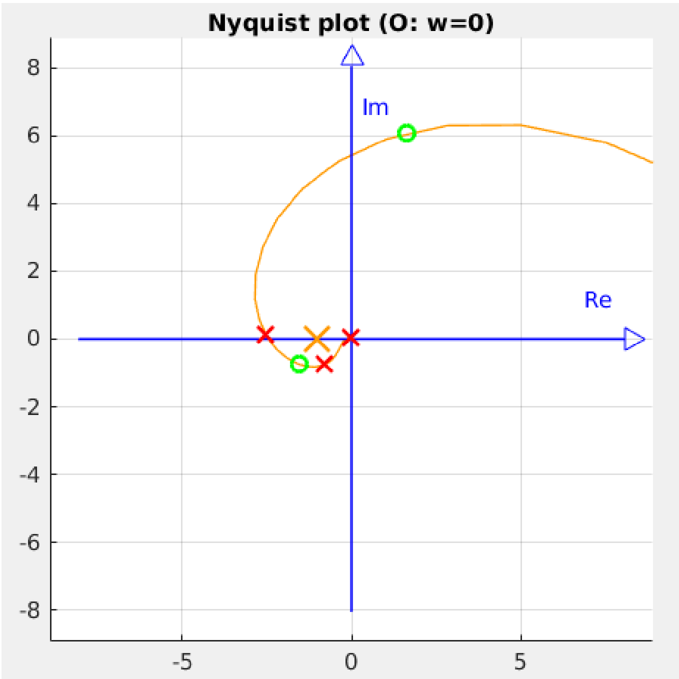


Abb. 5.13: Nyquist-Diagramm der offenen, inneren Kaskade mit optimiertem Regler

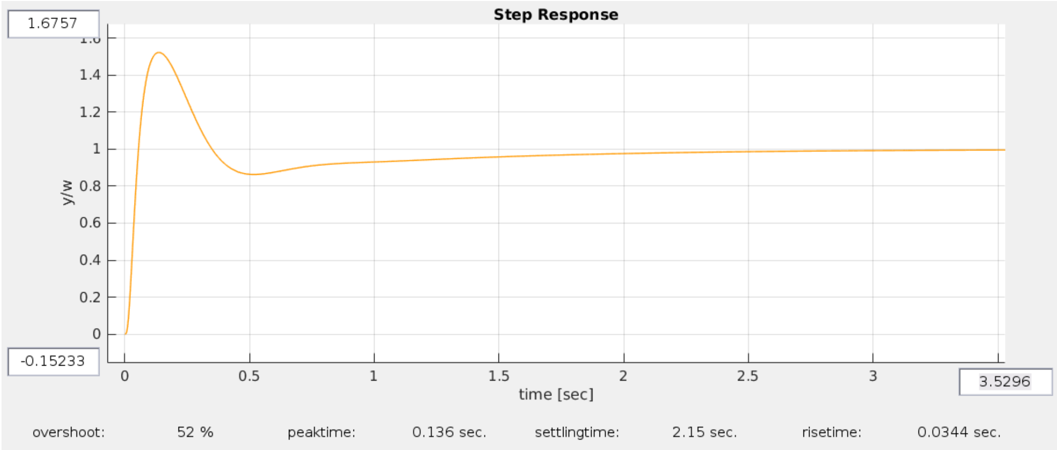


Abb. 5.14: Sprungantwort des geschlossenen Regelkreises der ersten Kaskade mit optimiertem Regler

Für die zweite Optimierung muss die Führungsgröße oder eine Störung gegeben sein. Da die Führungsgröße immer annähernd 0 ist, ist eine Störung festzulegen, anhand derer die Stellgrößenoptimierung vorzunehmen ist. Als Störung wird ein Stoß mit einem Finger angenommen, bei dem sich der Roboter innerhalb von 0,1 s um 12 mm radial nach vorne beugt. Die Stoßhöhe wird auf 200 mm über der Radwelle festgelegt. Die Welle bewegt sich dabei um $\varphi \cdot r$ nach vorne. Zusätzlich beugt sich der Pendelkörper um $\varphi \cdot l_{H\ Stoß}$ radial nach vorne. Addiert man beide Bewegungen erhält man Gl. 5.59, mit der man den Stoßwinkel berechnen kann. Setzt man für $l_{H\ Stoß}$ 200 mm, für r 28 mm und für $l_{Stoß}$ 12 mm ein, ergibt sich der Stoßwinkel zu 0,0526 rad oder 3 Grad. Die Stoßlänge von 0,1 s wird mit einem PT2-Verhalten angenähert. Der Vorfilter GD der Störung d in pzMove ist dazu mit Gl. 5.60 vorzubelegen.

$$\varphi \cdot r + \varphi \cdot l_{H\ Stoß} = l_{Stoß} \qquad \text{(Gl. 5.59)}$$

$$GD(s) = \frac{4900}{(s+70)\cdot(s+70)} \qquad \text{(Gl. 5.60)}$$

Zur Simulation ist bei Arbitrary input response im Eingabefeld der Störung d „0.0526/s“ einzugeben. Das Ergebnis der Stoßsimulation ist in Abb. 5.15 gezeigt. Zu Beginn der Störung ist fast keine Differenz zwischen Störung und Pendelwinkel festzustellen. Im weiteren Verlauf beschleunigen hingegen die Motoren nach vorne und der Pendelkörper erreicht eine maximale Auslenkung von 0,03 rad.

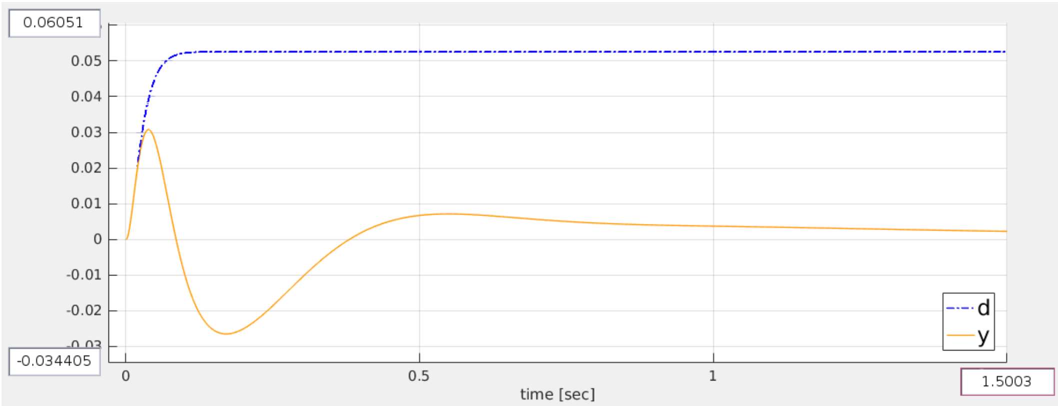


Abb. 5.15: Simulation des angenommenen Stoßes am geschlossenen Regelkreis

Die bei der Simulation erreichte maximale Stellgröße beträgt 80 %. Somit ist keine weitere Optimierung des Reglers bezogen auf die Stellgröße nötig. Die Übertragungsfunktion des fertigen Reglers ist in Gl. 5.61 dargestellt.

$$GR_1(s) = 200 \cdot \frac{1}{1-\frac{s}{1}} \cdot \frac{\frac{s}{2}+1}{\frac{s}{200}+1} \cdot \left(1+\frac{12}{s}\right) \cdot \left(1+\frac{12}{s}\right) \cdot \frac{\frac{s}{2}+1}{\frac{s}{200}+1} \qquad \text{(Gl. 5.61)}$$

Die auftretende Gyrodrift stellt eine linear steigende Störung auf die Strecke dar. Der Regler als doppelter Integrator kombiniert mit der Strecke, welche einen einfachen Differenziator darstellt, ergibt zusammen ein einfach integrierendes Verhalten. Durch die wachsende Störung ergibt sich dadurch eine konstante Abweichung am Pendelwinkel. Somit ist die innere Kaskade bezogen auf die Störung durch die Gyrodrift nicht stationär genau. Durch den instabilen Pol im Regler ergibt jede Störung eine, ins Unendliche abdriftende, Stellgröße. Die Auslegung des inneren Reglers bietet dadurch keine Möglichkeit einer sinnvollen Betrachtung der Störungen und deren Auswirkungen auf die Stellgröße. Der äußere Regelkreis berechnet aus dem Motorwinkel den Pendelwinkel für die innere Kaskade. Somit ist ein direkter Zusammenhang, zwischen Stellgröße des inneren Regelkreises und Störung auf die innere Strecke gegeben. Aufgrund dessen ist die Störkompensation anhand des äußeren Reglers durchzuführen.

5.3 Auslegung des äußeren Reglers

Die Auslegung der zweiten Kaskade erfolgt analog zur ersten Kaskade mit dem Ansatz der Schleifenoptimierung. Die Vorbelegung der Übertragungsfunktionen erfolgt nach folgendem Schema:

$$GRC = GFC = GVC = GCC = GDC = 1 \quad \text{(Gl. 5.62)}$$

$$GMC = GSC = G_{Sys2}(s) \quad \text{(Gl. 5.63)}$$

Für die Auslegung des zweiten Reglers der Kaskadenregelung muss, neben der Vordefinition der Übertragungsfunktionen, nur noch „second cascade“ im Popup für die Reglerstruktur gewählt werden. Auf die genaue Beschreibung der Schritte in pzMove wird auch hier verzichtet. Bei der ersten Betrachtung der Nyquist-Kurve der offenen zweiten Kaskade aus Abb. 5.16 fällt auf, dass die Kurve bereits, wie vom Nyquist-Kriterium gefordert, den Punkt P(-75, 0) um 180 Grad umschlingt.

