



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Masterthesis

Patrick Vogl

Design und Implementierung  
verschiedener Regelungskonzepte  
zur Lageregelung eines Quadropters

*Fakultät Technik und Informatik  
Department Informations- und  
Elektrotechnik*

*Faculty of Engineering and Computer Science  
Department of Information and  
Electrical Engineering*

Patrick Vogl  
Design und Implementierung  
verschiedener Regelungskonzepte  
zur Lageregelung eines Quadropters

Masterthesis, eingereicht im Rahmen der Masterprüfung  
im Masterstudiengang Automatisierung  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Florian Wenck  
Zweitgutachter: Prof. Dr.-Ing. Dipl.-Kfm. Jörg Dahlkemper

Abgegeben am 7. April 2014

**Patrick Vogl**

**Thema der Masterthesis**

Design und Implementierung verschiedener Regelungskonzepte zur Lageregelung eines Quadropters

**Stichworte**

Quadropters, Quadrocopter, Quadkopter, Quadcopter, Cortex-M4, LQR-Regelung, LQ-Regelung, dezentrale Regelung

**Kurzzusammenfassung**

Im Rahmen dieser Arbeit werden verschiedene Methoden zur Lageregelung eines Quadropters entwickelt, implementiert und an einem realen Modell getestet. Zudem werden die implementierten Regelungen miteinander verglichen.

**Patrick Vogl**

**Title of the paper**

Design and Implementation of different control techniques for stabilizing a quadcopter

**Keywords**

quadrocopter, quadcopter, quadrotor, Cortex-M4, LQR-control, LQ-regulator, decentralized controller

**Abstract**

This report describes the development, implementation and test of different approaches to control the position of a quadcopter. Furthermore the different approaches will be compared.

## Danksagung

An dieser Stelle möchte ich zunächst Herrn Prof. Dr.-Ing. Florian Wenck für die Hilfe bei der Wahl des Themas der Masterarbeit danken. Zudem hat er bei Fragen oder Problemen stets Zeit für ein persönliches Gespräch gefunden und sehr viel Interesse am Thema gezeigt.

Mein Dank gilt auch Herrn Prof. Dr.-Ing. Dipl.-Kfm. Jörg Dahlkemper, der sich nach einer kurzen Vorstellung des Themas der Masterarbeit bereit erklärt hat, die Rolle des Zweitprüfers zu übernehmen.

Meinen Eltern möchte ich danken, dass diese auch mein weiterführendes Masterstudium unterstützt haben.

Ein weiterer Dank gilt meinem Nachbarn Armin Welsow, der sich, wie schon bei meiner Bachelorarbeit, bereit erklärt hat, die Arbeit hinsichtlich Rechtschreibung und Zusammenhang zu prüfen.



# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>8</b>
<b>Bilderverzeichnis</b>	<b>9</b>
<b>1 Einleitung</b>	<b>11</b>
1.1 Problemstellung . . . . .	11
1.2 Zielsetzung . . . . .	12
1.3 Gliederung der Arbeit . . . . .	13
<b>2 Grundlagen</b>	<b>14</b>
2.1 Der Quadrocopter . . . . .	14
2.1.1 Einsatzmöglichkeiten . . . . .	15
2.2 Funktionsweise eines Quadrocopters . . . . .	16
2.3 Zustandsraumdarstellung . . . . .	18
2.3.1 Differentialgleichungen eines Systems . . . . .	18
2.3.2 Linearisierung nichtlinearer Systeme . . . . .	19
2.4 Regelungsansätze bei Mehrgrößensystemen . . . . .	21
2.4.1 Vernachlässigung der Querkopplungen . . . . .	21
2.4.2 Entkopplung der Regelstrecke . . . . .	22
2.4.3 Zustandsregelung . . . . .	24
2.5 Digitale Regelsysteme . . . . .	26
<b>3 Systemaufbau und Modellbildung</b>	<b>28</b>
3.1 Auswahl der Hardwarekomponenten . . . . .	28
3.1.1 Quadrocoptergestell . . . . .	28
3.1.2 Motoren und Propeller . . . . .	29
3.1.3 Lagesensor . . . . .	30
3.1.4 Hauptprozessor . . . . .	31
3.1.5 Fernsteuerung . . . . .	33
3.1.6 Akkumulator . . . . .	34
3.2 Aufbau und Test der Motorregler . . . . .	35
3.3 Konstruktion des 3D-Modells des Quadrocopters . . . . .	37
3.4 Aufbau und Inbetriebnahme des Quadrocopters . . . . .	39
3.4.1 Wuchten der Propeller . . . . .	39

---

3.4.2	Aufbau des Quadropters . . . . .	40
3.4.3	Auslesen der Fernsteuersignale . . . . .	40
3.4.4	Auslesen des Lagesensors . . . . .	42
3.4.5	Datenaufzeichnung auf der SD-Karte . . . . .	42
3.4.6	Funktionstest des Gesamtsystems . . . . .	43
3.5	Modellbildung des Quadropters . . . . .	44
3.5.1	Rotordrehzahlen . . . . .	44
3.5.2	Pitch . . . . .	45
3.5.3	Roll . . . . .	46
3.5.4	Yaw . . . . .	46
3.5.5	Wahl der Ausgänge . . . . .	48
3.6	Ermittlung der Modellparameter . . . . .	48
3.6.1	Verhalten des Propellerschubes in Abhängigkeit der Drehzahl . .	48
3.6.2	Verhalten der Rotordrehzahl auf eine Stromänderung . . . . .	51
3.6.3	Ermittlung der Hauptträgheitsmomente aus dem 3D-Modell . .	52
3.6.4	Ermittlung der Motordrehmomente . . . . .	53
3.7	Linearisierung des Modells . . . . .	53
3.7.1	Eigenwerte . . . . .	55
3.7.2	Steuer- und Beobachtbarkeit . . . . .	56
<b>4</b>	<b>LQR-Mehrgrößenregelung</b> . . . . .	<b>57</b>
4.1	Entwurf der LQR-Regelung . . . . .	57
4.2	Störgrößenkompensation . . . . .	61
<b>5</b>	<b>Dezentrale Mehrgrößenregelung</b> . . . . .	<b>64</b>
5.1	Entwurf des Entkopplungsgliedes . . . . .	64
5.2	Regelung durch mehrere Eingrößenregler . . . . .	66
5.2.1	Matlab PID-Regelalgorithmus . . . . .	66
5.2.2	Drehzahlregelstrecke . . . . .	67
5.2.3	Regelung von Pitch und Roll . . . . .	69
5.3	Regelung durch kaskadierte Eingrößenregler . . . . .	72
5.3.1	Regelung von Pitch und Roll . . . . .	73
5.3.2	Regelung von Yaw . . . . .	78
<b>6</b>	<b>Implementierung</b> . . . . .	<b>79</b>
6.1	Programmaufbau . . . . .	79
6.2	LQR-Regelung . . . . .	81
6.3	Dezentrale Regelung . . . . .	81
<b>7</b>	<b>Vergleich der implementierten Regelungskonzepte</b> . . . . .	<b>84</b>
7.1	Aufwand der Reglerauslegungen und Implementierungen . . . . .	84
7.2	Testsznarien und Referenzflüge . . . . .	85

---

7.2.1	Drehzahlannahme der vier Rotordrehzahlen . . . . .	86
7.2.2	Störgrößenkompensation im Schwebeflug . . . . .	87
7.2.3	Untersuchung der vollständigen Entkopplung . . . . .	89
7.2.4	Gezielte Steuereingaben zur Erlangung einer Systeminstabilität	92
7.3	Beurteilung des Flugverhaltens aus Pilotensicht . . . . .	93
7.3.1	Steuerung des Auftriebs . . . . .	93
7.3.2	Ausregelung von Störgrößen . . . . .	93
7.3.3	Steuerung von Pitch und Roll . . . . .	93
7.3.4	Steuerung von Yaw . . . . .	94
7.3.5	Gesamteindruck . . . . .	94
<b>8</b>	<b>Fazit und Ausblick</b>	<b>95</b>
	<b>Literaturverzeichnis</b>	<b>97</b>
	<b>Variablenverzeichnis</b>	<b>101</b>
	<b>Glossar</b>	<b>103</b>
<b>A</b>	<b>Verwendete Messgeräte</b>	<b>105</b>
<b>B</b>	<b>Inhalt der CD</b>	<b>106</b>
	<b>Versicherung über die Selbstständigkeit</b>	<b>107</b>

# Tabellenverzeichnis

3.1	Vergleich der Hauptprozessorboards . . . . .	32
3.2	PPM-Summensignalzeiten von einem Kanal . . . . .	41
3.3	Schubmessung in Abhängigkeit der Drehzahl . . . . .	50
3.4	Zusammenfassung der Eingangs-, Zustands- und Ausgangsgrößen des Zustandsraummodells . . . . .	55

# Bilderverzeichnis

2.1	e-volo VC200 . . . . .	15
2.2	Ansteuerungsarten eines Quadropters: Variante 1 links, Variante 2 rechts . . . . .	16
2.3	Ansteuerungsart Variante 1 des Quadropters . . . . .	17
2.4	Verkopplungen einer Regelstrecke . . . . .	21
2.5	Entkopplung einer Regelstrecke . . . . .	22
2.6	Zustandsrückführung mit Vorfilter . . . . .	24
2.7	Regelfläche der Stellgröße $u(t)$ . . . . .	25
2.8	Digitaler Regelkreis . . . . .	26
3.1	Bestelltes Quadroptergestell . . . . .	29
3.2	Bürstenloser Außenläufermotor EMAX MT2213 . . . . .	30
3.3	Raspberry Pi . . . . .	31
3.4	STM32F4-Discovery . . . . .	32
3.5	Fernsteuerung Graupner MX-16 . . . . .	34
3.6	Fertig aufgebaute Motorregler . . . . .	37
3.7	Vergleich zwischen 3D-Modell (links) und realem Modell (rechts) . . . . .	38
3.8	Propellerwaage mit gewuchtetem Propeller . . . . .	39
3.9	Verlauf des PPM-Summensignals . . . . .	40
3.10	Vergrößerte Ansicht des PPM-Summensignals . . . . .	41
3.11	Erstflug des Quadropters . . . . .	43
3.12	Nachbildung der Drehzahlregelstrecke . . . . .	44
3.13	Nachbildung der Strecke Pitch . . . . .	45
3.14	Ersatzschaltbild eines Gleichstrommotors . . . . .	47
3.15	Nachbildung der Strecke Yaw . . . . .	47
3.16	Wippe für die Schubmessung . . . . .	49
3.17	Approximierte Funktion der Schubkraft in Abhängigkeit von der Drehzahl . . . . .	50
3.18	Verhalten der Drehzahl auf einen Stromsollwertsprung . . . . .	51
3.19	Koordinatensystemausrichtung Quadropter . . . . .	52
4.1	Simulinkmodell Subsystem Discrete State Space . . . . .	58
4.2	Simulinkmodell zur Simulation der LQR-Regelung . . . . .	59
4.3	Simulierte Sprungantwort der Drehzahlen . . . . .	59
4.4	Simulierte Sprungantwort Pitch, Roll, Yaw . . . . .	60

---

4.5	LQR-Regelung Soll-, Istwertvergleich . . . . .	61
4.6	Störgrößenkompensation . . . . .	62
4.7	LQR-Regelung Soll-, Istwertvergleich mit Störgrößenkompensation . . . . .	63
5.1	Dezentrale Regelung der Drehzahlen (Sprungantwort) . . . . .	69
5.2	Wurzelortskurve der diskretisierten Strecke Pitch, geregelt durch einen P-Regler . . . . .	70
5.3	Dezentrale Regelung Pitch (Sprungantwort) . . . . .	72
5.4	Grundaufbau der kaskadierten Regelung . . . . .	73
5.5	Dezentrale Regelung Pitch, innerer Regelkreis (Sprungantwort) . . . . .	74
5.6	Dezentrale Regelung Pitch, innerer Regelkreis, nachjustiert (Sprungantwort) . . . . .	75
5.7	Sprungantwort der Strecke Pitch mit und ohne kaskadierter innerer Regelung . . . . .	76
5.8	Äußere geregelte Strecke Pitch (Sprungantwort) . . . . .	77
5.9	Äußere geregelte Strecke Yaw (Sprungantwort) . . . . .	78
6.1	Schematischer Programmablaufplan . . . . .	80
7.1	Vergleich der Drehzahlregelung zwischen LQR-Regelung und dezentraler Regelung . . . . .	86
7.2	LQR-Regelung mit Störgrößenkompensation im Schwebeflug . . . . .	87
7.3	Dezentrale kaskadierte Regelung im Schwebeflug . . . . .	88
7.4	Flugaufzeichnung der LQR-Regelung mit Störgrößenkompensation . . . . .	90
7.5	Flugaufzeichnung der dezentralen kaskadierten Regelung . . . . .	91
7.6	Erlangung einer Instabilität der LQR-Regelung von Pitch . . . . .	92
7.7	Erlangung einer Instabilität der dezentralen kaskadierten Regelung von Pitch . . . . .	92

# 1 Einleitung

## 1.1 Problemstellung

In meiner Bachelorarbeit wurde bereits eine Motoransteuerung für bürstenlose Modellbaumotoren entwickelt. Diese wurde unter Berücksichtigung eines späteren Einsatzes in Quadro- bzw. Multikoptern konzipiert. Zu erwähnen ist in diesem Fall vor allem die Einsetzbarkeit von verschiedenen Motoren und die I2C-Schnittstelle. So kann der Quadrokoopter mit geeigneten Motoren eingesetzt werden und die I2C-Schnittstelle ermöglicht das Senden von Soll- und Empfangen von Istwerten zur Regelung der Motoren. Jedoch wurde während der Bachelorarbeit kein Test mit anderen Motoren durchgeführt. Ein weiterer Aspekt des Motorreglers ist die implementierte Stromregelung, dessen Einsatz eventuell neue Ergebnisse hinsichtlich der Lageregelung liefern könnte. [Vogl, 2012]

Um die Flugfähigkeit eines Quadrokoopters zu erreichen, muss eine entsprechende Lageregelung zur Stabilisierung des Modells vorhanden sein, da eine direkte Steuerbarkeit durch einen Piloten nicht möglich ist. Dabei stellt sich die Frage, warum das Modell nicht direkt gesteuert werden und in welcher Art und Weise eine Regelung erfolgen kann.

In einer Dissertation von Tim Puls ist mit Verweis auf einen IEEE-Artikel die Rede davon, dass zur Lageregelung eines Quadrokoopters mehrere konventionelle PID-Regler besser geeignet sein sollen als eine Zustandsregelung. Die Zustandsregelung liefert zwar funktionierende Ergebnisse, ist jedoch langsamer und weist weniger Dynamik auf als die PID-Regelung. Durch den Einsatz meiner eigenen Motorregler mit Stromregelung könnte sich in diesem Fall ein anderes Ergebnis zeigen. [Puls, 2011, S. 14] [Bouabdallah u. a., 2004]

## 1.2 Zielsetzung

Im Rahmen dieser Arbeit sollen mindestens zwei verschiedene Methoden zur Lageregelung eines Quadropters entworfen, implementiert und getestet werden. Dazu soll die aus der Bachelorarbeit entwickelte Motoransteuerung eingesetzt werden, die in vierfacher Ausführung vorhanden sein muss. Es muss ein geeignetes Quadroptersmodell inklusive Motoren aufgebaut werden, das als Grundlage für alle weiteren Tests dienen soll. Außerdem müssen alle weiteren Hardwarekomponenten vorhanden sein, um ein vollständig flugfähiges und steuerbares Modell als Testplattform zu erhalten. In der Bachelorarbeit wurde die universelle Einsetzbarkeit mit unterschiedlichen Motoren des Motorreglers noch nicht untersucht. Für den Quadropter sollen allerdings andere Motoren eingesetzt werden, sodass in dieser Arbeit auch die Einsetzbarkeit mit zumindest einem anderen Motor getestet wird. Eventuell auftretende Fehler in der Soft- sowie der Hardware müssen dabei vor der Entwicklung der Lageregelung beseitigt werden.