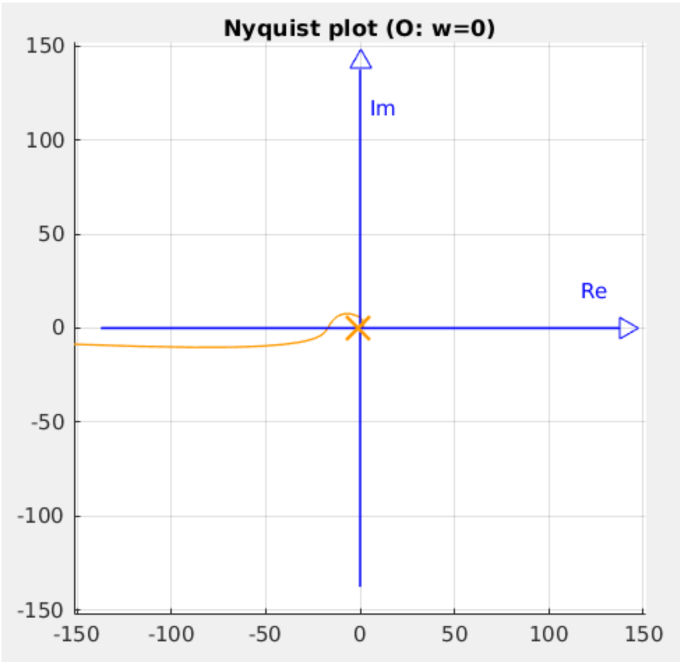


Abb. 5.16: Nyquist-Diagramm der zweiten, offenen Kaskade ohne Regler

Da die Strecke keinen Pol mit positivem Realteil hat, kann auf einen zusätzlichen instabilen Pol verzichtet werden. Auch die Verwendung eines Lead- oder Lag-Glieds ist nicht nötig. Für die Erfüllung des Nyquist-Kriteriums ist lediglich ein P-Glied mit einer Verstärkung von 1/75 = 0,0133 nötig. Das Nyquist-Diagramm mit dem eingefügten P-Glied ist in Abb. 5.17 dargestellt.

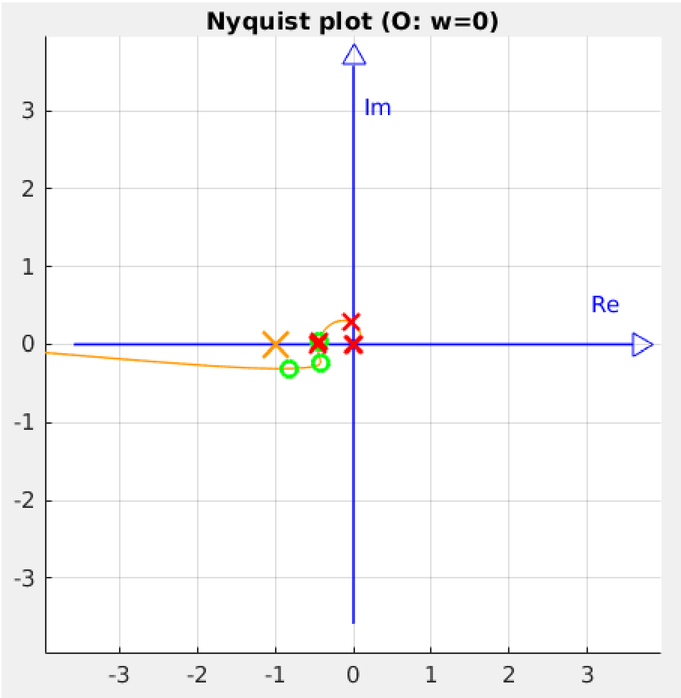


Abb. 5.17: Nyquist-Diagramm der offenen, äußeren Kaskade mit zusätzlichem P-Glied

Die Start-Phase liegt bei - 180 Grad, da die Strecke einen doppelten Integrator darstellt und die innere Kaskade keine Nullstellen im Ursprung hat. Deshalb ist bereits stationär genaues Verhalten gegeben, wodurch keine zusätzlichen PI-Glieder nötig sind.

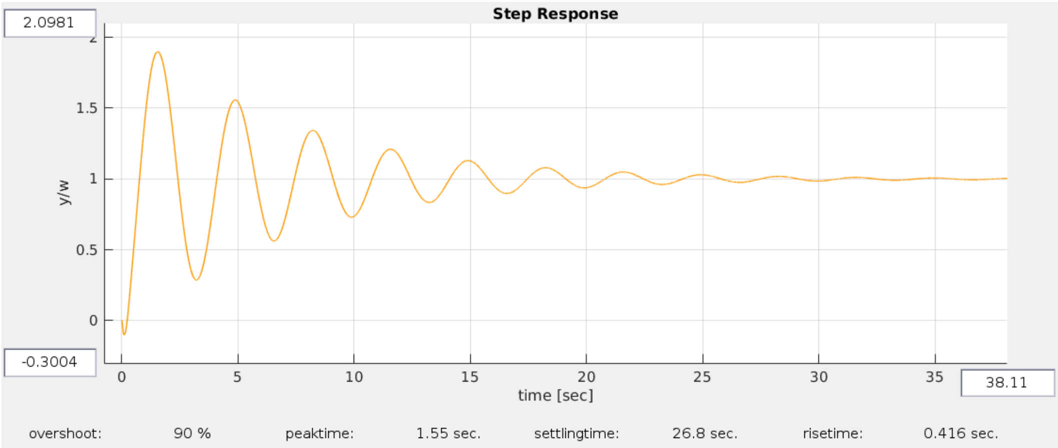


Abb. 5.18: Sprungantwort des geschlossenen Regelkreises mit stabilem und stationär genauem Regler

Die Sprungantwort des geschlossenen Regelkreises aus Abb. 5.18 zeigt zwar, dass der Regler stabil und stationär genau ist, jedoch ist die Regelgüte sehr gering. Ziel des ersten Optimierungsschrittes ist wieder die Vergrößerung des Abstandes zwischen Nyquist-Kurve und kritischem Punkt. Da bis jetzt kein Lead-Glied vorhanden ist, entfällt die Möglichkeit, die obere Grenzfrequenz des Gliedes zu erhöhen. Anstatt dessen ist ein Lead-Glied einzufügen. Dadurch wird der Kurvenbereich unter dem kritischen Punkt gegen den Uhrzeigersinn um den Ursprung gedreht und so der Abstand erhöht. Die Werte der Grenzfrequenzen sind so zu legen, dass die Schlaufe, die die Kurve im Endbereich um den Ursprung bildet, nicht zu groß ist und trotzdem eine ausreichend große Drehung erfolgt. Ein gutes Ergebnis kann mit $\omega_1 = 0,1$ rad/s und $\omega_2 = 3$ rad/s erzielt werden. Anschließend ist der Verstärkungsfaktor des P-Glieds auf $k = 0,001$ zu senken, damit die Schlaufe, wie in Abb. 5.19 gezeigt, wieder um den kritischen Punkt gelegt und so Stabilität gegeben ist. Vergleicht man die Sprungantwort aus Abb. 5.20 mit optimiertem Regler mit der Sprungantwort aus Abb. 5.18 ohne Optimierung, so ist eine deutliche Verbesserung zu erkennen. Die Überschwingweite ist von 90 % auf 8,6 % reduziert und das Abklingverhalten ist erheblich besser. Aus Gl. 5.56 ist zu erkennen, dass hier ein nichtminimalphasiges System vorliegt, weshalb in der Sprungantwort ein negativer Ausschlag vorhanden ist. [8, S. 321]

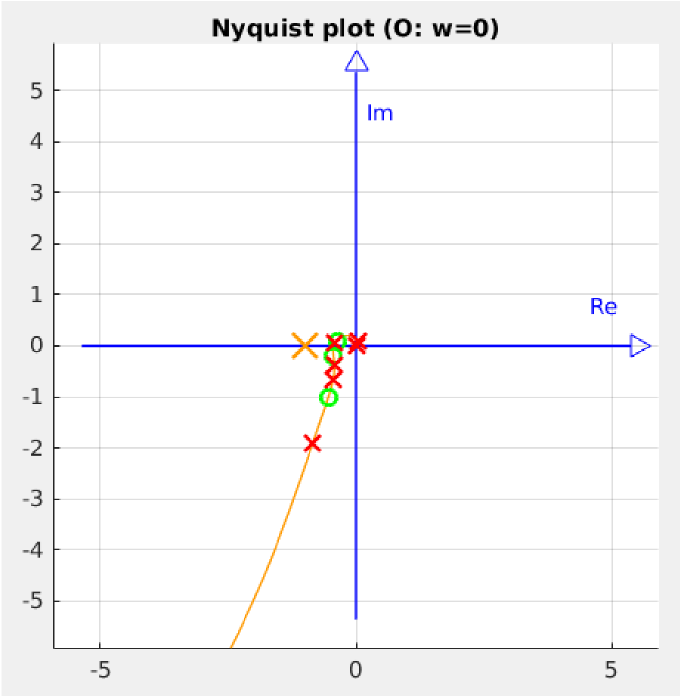


Abb. 5.19: Nyquist-Diagramm der offenen, äußeren Kaskade mit optimiertem Regler

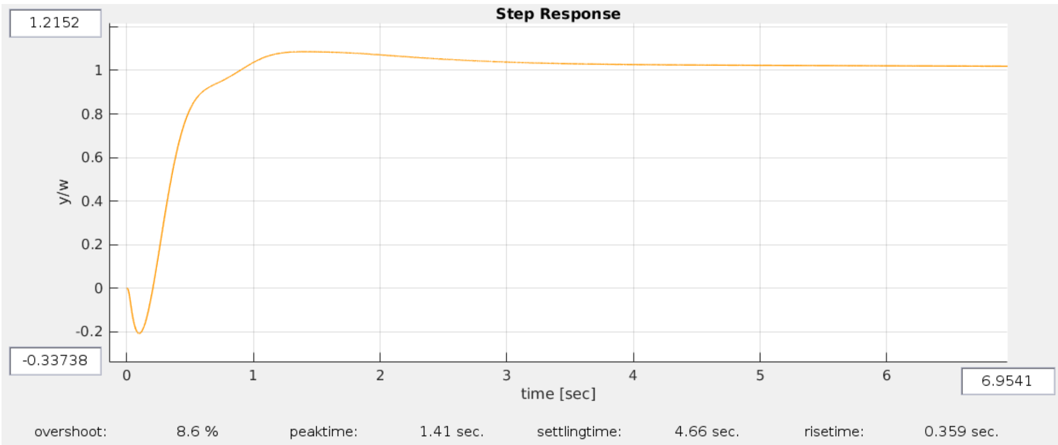


Abb. 5.20: Sprungantwort des geschlossenen Regelkreises mit optimiertem Regler

Die Drift des Gyrosensors stellt eine annähernd linear zunehmende Störung auf die Strecke der inneren Kaskade dar. Zur Simulation im pzMove ist dazu die Störung d mit „1/sˆ2“ zu aktivieren. Der sich ergebende Motorwinkel bei geschlossener, äußeren Kaskade ist in Abb. 5.21 gezeigt.

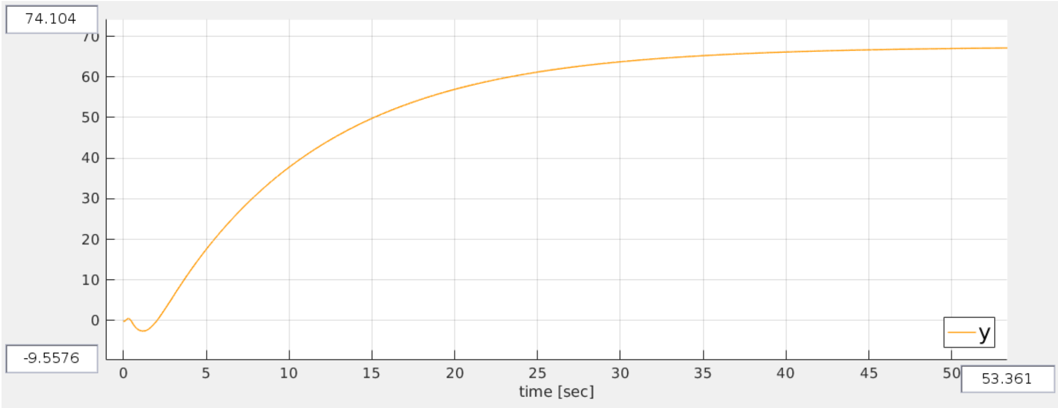


Abb. 5.21: Abweichung des Motorwinkels der geschlossenen, äußeren Kaskade bei aktivierter Gyrodrift

Wie oben erwähnt, erzeugt die Gyrodrift in der ersten Kaskade eine konstante Abweichung zwischen Führungsgröße und Ausgangsgröße. Vernachlässigt man das Verhalten der inneren Kaskade und betrachtet nur diese konstante Abweichung, kann die Gyrodrift als konstante Störung auf das Stellsignal des äußeren Reglers betrachtet werden. Diese wird über die Strecke, die einen doppelten Integrator darstellt, so lange aufintegriert, bis der proportionale Regler der äußeren Kaskade diese konstante Störung auf dem Stellsignal kompensiert. Daraus resultiert die konstante Abweichung des Motorwinkels aufgrund der Gyrodrift. Als Lösung ist deshalb ein integrierender Anteil dem äußeren Regler hinzuzufügen. Wählt man den Verstärkungsfaktor des PI-Glieds so groß wie möglich, ohne dass man dabei die Stabilität gefährdet, klingt die aus der Gyrodrift hervorgerufene Abweichung trotzdem nur sehr langsam ab. Als weitere Möglichkeit zur Verbesserung der Störkompensation kann der Grad des integrierenden Verhaltens des Reglers mit weiteren PI-Gliedern erhöht werden. Ein ausreichend gutes Ergebnis ist zu erzielen, wenn drei PI-Glieder mit $k_i = 0,35$ Verwendung finden. In Abb. 5.22 ist eine Gyrodrift von 1 rad/s simuliert. Geht man von einer maximalen Drift von 0,03 rad/s aus, ergibt dies eine maximale Abweichung von 0,18 rad. Das Führungsverhalten des geschlossenen Regelkreises ist dadurch etwas schlechter. In der Simulation in Abb. 5.23 zeigt sich ein gesteigertes Überschwingen von 51 %.

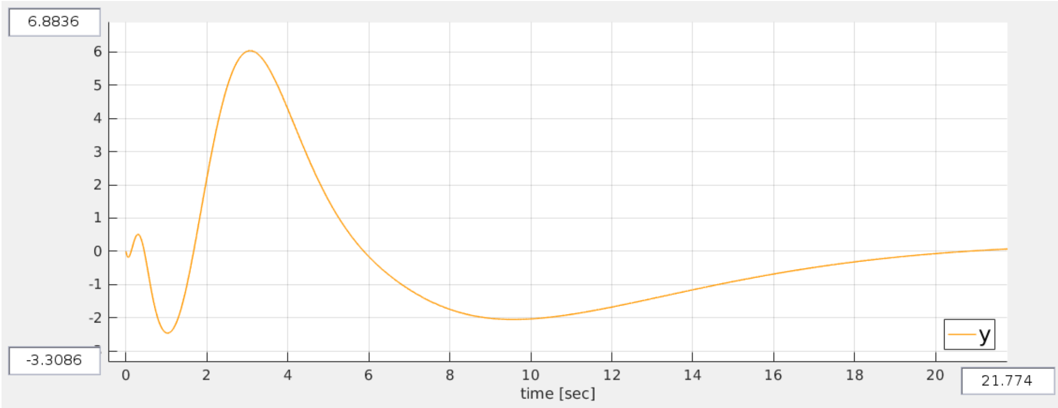


Abb. 5.22: Abweichung des Motorwinkels der geschlossenen, äußeren Kaskade bei aktivierter Gyrodrift mit zusätzlichen PI-Gliedern

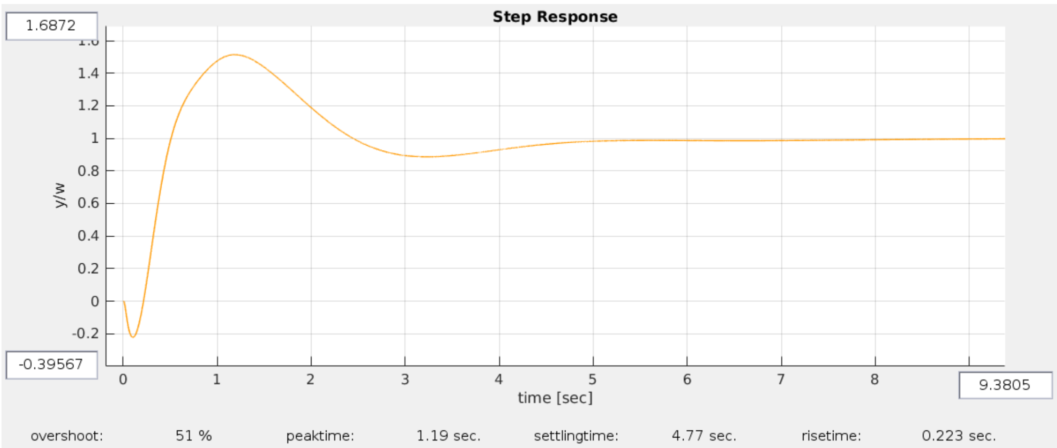


Abb. 5.23: Sprungantwort des geschlossenen Regelkreises mit zusätzlichen PI-Gliedern

Für den zweiten Optimierungsschritt ist wieder ein entsprechendes Szenario festzulegen, das zur Bestimmung der maximalen Stellwerte dient. Es ist nicht sinnvoll den Motorwinkel vorzugeben, da dieser bei einer sehr großen Fahrstrecke einen beliebig großen Sprung darstellen kann. Anstelle des Motorwinkels ist deshalb die Motordrehgeschwindigkeit mit einem Integrator als Vorfilter zu wählen. Bei diesem Beispiel ist von einer sprunghaften Erhöhung der Drehgeschwindigkeit der Motoren auszugehen. Bei der Simulation in pzMove ist zu erkennen, dass bei einer Erhöhung der geforderten Winkelgeschwindigkeit der Motoren von 0 rad/s auf 9 rad/s, das Stellsignal der

Motoren knapp unter 100 % bleibt. In Abb. 5.24 ist die Vorgabe des Motorwinkels mit der Einheit rad in blau dargestellt. In oranger Farbe ist der am mathematischen Modell simulierte Motorwinkel ebenfalls in rad gezeichnet. Bereits hier ist zu erkennen, dass mit den beiden optimierten Reglern in der Kaskadenregelung ein gutes bis sehr gutes Regelergebnis gegeben ist. Mit roter gestrichelter Linie ist das Stellsignal der Motoren in % gekennzeichnet.