Zum Vergleich der implementierten Regler wird angestrebt, diese hinsichtlich ihrer Dynamik möglichst gleich auszulegen. Zudem sollen die Daten während des Fluges aufgezeichnet werden, um diese später auswerten zu können. Der Test der Lageregler soll nicht - wie in anderen Dissertationen üblich - an einem Testgestell erfolgen, sondern durch reale Flüge im Freien. Dadurch ist es möglich, die Einsetzbarkeit des Quadropters mit den verschiedenen Regelungsarten bezüglich des realen Einsatzes beurteilen zu können, wodurch auch eventuell nicht berücksichtigte Störgrößen zum Tragen kommen. Zur Vergleichbarkeit müssen entsprechende Referenzflüge bzw. Testszenarien definiert werden, die für alle implementierten Regelungsarten angewandt werden.



## 1.3 Gliederung der Arbeit

Die Arbeit gliedert sich in acht Hauptkapitel. Kapitel eins beinhaltet die Einleitung. In Kapitel zwei werden die theoretischen Grundlagen erläutert, die den Quadrokopter und dessen Funktion umfassen und es wird auf die verschiedenen Arten der Regelung von Mehrgrößensystemen eingegangen. Zudem werden wichtige Grundlagen von digitalen Regelkreisen aufgezeigt, die bei der Entwicklung beachtet werden sollten.

Im Kapitel drei werden Aufbau und Test der gesamten Quadrokopterhardware behandelt. Dazu zählen die Fertigung und Inbetriebnahme der vier Motorregler, die Auswahl der Hardwarekomponenten und der Aufbau des Quadrokopters. Zudem wird ein entsprechendes regelungstechnisches Modell entworfen, dessen Parameter bestimmt und Eigenschaften untersucht werden.

In den Kapiteln vier und fünf werden die unterschiedlichen Arten von Regelungen entworfen und implementiert.

Es folgt in Kapitel sechs ein gemeinsamer Implementierungsteil für die zu implementierenden Regelungen, da sich die Implementierungen nur geringfügig unterscheiden.

Das siebte Kapitel behandelt den Vergleich der in den Kapiteln vier und fünf entwickelten und in Kapitel sechs implementierten Regelungsarten.

Im letzten Kapitel acht wird die Arbeit resümiert und eventuelle Verbesserungen thematisiert.

Eine genaue Beschreibung des Inhalts findet sich am Anfang jedes Kapitels.

## 2 Grundlagen

In diesem Kapitel werden wissenswerte Themen und Entwicklungen sowie die Funktionsweise eines Quadropters abgehandelt. Zudem wird auf die Zustandsraumdarstellung eingegangen, die unterschiedlichen Arten zur Regelung von Mehrgrößensystemen beschrieben und die Diskretisierung von kontinuierlichen Systemen behandelt.

### 2.1 Der Quadroptter

Bei einem Quadroptter handelt es sich um ein Flugobjekt, welches zur Klasse der Hubschrauber gezählt werden kann. Der Quadroptter ist in der Lage, senkrecht starten und landen zu können, sodass keine Landebahn notwendig ist. [[Wikipedia Quadroptter](#)]

Charakteristisch für den Quadroptter sind dessen vier Rotoren, die für den nötigen Auftrieb sorgen. Auch Konfigurationen mit mehr oder weniger Rotoren sind üblich. Es handelt sich dann zum Beispiel um Hexa- oder Oktokopter, die sechs bzw. acht Rotoren besitzen. Als Oberbegriff wird hier von Multikoptern gesprochen. Eine höhere Anzahl an Rotoren wird angewandt, um eine Redundanz bzw. Ausfallsicherheit zu gewährleisten oder die Traglast zu erhöhen. [[mikroptter MK-Funktion](#)] Das grundsätzliche Prinzip der Fortbewegung bleibt jedoch bei allen Multikoptern mit fest montierten Rotoren gleich. Eine Ausnahme bildet der Trikotter, der nur drei Rotoren besitzt und mindestens einen nicht fest montierten Rotor zum Drehmomentausgleich besitzt.

Durch die Eigenschaft an einer bestimmten Position zu schweben sowie sich in alle Richtungen bewegen zu können, sind die Einsatzmöglichkeiten von Multikoptern sehr vielfältig.

### 2.1.1 Einsatzmöglichkeiten

Spätestens seit den Plänen Amazons, mit Quadrokoptern bzw. Multikoptern Pakete auszuliefern, sind diese durch die Nachrichten recht publik geworden. Amazon plant dabei die Auslieferung von Paketen vollständig mit Drohnen zu automatisieren. Probleme ergeben sich derzeit noch bei der Flugaufsichtsbehörde, da es sich um unbemannte Fluggeräte handelt, deren Sicherheit unter Beweis gestellt werden muss. Jedoch bieten sich noch weitaus mehr Einsatzmöglichkeiten durch diese vielseitigen Flugobjekte. [Tagesschau Drohne, 2013]

Unter Modellbauern sind diese Flugobjekte schon längere Zeit bekannt. Seit es erschwingliche Lagesensoren für Privatpersonen zu erwerben gibt, schreitet die Entwicklung stetig voran. Bereits 2004 wurde beispielsweise ein ferngesteuertes Modell von der Firma Silverlit in China produziert. [Silverlit]



**Bild 2.1:** e-volo VC200<sup>1</sup>

Eine äußerst interessante Entwicklung stellt zudem der Einsatz als Transportmittel für Personen dar. Ein erster Testflug wurde durch die e-volo GmbH mit dem VC1 bereits durchgeführt. Die Ergebnisse waren derart zufriedenstellend, dass derzeit ein weiteres Modell gebaut wird und auch dieses schon seine Flugfähigkeit unter Beweis gestellt hat, jedoch zunächst ohne Insassen. Es handelt sich dabei um den VC200, der in Bild 2.1 zu erkennen ist. [e-volo]

Ein weiteres Einsatzgebiet ist beispielsweise auch die Nutzung von Multikoptern mit Kameramontage. Durch den Schwebeflug oder auch langsame Vorwärtsflüge lassen sich mit relativ geringem Kostenaufwand Luftaufnahmen der gewünschten Umgebung produzieren, ohne vergleichbar teure Helikopter- oder Flugzeugflüge mit Kameramann in Anspruch nehmen zu müssen.