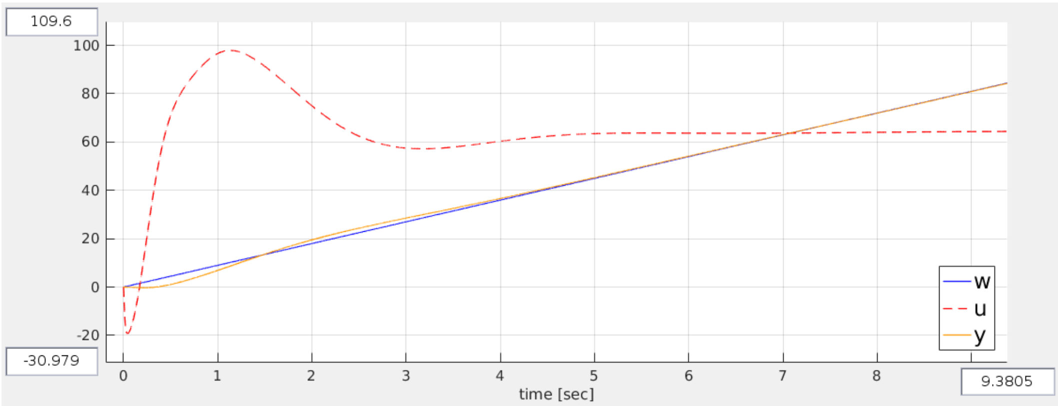


Abb. 5.24: Simulation der sprunghaften Geschwindigkeitszunahme aus dem Stillstand bei geschlossenem Regelsystem

Zu Beginn drehen beide Motoren kurz nach hinten, sodass sich der Roboter nach vorne lehnt. Anschließend drehen die Motoren fast mit voller Drehgeschwindigkeit nach vorne, um den Roboter kurz etwas nach hinten zu lehnen und so die Beschleunigung zu beenden. In voller Fahrt mit 9 rad/s werden die Motoren mit etwa 65 % angesteuert. Betrachtet man die identifizierte Übertragungsfunktion aus Gl. 5.52 und den maximalen Ansteuergrad, ergibt sich eine maximale Drehrate von 13,94 rad/s. Das heißt, dem Roboter kann aus dem Stillstand sprunghaft 64,6 % der maximalen Motorgeschwindigkeit vorgegeben werden, ohne dass dieser dabei die maximalen Stellgrenzen der Motoren überschreitet. Umgerechnet bedeutet eine Drehrate von 9 rad/s bei einem Radius der Räder von 0,028 m eine Geschwindigkeit von 25,2 cm/s. Wird die Geschwindigkeit mit einem zusätzlichen Vorfilter mit zum Beispiel PT2-Verhalten gefiltert, ist eine noch größere Geschwindigkeit möglich. Dabei ist zu beachten, dass durch den Vorfilter die Abweichung zwischen eigentlich vorgegebenem und tatsächlichem Motorwinkel steigt. Durch die 64,6 % der maximalen Motoransteuerung bleiben zudem 35,4 % für eventuell auftretende Störungen übrig. Durch einen zusätzlichen Vorfilter ist eine Erhöhung der maximalen Fahrgeschwindigkeit möglich, jedoch ist damit eine Einschränkung des verbleibenden Reaktionsspektrums der Stellglieder und somit eine größere Störanfälligkeit verbunden. In diesem Beispiel ist der Regler deshalb nicht weiter zu verändern. Die Übertragungsfunktion des optimierten Reglers der zweiten Kaskade ist in Gl. 5.64 gezeigt.

$$GR_2(s) = 0,001 \cdot \left(1 + \frac{0,35}{s}\right) \cdot \left(1 + \frac{0,35}{s}\right) \cdot \left(1 + \frac{0,35}{s}\right) \cdot \frac{\frac{s}{0,1} + 1}{\frac{s}{3} + 1} \qquad \text{(Gl. 5.64)}$$

5.4 Anwendung der Regelung am realen Segway®

Die reale Umsetztung der Regelung erfolgt in MATLAB® Simulink, da dafür ein Support-Package für den EV3-Brick vorhanden ist. Dazu sind die Übertragungsfunktion der Regler aus pzMove zu exportieren und in MATLAB® einzulesen. In Abb. 5.25 ist das entsprechende Modell der Implementation gezeigt. Damit ein Vergleich zwischen realem LEGO® MINDSTORMS® EV3-Roboter und mathematischem Modell möglich ist, wird das Modell zusätzlich mit gleicher Motorwinkelvorgabe und dem gleichen Regler parallel simuliert. Das MATLAB® Simulink-Modell zur Simulation des mathematischen Modells ist in Abb. 5.26 dargestellt. Bei der Simulation wird zu Beginn fünf Sekunden das Signal des Gyrosensors gemittelt, sodass die statische Gyrodrift kompensiert werden kann. Anschließend ist für 15 Sekunden die Motorwinkelvorgabe 0 rad. Weitere fünf Sekunden ist die Vorgabegeschwindigkeit 9 rad/s, wobei der LEGO® MINDSTORMS® EV3-Roboter 45 rad und somit etwa 1,26 m nach vorne fährt. Am Ende der Messung soll der Segway® wieder auf der Stelle stehen bleiben.

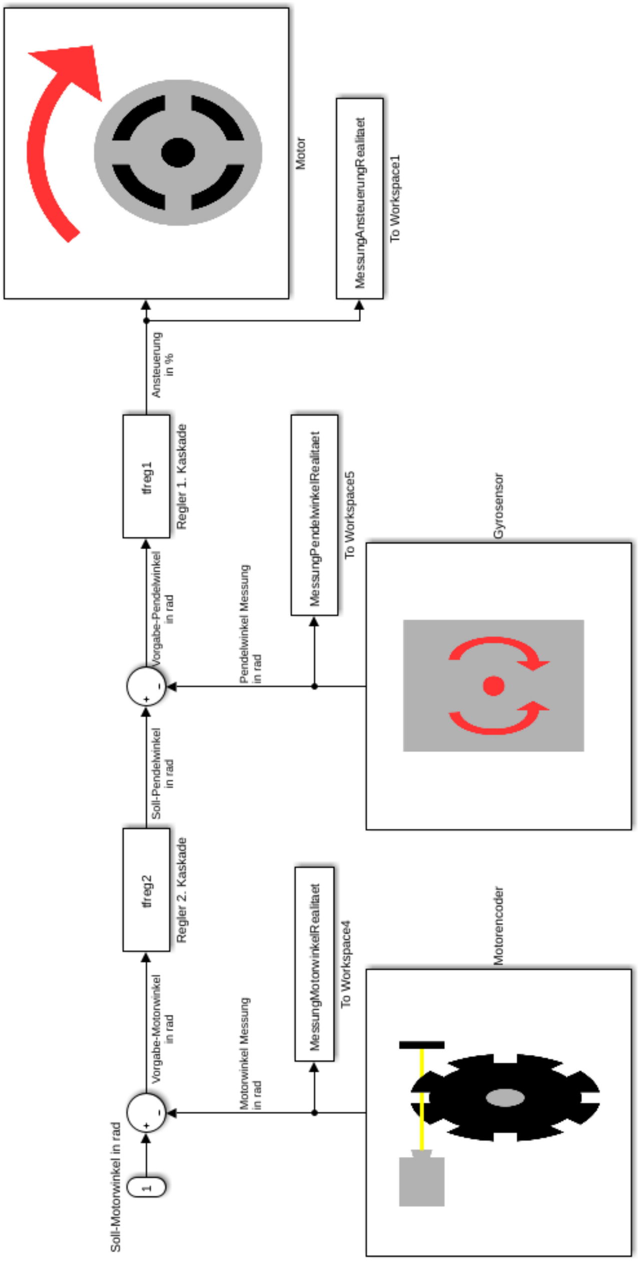


Abb. 5.25: MATLAB® Simulink-Modell zur realen Implementation der Regelung am EV3-Brick

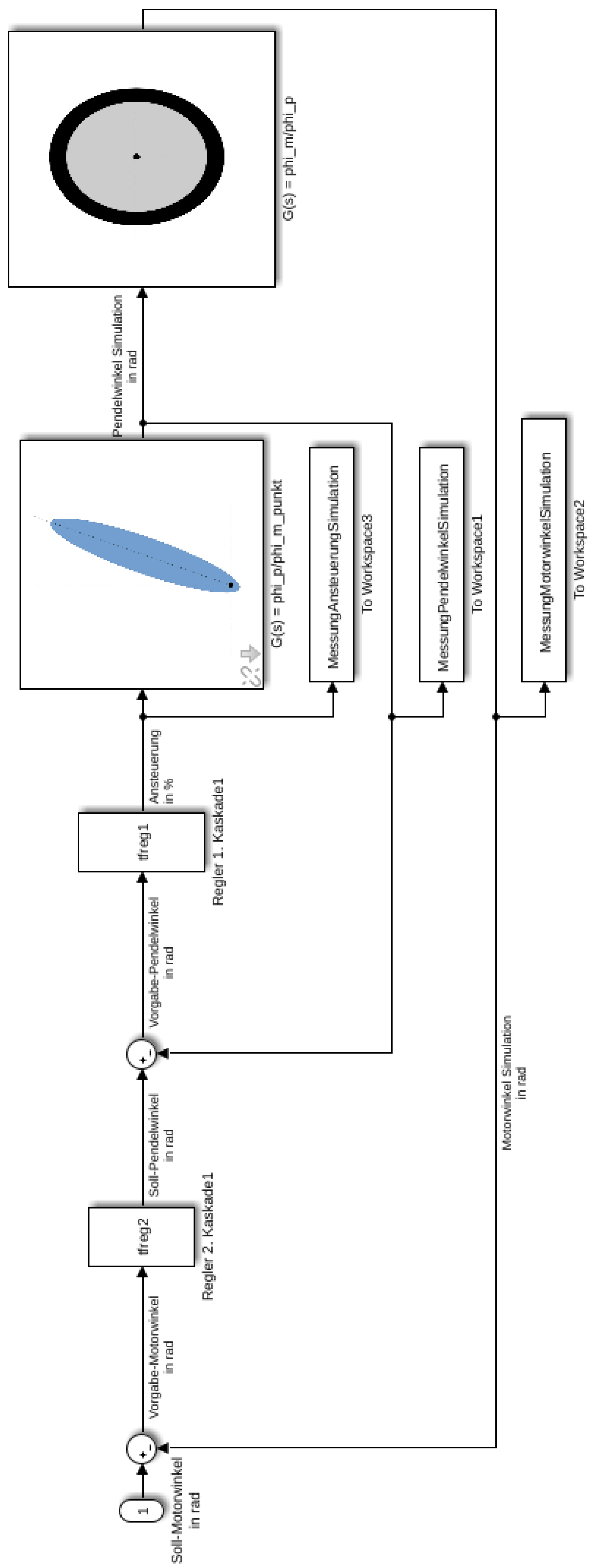


Abb. 5.26: MATLAB® Simulink-Modell zur Simulation des mathematischen Modells

5.4.1 Vergleich von Modell und Realität anhand des Motorwinkels

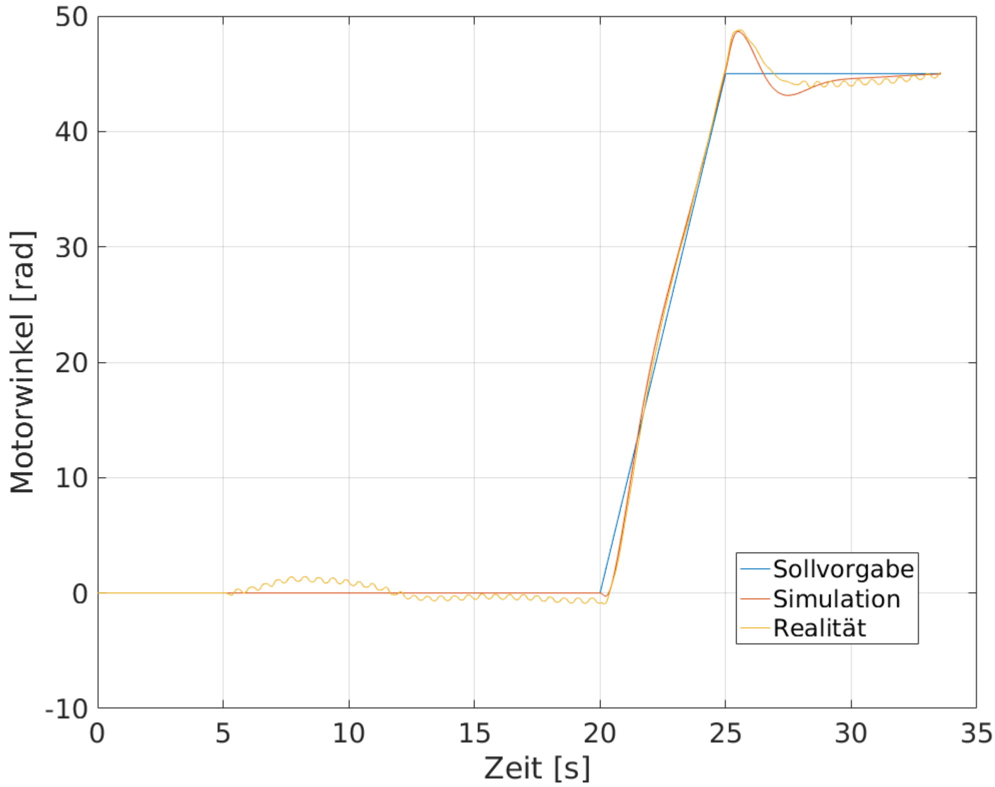


Abb. 5.27: Motorwinkel der Vergleichsmessung

Zu Beginn der eigentlichen Messung nach fünf Sekunden fährt der Segway® aufgrund der Gyrodrift etwas nach vorne. Dieses Verhalten ist analog zur Simulation in Abb. 5.22, jedoch viel geringer, da die Gyrodrift bei dieser Messung kleiner ist. Die Simulation des mathematischen Modells zeigt hier keinen Ausschlag. Die Drift des Sensors ist von Messung zu Messung unterschiedlich, weshalb keine sinnvolle Integration dieser Störung in die Simulation möglich ist. Die Störkompensation der Drift zeigt, dass nach der ersten Auslenkung die Abweichungen vernachlässigbar und somit die fehlende Berücksichtigung in der Simulation tragbar ist. Beim Vorfahren mit 9 rad/s zeigt sich fast kein Unterschied zwischen Simulation und Messung. Das mathematische Modell gibt somit sehr gut den realen Segway® wieder. Beim Bremsen bei t = 25 Sekunden zeigen sich hingegen kleine Differenzen. Bei genauerer Betrachtung der Herleitung des mathematischen Modells fällt auf, dass diese Abweichung zum Beispiel mit der vernachlässigten Rollreibung zwischen Rad und Boden begründet werden kann. Diese Reibung wirkt dämpfend auf das System und vermindert so das Überschwingen des realen LEGO® MINDSTORMS® EV3-Roboters. Auffällig ist auch, dass beim Stillstand des Roboters kleinere, hochfrequente Auslenkungen eintreten. Dies ist dem Getriebespiel geschuldet. Fährt der Segway® nach vorne, werden die Störungen durch Korrektur der Vorfahrgeschwindigkeit ausgeglichen, weshalb dort keine hochfrequenten Schwingungen auftreten. Erst wenn der Roboter im Stillstand ist und dort die Kompensation der Störungen Drehrichtungsänderungen bedürfen, kommt das Getriebespiel zum tragen.

5.4.2 Vergleich von Modell und Realität anhand des Pendelwinkels

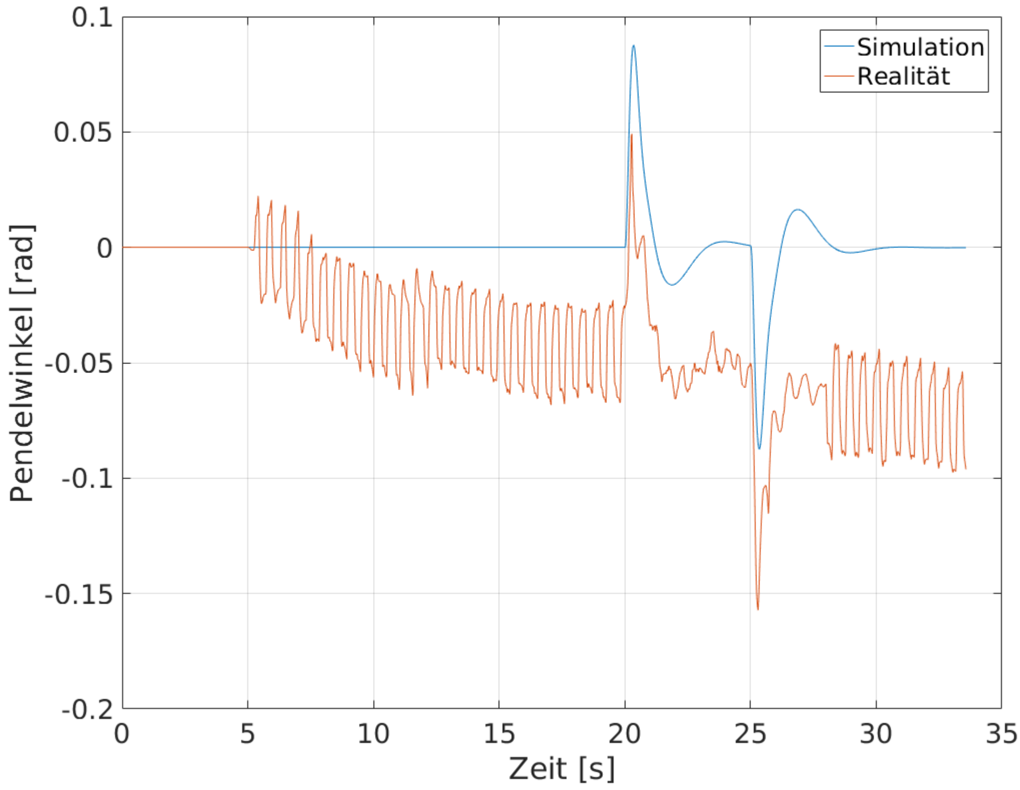


Abb. 5.28: Pendelwinkel der Vergleichsmessung

Beim Vergleich des Pendelwinkels ist die Gyrodrift deutlich erkennbar. Ab dem Messstart nach fünf Sekunden weicht der gemessene Pendelwinkel fast linear immer weiter vom simulierten Modell ab. Vergleicht man das Auftreten der hochfrequenten Störungen mit der Gyrodrift und somit mit der Steigung des gemessenen Pendelwinkels, fällt auf, dass dort ein direkter Zusammenhang besteht. Im Bereich der hochfrequenten Auslenkungen ist die Gyrodrift größer als in den anderen Bereichen. Durch die schnelleren Änderungen der Drehbeschleunigung nimmt bei gleichbleibender Abtastrate die Genauigkeit, mit der das interpolierte Signal auf das tatsächliche Signal passt, erheblich ab. Dadurch steigt mit zunehmender Frequenz der Pendelbewegungen die Abweichung zwischen tatsächlichem und integriertem Signal. Vernachlässigt man die Gyrodrift, ist sehr deutlich erkennbar, dass das mathematische Modell sehr gut zur Realität passt.