---

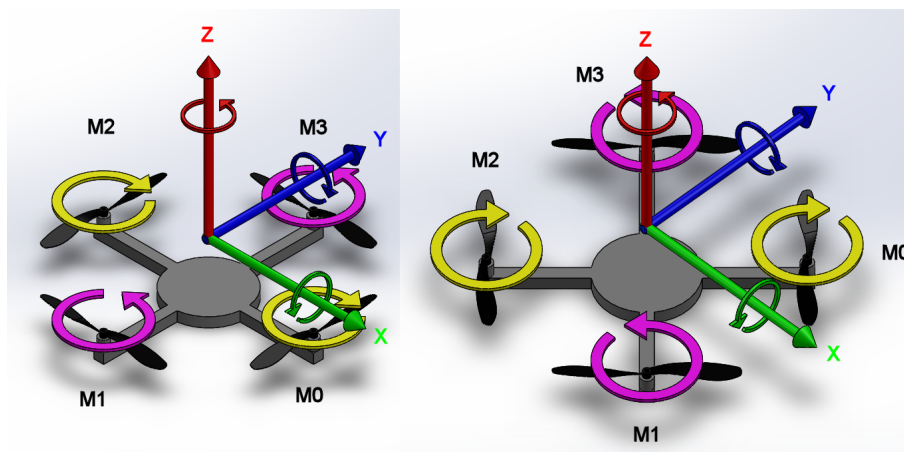
<sup>1</sup>Bildquelle: [http://www.e-volo.com/wp/wp-content/gallery/vc200/VC200\\_Seite\\_T\\_A.jpg](http://www.e-volo.com/wp/wp-content/gallery/vc200/VC200_Seite_T_A.jpg) (Stand: 12.02.2014)

## 2.2 Funktionsweise eines Quadropters

Wie schon im Kapitel 2.1 besprochen, besitzt der Quadropters vier Rotoren, die für den nötigen Auftrieb sorgen.

Um dem Piloten die Erkennung der Ausrichtung des Modells im Raum zu ermöglichen, werden Quadropters oft mit Sichtmarkierungen gekennzeichnet. Dies kann zum Beispiel durch unterschiedliche Farben der Ausleger des Quadropters oder durch montierte Sichtmarkierungen erfolgen. Die Kenntnis der Orientierung im Raum ist äußerst wichtig, da sich zum Beispiel bei der Drehung des Modells um  $180^\circ$  um die Hochachse (Yaw) die Steuerfunktionen für Pitch und Roll „umkehren“. Gerade dieses Verhalten als Pilot zu erlernen erfordert viel Übung. Bei Modellhelikoptern wird in diesem Zusammenhang vom Nasenschweben gesprochen.

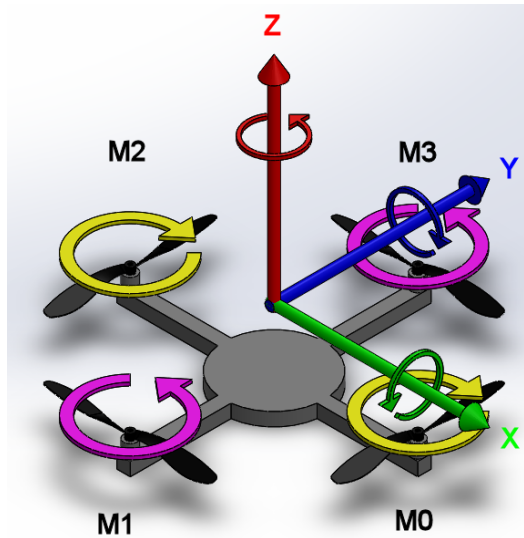
Grundsätzlich gibt es zwei verschiedene Arten einen Quadropters anzusteuern, die sich letztendlich jedoch nur durch die spätere Flugrichtung unterscheiden.



**Bild 2.2:** Ansteuerungsarten eines Quadropters: Variante 1 links, Variante 2 rechts

Bild 2.2 zeigt die zwei unterschiedlichen Ansteuerungsarten bzw. die Lage der unterschiedlich gedrehten Koordinatensysteme. Die Koordinatenachsen sind dabei mit X (grün), Y (blau), Z (rot), die Motoren mit M0 bis M3 gekennzeichnet. Zudem sind die Rotationsrichtungen um die Achsen jeweils in Achsenfarbe eingezeichnet. Die Drehrichtungen der vier Motoren sind mit gelben und violetten Drehpfeilen dargestellt. In diesem Fall wird die erste Ansteuerungsvariante bevorzugt, da zur Änderung der Lage um die X- und Y-Achsen jeweils lediglich zwei Motoren notwendig sind. Zudem kann das Verhalten der Variante zwei später durch eine einfache Drehung der Sollwerte nachgebildet werden. Die zweite Ansteuerungsvariante wird zum Beispiel bevorzugt, wenn eine Kamera oberhalb des Quadropters montiert werden soll. Die Kamera

kann in diesem Fall so montiert werden, dass sie in Flugrichtung keine bzw. eine geringere Sichtbehinderung durch die sich drehenden Propeller erfährt. Es wird jedoch nachfolgend lediglich die erste Variante betrachtet und die Funktion anhand dieses Modells erläutert.



**Bild 2.3:** Ansteuerungsart Variante 1 des Quadrokopters

Bild 2.3 zeigt nochmals eine vergrößerte Ansicht der ersten Ansteuerungsvariante, die im Folgenden näher betrachtet wird.

Entscheidend für das Steigen und Sinken des Quadrokopters, also die Bewegung in Richtung der Z-Achse, sind die Schubkräfte aller vier Antriebe. Soll der Quadrokopter steigen, muss die Schubkraft aller vier Antriebe erhöht werden. Dies wird durch eine Drehzahlerhöhung der Propeller erreicht. Ein Flug in eine beliebige Richtung wird durch eine nicht horizontale Ausrichtung des Quadrokopters zur Erdoberfläche erreicht. In diesem Fall teilen sich die Schubkräfte in Auftrieb und Vortrieb auf.

Gut zu erkennen ist, dass sich jeweils zwei Motoren in die entgegengesetzte Richtung drehen. Dies dient dem Drehmomentausgleich, wodurch ein Quadrokopter im Gegensatz zu einem normalen Hubschrauber mit Heckrotor keinen extra Antrieb zum Drehmomentausgleich benötigt. Eine Drehung um die Z-Achse (Yaw) wird erreicht, indem zwei Motoren mit gleicher Drehrichtung ein höheres Drehmoment erzeugen als die beiden Motoren, die entgegengesetzt drehen. Für die Motordrehmomente sind die eingepprägten Motorströme verantwortlich. Eine Drehung um die Z-Achse in Richtung des eingezeichneten Drehpfeils in Bild 2.3 wird erreicht, indem M0 und M2 einen höheren Strom als M1 und M3 führen.

Für die Änderung der Fluglage um die X-Achse (Roll) und Y-Achse (Pitch) sind die Schubkräfte der einzelnen jeweils gegenüberliegenden Propeller zuständig. Soll

beispielsweise eine Drehung um die Y-Achse in Drehpfeilrichtung erfolgen, erzeugt M0 einen geringeren und M2 einen höheren Schub. In gleicher Weise wird bei der Drehung um die X-Achse verfahren, wobei in diesem Fall M1 und M3 für die Drehung zuständig sind. [Bresciani, 2008, S. 8-11]

## 2.3 Zustandsraumdarstellung

Bei der Zustandsraumdarstellung handelt es sich um eine allgemeingültige Form der mathematischen Beschreibung von SISO- und MIMO-Systemen. Diese Darstellungsart findet oft bei Mehrgrößensystemen Verwendung und eignet sich sehr gut für die Auslegung von Regelungen.

### 2.3.1 Differentialgleichungen eines Systems

Aus der physikalischen oder auch elektrotechnischen Modellbildung ergeben sich in vielen Fällen Differentialgleichungen. Diese können auch nichtlineare Zusammenhänge abbilden.

Als Beispiel sei die Differentialgleichung eines einfachen Pendels in Gleichung 2.1 genannt, dass durch einen Motor mit der Stellgröße  $w$  angeregt werden kann und dessen Auslenkung zum unteren Gleichgewichtspunkt  $\theta$  beträgt. Die Gleichung wurde zur Veranschaulichung vereinfacht.

$$\ddot{\theta}(t) = -\sin(\theta(t)) + w(t) \quad (2.1)$$

Es handelt sich in diesem Fall um ein nichtlineares System zweiter Ordnung.