5.4.3 Vergleich von Modell und Realität anhand des Stellsignals

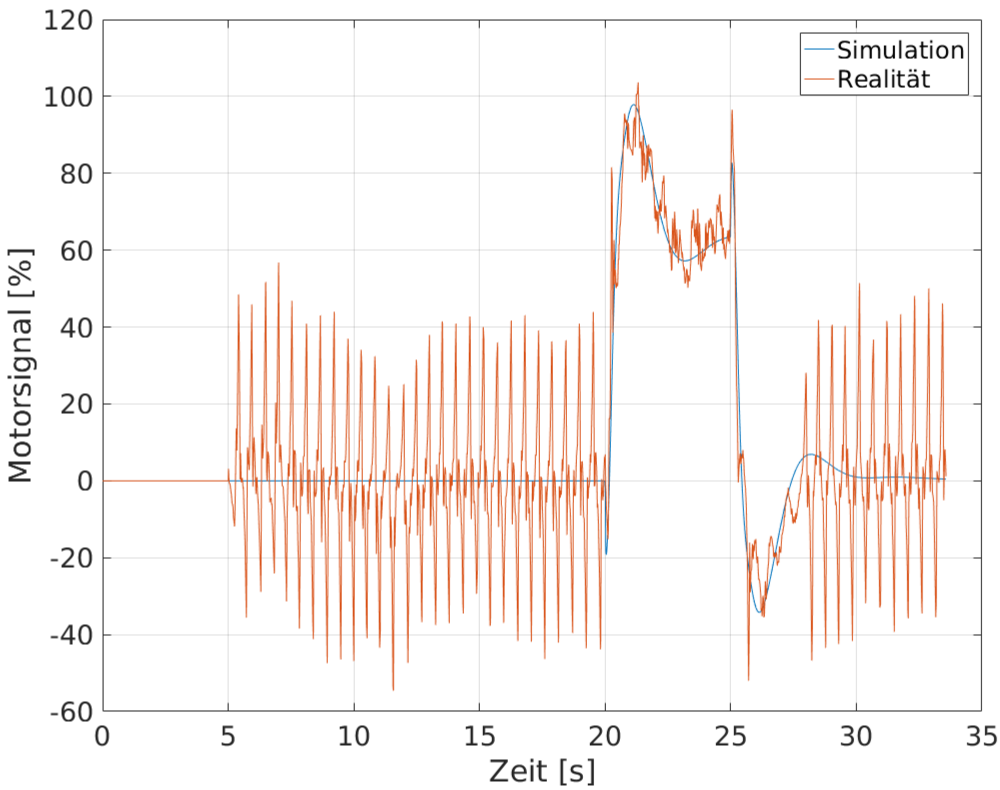


Abb. 5.29: Ansteuergrad der Motoren während der Vergleichsmessung

Der Vergleich des Ansteuergrades zeigt erneut deutlich die Auswirkungen des Getriebespiels. Das Stellsignal hat durch die Totzeit bei der Drehrichtungsänderung impulsartige Ausschläge mit bis zu 50 % Höhe. Vor allem beim Vorfahren ist die Genauigkeit zwischen Modell und Realität erkennbar. Die Kurven von Simulation und Messung passen sehr gut aufeinander. Nur beim Bremsen ist erneut die fehlende Dämpfung im mathematischen Modell erkennbar. Zusammengefasst ist durch das mathematische Modell eine sehr gute Abbildung des realen Segways® gegeben.

5.4.4 Überprüfung der Stoßannahme für die innere Kaskade

Der Stoß, der bei der Auslegung des inneren Reglers Anwendung findet, wird nachfolgend auf dessen Bezug zur Realität überprüfen. Dieser dient bei der Reglerauslegung als Störung, anhand dessen die maximalen Stellwerte ermittelt werden. Da der Stoß am realen LEGO® MINDSTORMS® EV3-Roboter nicht direkt gemessen werden kann, wird auf die genaue Bestimmung der Stoßgrößen verzichtet. Anhand einer einfachen Messung des Pendelwinkels, des Motorwinkels und des Stellsignals kann die Überprüfung der Stellsignale am stabilen System erfolgen. Der Einfluss des Getriebespiels führt zu sehr hohen und impulsartigen Ausschlägen im Stellsignal, wodurch dieses in der Messung möglichst auszuschließen ist. Das erfolgt durch Vorgabe einer geringen Vorwärtsgeschwindigkeit. Zur Messung kommt das MATLAB® Simulink-Modell aus Abb. 5.25 zum Einsatz. Parallel dazu sind mit dem Modell aus Abb. 5.26 die Sollwerte zu bestimmen.

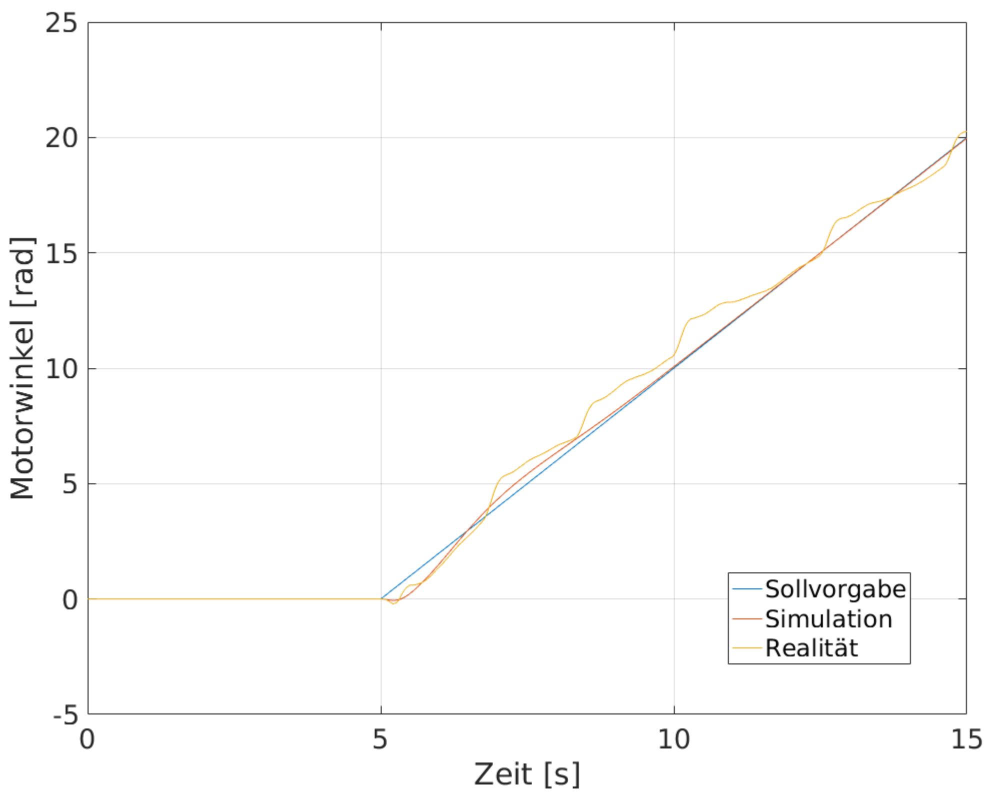


Abb. 5.30: Motorwinkel der Störmessung

Die Motorgeschwindigkeit wird ab einer Initialisierungszeit von fünf Sekunden mit 2 rad/s vorgegeben. Dadurch fährt der LEGO® MINDSTORMS® EV3-Roboter nach vorne und ist in der Lage, geringfügige Störungen durch Geschwindigkeitsanpassungen ohne Drehrichtungswechsel auszugleichen. Die, mit dem mathematischen Modell, berechnete Sollposition ist in 5.30 rot dargestellt. Die Störungen sind manuell als Stöße mit einem Finger am oberen Ende des Roboters einzutragen. Die Größe der Störungen ist hier nur subjektiv festzulegen. Die in dieser Messung aufgebrachten Störungen sind mit mittlerer Kraft ausgeführte Fingerstupser, die für den Segway® und dessen Stellglieder als groß zu betrachten sind. In 5.31 ist die Messung des Pendelwinkels zu sehen. Vernachlässigt man die Gyrodrift und zieht den durch die Fahrbewegung geforderten Pendelwinkel in blau ab, erhält man Ausschläge mit einer Amplitude von etwa 0,02 rad in positiver Richtung und 0,05 rad in negativer Richtung. Ein Vergleich mit der Simulation aus Abb. 5.15 zeigt, dass die realen, positiven Ausschläge im Vergleich zur Simulation mit 0,03 rad um etwa 33 % geringer ausfallen. Die darauf folgenden Ausschläge in die Gegenrichtung sind bei der Messung viel höher als bei der Simulation. Diese großen Ausschläge verursacht das Getriebespiel. Bewegt man die Räder am LEGO® MINDSTORMS® EV3-Roboter mit der Hand leicht vor und zurück, fällt auf, dass die Reibung im Getriebe im Vergleich zum Schleppmoment der Motoren sehr gering ist. Wenn die Motoren in der Vorwärtsbewegung sind und stark abbremsen, ohne dabei die Drehrichtung zu ändern, sinkt die Geschwindigkeit hauptsächlich aufgrund des Schleppmoments der Motoren, wodurch beim Abbremsen die Räder weiterdrehen, bis das Getriebespiel überwunden ist und erst dann vom Motor gebremst werden. In dieser Messung verursacht das genau diesen großen Ausschlag in negativer Richtung. Durch den Fingerstupser lehnt sich der Segway® nach vorne, woraufhin der Regler über eine positive Ansteuerung den LEGO® MINDSTORMS® EV3-Roboter wieder aufrichtet. Wenn der Regler dann die Vorwärtsfahrt abbremst, drehen die Räder durch das Getriebespiel noch etwas weiter und der Roboter lehnt sich zu weit zurück.

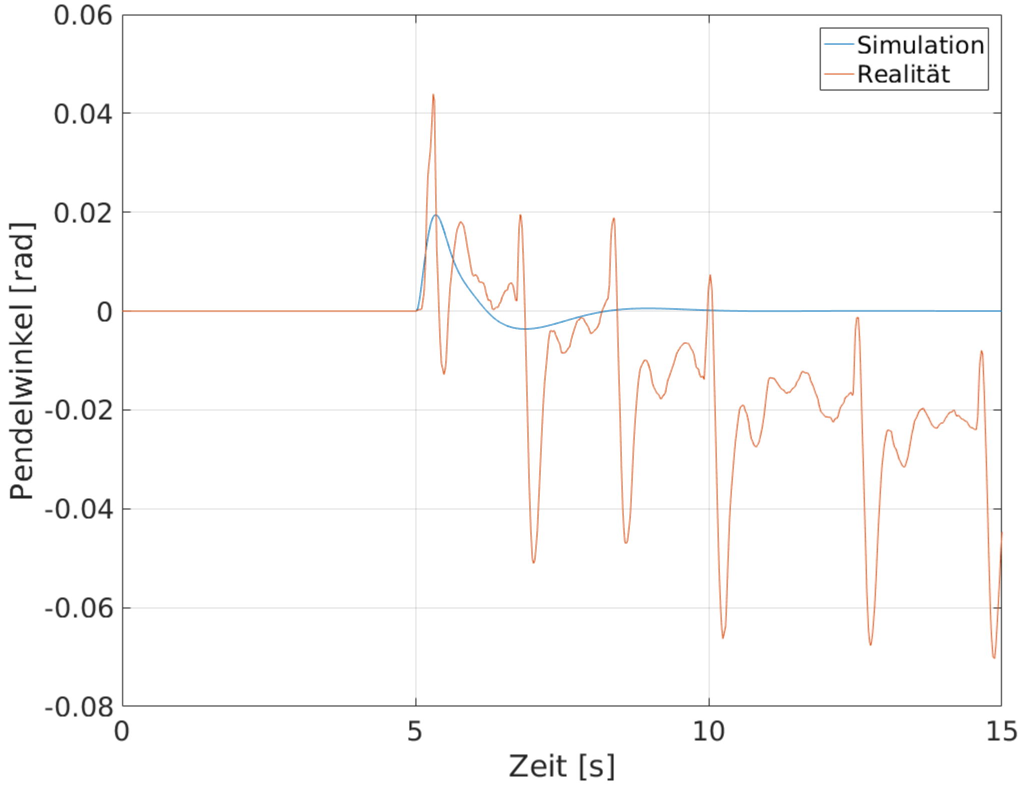


Abb. 5.31: Pendelwinkel der Störmessung

Die maximalen Ausschläge bei den auftretenden Störungen liegen im Bereich zwischen 60 % und 70 %. Bei 100 % Ansteuerung drehen die Motoren mit 13,94 rad/s, was bei 2 rad/s ein Stellsignal von 14,35 % bedeutet. Zieht man diese durch die Vorwärtsfahrt verursachte zusätzliche Ansteuerung ab, ergibt sich ein maximales Stellsignal bei den auftretenden Störungen von 45 % bis 55 %. Vergleicht man die positiven Ausschläge in der Messung mit 0,02 rad mit denen von der Simulation mit 0,03 rad und die Stellsignale von der Messung mit etwa 50 % mit denen von der Simulation mit 80 %, ist zu erkennen, dass die Relation zwischen Messung und Simulation beim Pendelwinkel und beim Stellsignal identisch ist. Daraus lässt sich schließen, dass die bei der Reglerauslegung verwendete Störung um 50 % größer ist, als die in der Realität auftretenden Störungen. Die Annahme ist somit zu groß gewählt und stellt somit eine vertretbare Maximalauslegung dar.

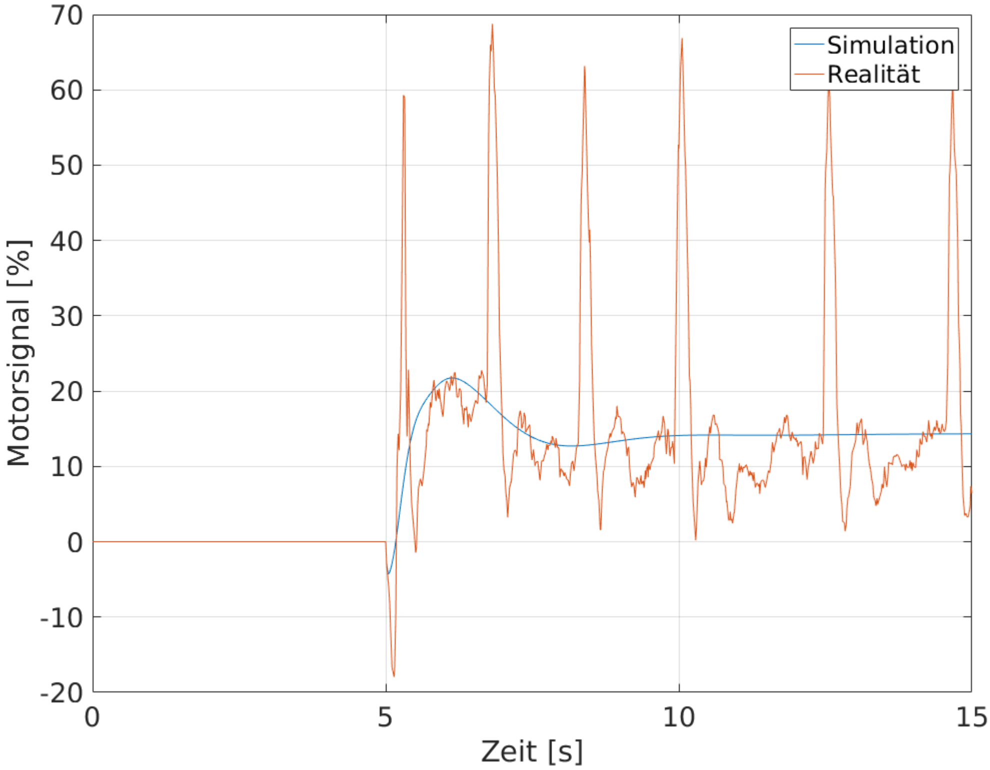


Abb. 5.32: Ansteuergrad der Motoren während der Störmessung

6 Ausblick

Neben der Erweiterung von pzMove ist zukünftig auch vermehrt die Fehlerbehebung zu berücksichtigen. Nur wenn möglichst viele Fehler an das Entwicklerteam gemeldet und dadurch beseitigt werden, ist eine kontinuierliche Verbesserung zu erzielen. Eine ausgereifte und möglichst fehlerfreie Anwendung führt zu zufriedenen Anwendern und somit auch zu einem großen Anwenderkreis.

Sind weitere Funktionen hinzuzufügen, muss die Eignung des bestehenden Programmcodes zur Aufnahme der Erweiterung geprüft werden. Ermöglicht der Code nur eine teilweise oder nicht optimale Integration der Zusatzfunktion, ist der bestehende Code komplett zu überarbeiten und dahingegen zu optimieren.

Die Auslegung einer Kaskadenregelung ist mit pzMove erheblich einfacher möglich, als bei der manuellen Auslegung. Beschränkt ist dies jedoch auf eine zweischleifige Anwendung. Vor allem bei komplexeren Regelaufgaben kann diese Einschränkung Probleme verursachen. Eine Erweiterung auf mehrere Kaskaden, ist mit der bestehenden Programmierung durchaus möglich. Zu beachten ist jedoch, dass die Erweiterung der zur Verfügung stehenden Regelstrukturen zu Programmierarbeit in fast jedem Bereich des AnalyDesign-Teils führen. Des Weiteren kann es dann durchaus sinnvoll sein, die Anzahl der Eingangs- und Ausgangsgrößen zu erhöhen und so weitere Möglichkeiten zur Störeinleitung und zur Stellgrößenbetrachtung zu bieten.

Bei der Integration des Loop-Shaping-Verfahrens ist in dieser Arbeit das Augenmerk auf die manuelle Umsetzung gelegt. Der Anwender muss entscheiden welche Korrekturglieder mit welchen Grenzfrequenzen oder Verstärkungsfaktoren einzusetzen sind. Ein erster Erweiterungsschritt könnte dem Anwender per Knopf-Druck die nötigen Korrekturglieder mit entsprechenden Kennwerten für einen stabilen Regler zur Verfügung stellen. Dazu ist ein ausgefeiltes Konzept zur Ermittlung dieser Glieder nötig. Die Integration in pzMove kann über eine neue Unterfunktion und einen zusätzlichen Button ohne großen Aufwand erfolgen. Durch eine Variation der gesamten Reglervariablen kann anschließend mit den „stabilen“ Gliedern der Regler optimiert werden. Die Optimierung durch Variablenvariation ist auch auf einen vom Anwender erstellten Regler anwendbar.