Dieses System soll in die Zustandsraumdarstellung überführt werden. Dazu wird die Differentialgleichung zweiter Ordnung durch Wahl von zwei Zuständen in zwei Differentialgleichungen erster Ordnung aufgeteilt. Die Reihenfolge der Wahl der Zustände ist beliebig.

$$x_1(t) = \theta(t) \quad (2.2)$$

$$x_2(t) = \dot{\theta}(t) \quad (2.3)$$

Die Differentialgleichungen ergeben sich mit den gewählten Zuständen  $x_1(t)$  und  $x_2(t)$  und der Eingangsgröße  $u_1 = w$  wie folgt:

$$\dot{x}_1(t) = x_2(t) \quad (2.4)$$

$$\dot{x}_2(t) = -\sin(x_1(t)) + u_1 \quad (2.5)$$

Als Ausgang soll der Pendelwinkel betrachtet werden.

$$y(t) = x_1(t) \quad (2.6)$$

### 2.3.2 Linearisierung nichtlinearer Systeme

Die Zustandsraumdarstellung kann nur lineare Zusammenhänge abbilden. Aus diesem Grund müssen nichtlineare Differentialgleichungen linearisiert werden. Diese Linearisierung erfolgt idealerweise im Arbeitspunkt  $(\mathbf{x}_0, \mathbf{u}_0)$ , in dem das System später betrieben werden soll.

Das System liegt in der folgenden allgemeinen Form vor:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (2.7)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \quad (2.8)$$

Eine Linearisierung über die Taylorreihe führt zu folgenden Berechnungsvorschriften für die Zustandsmatrizen, bei denen die partiellen Ableitungen der Funktionen gebildet werden müssen.

$$\mathbf{A} = \left( \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)_{\mathbf{x}_0, \mathbf{u}_0} \quad (2.9)$$

$$\mathbf{B} = \left( \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{\mathbf{x}_0, \mathbf{u}_0} \quad (2.10)$$

$$\mathbf{C} = \left( \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right)_{\mathbf{x}_0, \mathbf{u}_0} \quad (2.11)$$

$$\mathbf{D} = \left( \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \right)_{\mathbf{x}_0, \mathbf{u}_0} \quad (2.12)$$

Die Zustandsraumdarstellung lässt sich mit diesen Matrizen wie folgt ausdrücken:

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \quad (2.13)$$

$$\mathbf{y}(t) = \mathbf{C} \mathbf{x}(t) + \mathbf{D} \mathbf{u}(t) \quad (2.14)$$

Angewendet auf das Beispiel des Pendels ergeben sich die nachfolgend angegebenen Zustandsmatrizen. Zu beachten ist, dass es sich der einfachen Veranschaulichung halber um ein SISO-System handelt, bei dem die Matrizen  $\mathbf{B} = \mathbf{b}$ ,  $\mathbf{C} = \mathbf{c}^T$  und  $\mathbf{D} = \mathbf{d}$  entsprechen. [Lunze, 2010a, S. 106 ff.] [Wenck, 2012/13] [Holzhüter, 2012]

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\cos(x_{10}) & 0 \end{pmatrix} \quad (2.15)$$

$$\mathbf{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (2.16)$$

$$\mathbf{c}^T = (1 \ 0) \quad (2.17)$$

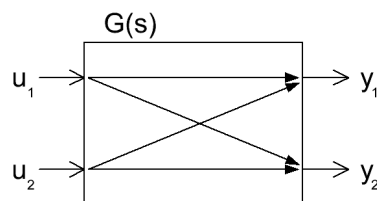
$$\mathbf{d} = 0 \quad (2.18)$$



## 2.4 Regelungsansätze bei Mehrgrößensystemen

Ein Mehrgrößensystem ist ein System mit mehreren Ein- und Ausgängen. Es zeichnet sich dadurch aus, dass eine Eingangsstellgröße Auswirkungen auf mehrere Ausgänge hat.

Diese Verkopplungen zwischen Ein- und Ausgängen sind beispielhaft in Bild 2.4 zu erkennen. Die Eingangsstellgröße  $u_1$  führt in diesem Fall zu einer Änderung der Ausgangsgrößen  $y_1$  und  $y_2$ . Gleiches gilt für das Verhalten des Eingangs  $u_2$  auf die Ausgänge. [Schulz und Graf, 2013, S. 23 ff.]



**Bild 2.4:** Verkopplungen einer Regelstrecke

Die Regelstrecke  $\mathbf{G}(s)$  hat im Laplace-Bereich nachfolgend aufgeführte Form mit  $m$  Zeilen und  $r$  Spalten:

$$\mathbf{G}(s) = \begin{pmatrix} G_{11}(s) & G_{12}(s) & \dots & G_{1m}(s) \\ G_{21}(s) & G_{22}(s) & \dots & G_{2m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{r1}(s) & \dots & G_{r2}(s) & G_{rm}(s) \end{pmatrix} \quad (2.19)$$

Wären nur die Hauptstreckenelemente  $\mathbf{G}_{ii}(s)$  in der Hauptdiagonalen der Matrix vorhanden und alle anderen Elemente gleich null, würde es sich um mehrere einzelne Regelstrecken handeln, die jedoch in einer Matrixschreibweise zusammengefasst wurden. Sobald Terme neben der Hauptdiagonalen auftreten, besitzt das System Querkopplungen. Bei Auftreten von Querkopplungen gibt es mehrere Ansätze diese Art Systeme zu regeln.

### 2.4.1 Vernachlässigung der Querkopplungen

Eine Möglichkeit der Regelung besteht darin, die vorhanden Querkopplungen der Regelstrecke zu vernachlässigen und z. B. mehrere konventionelle PID-Regler zu verwenden. Diese Möglichkeit besteht allerdings nur, wenn die Querkopplungen hinreichend klein sind.

Die Überprüfung der Stabilität des geregelten Systems kann in diesem Fall mit Hilfe von sogenannten Gershgorinbändern erfolgen. Über dieses Verfahren ist es möglich, abzuschätzen, in welchem Bereich die Eigenwerte des Systems liegen. [Lunze, 2010b, S. 369 ff.]

Allerdings soll diese Vernachlässigung der Querkopplungen nicht angewandt werden und wird aus diesem Grund nicht näher betrachtet. Es sei nur darauf hingewiesen, dass auch diese Möglichkeit der Regelung von Mehrgrößensystemen besteht, diese jedoch unter Umständen aufgrund zu großer Querkopplungen nicht anwendbar ist.

### 2.4.2 Entkopplung der Regelstrecke

Eine weitere Art der Regelung von Mehrgrößensystemen ist es, ein Entkopplungsglied für die Regelstrecke zu entwerfen. Dieses wird vor die zu regelnde Strecke geschaltet und sorgt dafür, dass eine Stellgrößenänderung am ersten Eingang nur noch Auswirkung auf den ersten Ausgang hat. Zu erkennen ist diese Art der Entkopplung in Bild 2.5, bei der die Strecke  $\mathbf{G}(s)$  durch ein Entkopplungsglied  $\mathbf{L}(s)$  entkoppelt wird.

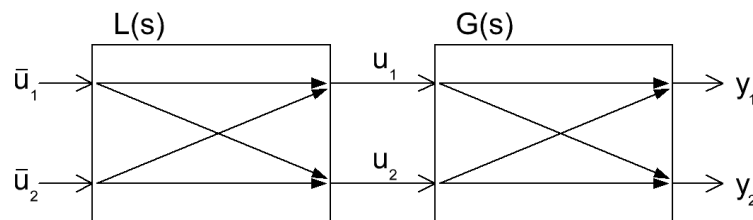


Bild 2.5: Entkopplung einer Regelstrecke

Ein Entkopplungsglied  $\mathbf{L}(s)$  für eine vollständige Entkopplung kann wie folgt berechnet werden:

$$\mathbf{L}(s) = \mathbf{G}(s)^{-1} \text{diag} \tilde{\mathbf{G}}_i(s) \quad (2.20)$$

Dabei können für die Diagonalelemente von  $\tilde{\mathbf{G}}_i(s)$  beliebige Übertragungsfunktionen gewählt werden.