Die High-End-Version stellt eine Optimierungsfunktion dar, welche nicht nur durch Variablenvariation, sondern auch durch Korrekturgliederkombination die Regelgüte verbessert. Dabei sind grundsätzlich zwei Möglichkeiten denkbar. Bei der ersten Umsetzung kommen die erwähnten Optimierungsgedanken aus Abschnitt 4.2.1 zum Einsatz. So kann zum Beispiel eine bestehende Umschlingung mit einem zusätzlichen Lead-Glied, Erhöhung der Verstärkungsfaktoren der PI-Glieder und Anpassung der Verstärkung des P-Glieds optimiert werden. Denkbar ist jedoch auch, dass für jeden Faktor der bestehenden Glieder eine Variationsoptimierung durchgeführt wird und die Verbesserung anhand einer Bewertung bestimmt wird. Dies kann auch für zusätzliche Glieder erfolgen. Dadurch ist die beste Optimierungsmöglichkeit erkennbar. So ist Schritt für Schritt ein optimaler Regler, ausgehend von einem stabilen Regler, erzeugbar. Dabei ist das Vorgehen so zu gestalten, dass auch die Kombination verschiedener Glieder als Optimierungsmöglichkeit erkannt werden. Dies ist aufgrund der vielen Kombinationsmöglichkeiten mit der großen Variationsbreite der Grenzfrequenzen und der Verstärkungsfaktoren nur schwer umsetzbar.

Die beschriebenen Optimierungsansätze sind auch auf eine Stellgrößen- oder Störkompensationsoptimierung anwendbar. In den meisten Fällen ist dabei eine kombinierte Betrachtung von Regelgüte, Stellgröße und Störkompensation nötig, was zu Interessenskonflikten führen kann. Von einfachen Verfahren bis zu hochkomplexen Ansätzen, die die Regelgüte hinsichtlich verschiedener Kriterien erhöhen, sind viele Möglichkeiten gegeben. Zusammengefasst ist im Bereich des Loop-Shapings bei pzMove erhebliches Erweiterungspotential gegeben.

7 Zusammenfassung

Die Überarbeitung der ursprünglichen pzMove-Version bedurfte großen Zeitaufwands. Der Programmcode ist mit den wenigen, einfachen Richtlinien deutlich kürzer und strukturierter. Durch die Strukturierung sollte der Code für die zukünftige Bearbeitung leichter verständlich und so schneller bessere Ergebnisse erzielbar sein. Der Aufbau der Oberfläche nach dem WYSIWYAF-Prinzip vergrößert die Einarbeitungszeit zum Bearbeiten der GUI erheblich, jedoch sind nur so die Darstellungsfehler versionsunabhängig zu beseitigen. Zudem sind zunehmende Bearbeitungseinschränkungen der immer komplexer werdenden GUI im vorher verwendeten GUIDE hinfällig.

Die Auslegung einer Kaskadenregelung ist mit pzMove genau so einfach möglich wie die Auslegung eines Standardregelkreises. Zuerst kann mit den implementierten Verfahren ein Regler für die innere Kaskade erzeugt werden. Anschließend schaltet man auf die äußere Kaskade um und erzeugt mit dem gleichen Verfahren einen Regler für diese Kaskade. Lediglich die Einschränkung auf zwei Kaskaden ist gegeben.

Der durch die Schleifenoptimierung gefundene Regler in Gl. 4.4 für die Beispielstrecke aus Gl. 4.2 unterscheidet sich vom Regler aus der Kriteriumsherleitung aus Gl. 4.3. Dies stellt allgemein den Nachteil des Loop-Shaping-Verfahrens dar. Es gibt nicht den „Einen Regler“. Durch die Variationsmöglichkeiten bei der Auswahl der Korrekturglieder und bei der Wahl der Grenz- oder Eckfrequenzen und der Verstärkungsfaktoren gibt es unzählige Möglichkeiten, für die gleiche Strecke einen stabilen Regler mit ausreichend guter Regelgüte zu erstellen. Die Güte des Reglers ist dabei durchaus von den Kenntnissen des Anwenders abhängig. Während es für einen Neuanwender relativ einfach ist einen stabilen Regler zu erzeugen, erfordert der Einbezug der Regelgüte etwas Erfahrung. Durch den Lehransatz kann das Loop-

Shaping-Verfahren in seinen Grundzügen den Studenten einfach dargereicht werden, ohne dass umfangreiche Hintergrundkenntnisse erforderlich sind. Für die Anwendung ist hingegen nur die Schleifenoptimierung gedacht. Somit ist auch der Anwendungsansatz in die Lehre mit einzubeziehen, sobald die Zusammenhänge klar sind. Im pzMove ist die Schleifenoptimierung hinterlegt. Mit pzMove, dem Ablauf aus Abschnitt 4.2 und den in Abschnitt 4.3 kurz zusammengefassten Auswirkungen der Korrekturglieder auf die Nyquist-Kurve sind drei Hilfsmittel gegeben, die Erstanwender nach begrenzter Einarbeitungszeit zu einem sinnvollen Ergebnis führen sollen.

Als Beispiel für die Anwendung des Loop-Shaping-Verfahrens und der Kaskadenregelung im Rahmen eines Praktikums für Studenten kann der LEGO® MINDSTORMS® EV3-Roboter dienen. Mit dem Loop-Shaping-Verfahren und dem mathematischen Modell kann dieses instabile System trotz gewisser Hardwareschwächen stabilisiert werden. Durch die weitreichenden Optimierungsmöglichkeiten sind gute Ergebnisse im Regelverhalten zu erzielen, auch wenn vor allem durch die Gyrodrift der äußere Regler aufwändiger zu gestalten ist.

Auch weiterhin bestehen viele Optimierungs- und Erweiterungsmöglichkeiten bei pzMove. Von weiteren Kaskaden, zusätzlichen Reglerauslegungsmethoden oder Optimierungsmethoden für die Loop-Shaping-Methode ist alles Erdenkliche möglich. Dabei ist vor allem auf eine strukturierte Programmierung und Übersichtlichkeit Wert zu legen. Können zusätzliche Funktionen nicht sinnvoll in die bestehende Version integriert werden, sind groß angelegte Programmänderungen sinnvoller als Funktionseinschränkungen, die zu einem unübersichtlichen Porgramm führen.

Tabellenverzeichnis

Abbildungsverzeichnis

Literaturquellen

[1] Fischer, Johannes: *Minimieren der Störeinflüsse eines Segway®-Roboters*, Hochschule Rosenheim, Projektarbeit, 2016

[2] Föllinger, Otto: *Regelungstechnik*. 10. Heidelberg : Hüthig, 2008. – ISBN 978–3–7785–2970–6

[3] Gieck, Reiner: *Technische Formelsammlung*. 32. München : Carl Hanser Verlag, 2011. – ISBN 978–3–446–42113–4

[4] Jautze, Prof. Dr. M.: *Regelungs- und Steuerungstechnik*. Hochschule für angewandte Wissenschaften Landshut : Vorlesungsskriptum, SS 2013

[5] LEGO®: *Intelligenter EV3-Stein*. `https://shop.lego.com/de-DE/Intelligenter-EV3-Stein-45500`. Version: 18:00 21. September 2016

[6] LEGO®: *LEGO® MINDSTORMS® EV3*. `shop.lego.com/de-DE/LEGO-MINDSTORMS-EV3-31313`. Version: 18:00 21. September 2016

[7] Liu, Prof. Dr.-Ing. S.: *Drehzahlregelung eines Gleichstrommotors kleiner Leistung*. `https://www.eit.uni-kl.de/liu/downloads_op/Versuch_GSM.pdf`. Version: 18:00 21. September 2016

[8] Lunze, Jan: *Regelungstechnik 1*. 10. Berlin Heidelberg : Springer Vieweg Verlag, 2014. – ISBN 978–3–642–53908–4

[9] MathWorks®: *Create a Simple App Programmatically*. `http://de.mathworks.com/help/matlab/creating_guis/`

about-the-simple-programmatic-gui-example.html. Version: 18:00 21. September 2016

[10] MathWorks®: *filesep*. http://de.mathworks.com/help/matlab/ref/filesep.html. Version: 18:00 21. September 2016

[11] MathWorks®: *Ways to Build Apps*. http://de.mathworks.com/help/matlab/creating_guis/ways-to-build-matlab-guis.html. Version: 18:00 21. September 2016

[12] Stuart McGarrity, MathWorks®: *Einführung in die Objekt-Orientierte Programmierung mit MATLAB*. http://de.mathworks.com/company/newsletters/articles/introduction-to-object-oriented-programming-in-matlab.html. Version: 18:00 21. September 2016

[13] Trogemann, Georg ; Viehoff, Jochen: *CodeArt*. Wien : Springer Verlag, 2005. – ISBN 3–211–20438–5

Abbildungsquellen

[1] Fischer, Johannes: *Minimieren der Störeinflüsse eines Segway®-Roboters*, Hochschule Rosenheim, Projektarbeit, 2016

[2] LEGO®: *LEGO® MINDSTORMS® EV3 Gyro Boy*. http://static-cdn-2.myrobotcenter.com/media/catalog/product/u/n/unterhaltungsroboter-roboterbausatz-ev3-gyroboy.jpg. Version: 16:00 24. August 2016