Wird das Entkopplungsglied nach dieser Vorschrift entworfen, handelt es sich um eine dynamische Entkopplung.  $\mathbf{L}(s)$  kann allerdings auch als statische Entkopplung ausgelegt werden. Dazu wird die statische Verstärkung von  $\mathbf{L}(s)$  ermittelt. Zu beachten ist bei einer statischen Entkopplung, dass diese nicht zu einer vollständigen Entkopplung des Systems führen muss. In diesem Fall ist eine Stabilitätsprüfung der geregelten Strecke mittels der Gershgorinbänder durchzuführen.

Nach dem Entwurf dieser Entkopplung können die einzelnen entkoppelten Regelstrecken mit konventionellen Regelungsmethoden geregelt werden. Dazu eignen sich beispielsweise PID-Regler oder auch Zustandsregler. Die Auslegung kann über bekannte Methoden der Reglerauslegung erfolgen, da es sich lediglich um mehrere SISO-Systeme handelt.

### Reglerauslegung von SISO-Systemen durch Polvorgabe

Eine Möglichkeit der Reglerauslegung ist die Polvorgabe. Dieses Verfahren kann angewandt werden, wenn die Führungsübertragungsfunktion des geschlossenen Regelkreises die statische Verstärkung von eins und ein dominantes Polpaar besitzt. Das Verhalten des Regelkreises kann in diesem Fall durch ein PT2-Verhalten approximiert werden.

Die approximierte Führungsübertragungsfunktion lautet wie folgt:

$$\hat{G}_w(s) = \frac{1}{T^2 s^2 + 2dT s + 1} \quad (2.21)$$

Die Polstellen des geregelten Systems bestimmen bei dieser approximierten Betrachtung das Systemverhalten. Durch dynamische Güteforderungen an den Regelkreis können die gewünschten Polstellen des Systems ermittelt werden und anschließend die Reglerparameter bestimmt werden.

Die Polstellen des Systems  $\hat{G}_w(s)$  lauten wie folgt:

$$s_{1,2} = -\delta_e \pm j\omega_e \quad (2.22)$$

Als Gütekriterien können beispielsweise die Überschwingweite  $\Delta h$  und die Beruhigungszeit  $T_{5\%}$  gewählt werden. Die Überschwingweite  $T_{5\%}$  gibt dabei an, innerhalb welcher Zeit das System sich im Fünf-Prozentband eingeschungen hat. [Wenck, 2012/13]

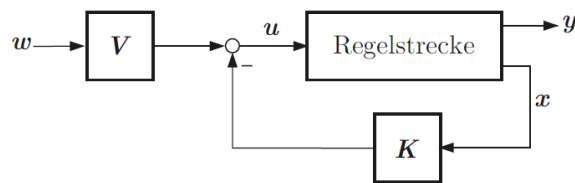
Die Polstellen können anhand dieser Gütekriterien wie folgt ermittelt werden:

$$T_{5\%} \approx \frac{3}{\delta_e} \quad (2.23)$$

$$\Delta h = \exp\left(\frac{-\delta_e}{\omega_e} \pi\right) \quad (2.24)$$

### 2.4.3 Zustandsregelung

Beim Entwurf einer Zustandsregelung werden alle vorhandenen Querkopplungen eines Systems berücksichtigt. Dazu ist es notwendig, alle Zustandsgrößen des Mehrgrößensystems im Idealfall messen zu können oder durch einen Beobachter zu schätzen. Das System muss für den Entwurf vollständig steuerbar sein. Soll ein Beobachter zur Schätzung der Zustände hinzugezogen werden, ist zudem eine vollständige Beobachtbarkeit des Systems notwendig.



**Bild 2.6:** Zustandsrückführung mit Vorfilter<sup>2</sup>

Bild 2.6 veranschaulicht den Aufbau einer Zustandsrückführung. Der Zustandsvektor  $\mathbf{x}$  wird über die Rückführmatrix  $\mathbf{K}$  auf die Systemeingänge  $\mathbf{u}$  zurückgeführt. Um ein Erreichen der vorgegebenen Sollwerte  $\mathbf{w}$  zu erreichen, muss ein Vorfilter  $\mathbf{V}$  entworfen werden. [Lunze, 2010b, S. 142 f.]

Beschrieben wird das Zustandsraummodell mit der Zustandsrückführung nach folgenden Gleichungen:

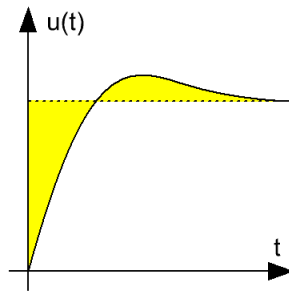
$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK}) \mathbf{x}(t) + \mathbf{BV} \mathbf{w}(t) \quad (2.25)$$

$$\mathbf{y}(t) = \mathbf{C} \mathbf{x}(t) \quad (2.26)$$

#### LQR-Reglerentwurf

Bei dem LQR-Verfahren handelt es sich um einen optimalen Reglerentwurf. Durch ein definiertes Gütefunktional werden die Regelflächen der Zustände  $\mathbf{x}(t)$  und Systemeingänge  $\mathbf{u}(t)$  minimiert. Der große Vorteil z. B. gegenüber einer Polzuweisung ist, dass alle Stell- und Regelgrößen bei dem Entwurf berücksichtigt werden. Die Herleitung und genaue Berechnung soll hier nicht behandelt werden. Eine sehr gute Erläuterung findet sich im Buch Regelungstechnik 2 von Jan Lunze ab Seite 300. Es soll allerdings nahegebracht werden, wie dieses Einstellverfahren zu verstehen und was dabei zu beachten ist. [Lunze, 2010b, S. 300 ff.]

<sup>2</sup>Bildquelle: [Lunze, 2010b, S. 142]



**Bild 2.7:** Regelfläche der Stellgröße  $u(t)$

Veranschaulicht werden kann die Minimierung der Regelflächen anhand von Bild 2.7, dass beispielhaft den Systemeingang bzw. die Stellgröße  $u(t)$  eines SISO-Systems bei einem Regelvorgang darstellt. Bei dem LQR-Verfahren werden die hier gelb markierten Regelflächen minimiert. Zur Einstellung der Regelung dienen zwei Wichtungsmatrizen  $\mathbf{Q}$  und  $\mathbf{R}$ . Über diese kann definiert werden, ob die Priorität darin liegt, einen schnellen Einschwingvorgang der Regelgrößen zu erreichen oder eher ein möglichst niedriger Energieverbrauch der Stellgrößen benötigt wird. Größere Werte in der Wichtungsmatrix  $\mathbf{Q}$  führen zu einem schnellerem Einschwingvorgang der Regelgrößen, wohingegen größere Werte in  $\mathbf{R}$  zu einer Minimierung des Energieverbrauchs des Systems führen. Die beiden Wichtungsmatrizen verhalten sich somit genau gegensätzlich, da für einen schnellen Einschwingvorgang der Zustandsgrößen hohe Stellgrößen bzw. Energieverbräuche benötigt werden. Soll beispielsweise ein Satellit in seiner Lage geregelt werden, spielt der Energieverbrauch eine große Rolle, da dieser möglichst lange autark betrieben werden soll. In diesem Fall sollten die Wichtungsmatrizen derart definiert werden, dass der Energieverbrauch möglichst niedrig ausfällt.

Um einen LQR-Reglerentwurf durchführen zu können, müssen einige Kriterien erfüllt werden, die nachfolgend aufgezeigt werden:

1.  $(\mathbf{A}, \mathbf{B})$  vollständig steuerbar
2.  $\mathbf{Q}$  symmetrisch, reell, positiv semidefinit
3.  $\mathbf{R}$  symmetrisch, reell, positiv definit
4.  $(\mathbf{A}, \overline{\mathbf{Q}})$  vollständig beobachtbar, mit  $\overline{\mathbf{Q}}^T \overline{\mathbf{Q}} = \mathbf{Q}$

Wurden die Wichtungsmatrizen  $\mathbf{Q}$  und  $\mathbf{R}$  gewählt und sind alle Kriterien erfüllt, kann über die Lösung der Matrix-Riccatigleichung und Berechnung der optimalen Zustandsrückführung  $\mathbf{K}$  die Funktion der Regelung getestet werden. Entspricht das Verhalten der Regelung nicht den Erwartungen, müssen die Wichtungsmatrizen angepasst werden. Dies erfolgt so lange, bis sich ein annehmbares Verhalten einstellt. Es ist in diesem Zusammenhang empfehlenswert, die Reglertests zunächst anhand von

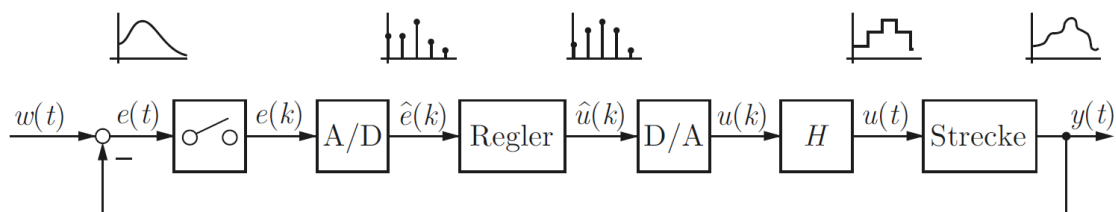
Simulationen durchzuführen, da nicht vorhergesagt werden kann, wie sich die ausgelegte Regelung mit den Wichtungsmatrizen verhält. In Matlab steht der Befehl „lqr“ zur LQR-Reglerentwicklung zur Verfügung.

## 2.5 Digitale Regelsysteme

Der Einzug der Nutzung von Computern bzw. Prozessoren findet auch in der Regelungstechnik statt und ersetzt in der heutigen Zeit fast vollständig analoge Regelsysteme. Eine kontinuierliche Regelung ist lediglich mit analogen Reglern zu realisieren. Eine Möglichkeit des Aufbaus besteht darin, Operationsverstärker einzusetzen, mit denen beispielsweise ein PID-Regler aufgebaut werden kann. Änderungen in der Parametrierung des Reglers müssen in diesem Fall allerdings über geänderte Bauteilwerte erfolgen. Die Folge dessen ist, dass dafür Bauteile ein- und ausgelötet werden müssen, vorausgesetzt, es werden keine Potentiometer eingesetzt.

Der große Vorteil von digitalen Reglern liegt darin, dass sich diese sehr vielseitig einsetzen und anpassen lassen. Unterschiedlichste Reglergesetze können implementiert werden und Parameteränderungen können einfach über das Einspielen einer neuen Firmware oder das Einstellen mittels integrierter Menüs erfolgen.

Sobald allerdings eine Regelung mittels eines Computers, Prozessors oder einer anderen digitalen Form realisiert wird, handelt es sich um eine digitale Regelung. Bei dieser müssen einige Besonderheiten beachtet werden, da sich ein digitaler und ein kontinuierlicher Regler der gleichen Reglerart unter gewissen Umständen nicht gleich verhalten.



**Bild 2.8:** Digitaler Regelkreis<sup>3</sup>

In Bild 2.8 ist ein digitaler Regelkreis zu erkennen. Der Unterschied zu einem kontinuierlichen analogen Regler besteht darin, dass der diskrete Regler nur zu bestimmten Abtastzeitpunkten neue Stellwerte berechnet. Zudem müssen für die Berechnung sowohl der Sollwert  $w(t)$  als auch der Istwert  $y(t)$  bzw. die Regeldifferenz  $e(t)$  in digitaler Form vorliegen. Dazu ist es notwendig, Analog-Digital-Wandler einzusetzen, die zur

<sup>3</sup>Bildquelle: [Lunze, 2010b, S. 410]

Messung und Digitalisierung der Messwerte dienen. Zudem muss der neu berechnete Stellwert  $\hat{u}(k)$  wieder der Strecke zugeführt werden. Hierfür findet ein Digital-Analog-Wandler Verwendung, dessen Ausgangsstellgröße  $u(k)$  über ein Halteglied solange beibehalten wird, bis zum nächsten Abtastzeitpunkt wieder ein neuer Stellwert zur Verfügung steht.

Bei der Wandlung von analogen in digitale Signale und umgekehrt sind technische Beschränkungen zu beachten. Ein Analog-Digital-Wandler kann Signale nur mit einer begrenzten Auflösung digitalisieren. Es kommt dadurch zu Quantisierungsfehlern. Die Verwendung von Analog-Digital-Wandlern führt allerdings auch zu Messrauschen, Offset- und Linearisierungsfehlern.

Bei der Systemnachbildung digitaler Regelkreise werden diese Wandlungen zwischen digitalen und analogen Signalen der einfacheren Berechenbarkeit halber als ideal betrachtet. Quantisierungsfehler, Messrauschen usw. werden also vernachlässigt. Zudem spielen diese Effekte aufgrund des technischen Fortschritts eine eher untergeordnete Rolle. Viel entscheidender ist hingegen die Abtastzeit des Reglers. Das beste Beispiel ist ein Sinussignal, das bei zu groß bzw. falsch gewählter Abtastzeit immer genau die Nullpunkte der Schwingung erfasst. Eine Regelung würde in diesem Fall scheitern. Sogenanntes Aliasing sollte also in jedem Fall vermieden werden. Nach dem Abtasttheorem von Shannon muss die Abtastfrequenz mindestens doppelt so groß sein wie die maximal auftretende zu messende Frequenz (Grenzfrequenz). Allerdings handelt es sich dabei um eine mathematische Betrachtung. Um in der Realität einen funktionsfähigen digitalen Regelkreis zu realisieren, sollten die Abtastfrequenzen deutlich höher ausfallen. [Lunze, 2010b, S. 409 ff.]

Ist die Abtastfrequenz 20- bis 30-mal größer als die Grenzfrequenz, kann von einer quasikontinuierlichen Regelung ausgegangen werden. Je höher die Abtastfrequenz gewählt wird, desto ähnlicher ist das Verhalten der eines kontinuierlichen Regelsystems. Die Auslegung der Regelung kann in diesem Fall im Kontinuierlichen erfolgen. Das Reglergesetz kann über verschiedene Approximationen aus dem Kontinuierlichen in ein Diskretes überführt werden. Es eignen sich beispielsweise das Euler-Verfahren oder das diskrete Äquivalent, zu finden im Buch Regelungstechnik 2 von Gerd Schulz und Klemens Graf ab Seite 273. [Schulz und Graf, 2013, S.273ff.] [Wenck, 2013]

Liegt keine Quasikontinuität vor, sollte ein zeitdiskreter Regler entworfen werden. [Schulz und Graf, 2013, S. 319 ff.]

Um ein vorliegendes System zu testen, kann die meist kontinuierliche Strecke z. B. über ein Halteglied 0-ter Ordnung (Halteglied Äquivalent) diskretisiert werden. Anschließend ist es möglich, das Regelsystem mit dem diskreten Reglergesetz zu testen und gegebenenfalls mit dem kontinuierlichen Verhalten zu vergleichen.

## 3 Systemaufbau und Modellbildung

Dieses Kapitel behandelt den Aufbau des Quadropters und die Vorbereitungen, die getroffen wurden, um die verschiedenen Regelungen in den folgenden Kapiteln implementieren zu können. Dabei müssen zum einen entsprechende Hardwarekomponenten ausgewählt und zum anderen Teile der Software vorbereitet werden. Weiter findet in diesem Kapitel die Modellbildung des Quadropters inklusive der Parameterbestimmung statt, da die spätere Auslegung der Regler an diesem Modell erfolgt.

### 3.1 Auswahl der Hardwarekomponenten

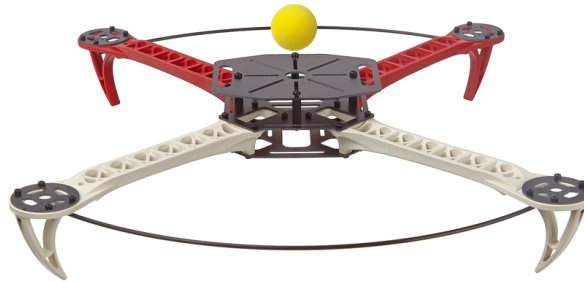
Für den praktischen Test der ausgelegten Regelungen wird ein Quadropter mit allen dazu notwendigen Komponenten aufgebaut. Dabei geht es nicht darum, ein marktreifes Modell aufzubauen, sondern es soll lediglich als Testmodell dienen.

#### 3.1.1 Quadroptergestell

Aus Kostengründen und Aspekten der Sicherheit wird für den Test ein eher kleines Modell gewählt, bei dem zwar die für bis zu 50 A ausgelegten Motorregler nicht voll ausgereizt werden, dieses jedoch für den Praxistest ausreichend sein sollte. Gewählt wurde dabei ein Quadroptergestell der 450er-Klasse, was bedeutet, dass der Abstand von Motorwelle zur gegenüberliegenden Motorwelle etwa 450 mm beträgt. Das Rahmengewicht liegt bei 320 g und die Zuladung ist mit 600 g bis 750 g, je nach ausgewählten Motoren, angegeben. Der Kaufpreis liegt bei unter 30 €, sodass auch ein Neukauf nach einem Absturz nicht weiter ins Gewicht fällt.

Das entsprechende Gestell ist in Bild 3.1 abgebildet. Das Gestell ist recht einfach aufgebaut und besteht aus drei Trägerplatten, die zur Befestigung der vier Ausleger dienen und ausreichend Unterbringungsmöglichkeiten für die Elektronik bieten.





**Bild 3.1:** Bestelltes Quadrokoptergestell<sup>4</sup>

### 3.1.2 Motoren und Propeller

Als Motoren müssen entsprechend des einzusetzenden Motorreglers aus meiner Bachelorarbeit bürstenlose Motoren für den sensorlosen Betrieb via Blockkommutierung ausgewählt werden. Das Gesamtgewicht des Quadropters wurde dabei vorerst grob auf ca. 1000 g geschätzt. Da alle vier Motoren sich die Traglast teilen, muss jeder Motor nur 1/4 des Schubes zum Tragen des Gesamtgewichts aufbringen. Jedoch muss eine entsprechende Reserve eingeplant werden, die in diesem Fall das drei- bis vierfache der durchschnittlichen Schubleistung beträgt. Diese Reserve ist notwendig, da die Motoren zusätzliche Schubkraft für das Steigen und die Lageregelung benötigen. Eine maximale Schubleistung pro Motor von 750 g bis 1000 g ist also sinnvoll.

Die Auswahl ist dabei auf Motoren des Modells MT2213 von EMAX gefallen, die standardmäßig mit dazu passenden links- und rechtsdrehenden Luftschrauben ausgeliefert werden. Die wichtigsten Nenndaten sind nachfolgend aufgelistet:

- Brushless-Außenläufermotor
- Spezifische Drehzahl: 935 kV
- Gewicht: 55 g
- Höhe x Durchmesser: 42,2 mm x 27,9 mm
- Akkus: Bis zu drei Lithiumpolymerakkus in Reihe
- Maximaler Schub: 850 g
- Reglerauslegung:  $\geq 18$  A
- Propellergröße: 10 x 45 "

<sup>4</sup>Bildquelle: <http://www.clever4buy.de/artikelbilder/14022.png> (Stand: 27.01.2014)



**Bild 3.2:** Bürstenloser Außenläufermotor EMAX MT2213

Die Motoren bieten die Besonderheit, dass die Propelleraufnahme direkt in der Motorwelle integriert ist, erkennbar in Bild 3.2. Die Motorwelle ist dabei mit einem Gewinde versehen. Durch die direkte Montage des Propellers auf der Motorwelle, ohne einen extra Spinner bzw. Adapter, sollen störende Vibrationen minimiert werden.

### 3.1.3 Lagesensor

Zur Messung der Lage des Quadropters im Raum bieten sich einige Möglichkeiten an. Mit Gyroskopen können die Drehraten um die drei Achsen bestimmt werden. Über eine Integration der Drehraten lässt sich letztlich die Lage berechnen. Es handelt sich in diesem Fall allerdings lediglich um eine Relativangabe. Es wird in diesem Zusammenhang auch von Dead-Reckoning-Sensoren gesprochen. Das Problem bei dieser Art Sensoren besteht in deren Eigenschaft, einen gewissen Drift aufzuweisen, der dazu führt, dass die Lageangabe mit größer werdender Zeit nicht mehr mit der absoluten Lageangabe übereinstimmt. Diese Art Drift tritt durch Offsetfehler, Nichtlinearitäten der Messung und einen nicht optimal positionierten Sensor auf. Zudem führen Vibrationen des Systems zu Messfehlern. Gyroskope, die auf der optischen Korrelation basieren, weisen zwar eine sehr geringe Fehlertoleranz auf, scheiden jedoch aufgrund der immensen Kosten und der zu großen Bauart aus. Eine andere Art von Gyroskopen basiert auf der Messung der Drehraten mittels Schwingungsbewegungen eines Körpers und Ausnutzung der Corioliskraft bei Rotation. Gerade bei dieser Art Sensoren treten starke Messfehler auf. Jedoch sind diese in sehr kleinen Bauformen in Chipgröße verfügbar und kostengünstig.

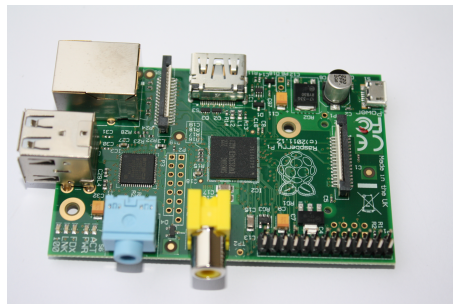
Die alleinige Messung der Lage mittels Gyroskop muss somit noch durch weitere Sensoren unterstützt werden, die für eine Korrektur der relativen Lage sorgen. In diesem Fall bieten sich Systeme mit Kameras zur optischen Korrelation an oder es besteht auch die Möglichkeit, Beschleunigungssensoren einzusetzen.

Es wird an dieser Stelle das derzeit in Quadrocoptern bewährte Prinzip der Sensorfusion aus Gyroskop und Beschleunigungssensor gewählt. Einer der führenden Hersteller ist die Firma InvSense. Diese bietet einen Chip mit einer Baugröße von 4x4x0,9 mm an, der ein 3-Achsen-Gyroskop sowie einen 3-Achsen-Beschleunigungssensor beinhaltet. Es handelt sich dabei um das MPU-6050-IC. Zudem ist im Chip ein Digital-Motion-Processor™ (DMP™) implementiert, der in der Lage ist, intern eine Sensorfusion zu übernehmen, um so den Hauptprozessor zu entlasten und die Daten bereits in fusionierter Form bereitzustellen. Eine Erweiterungsmöglichkeit stellt das MPU-9150-IC dar, das kompatibel zum MPU-6050-IC ist und als Zusatz einen 3-Achsen-Kompassensensor beinhaltet.

Für den ersten Test wurde das MPU-6050-IC erworben, das über eine I2C-Schnittstelle angesteuert wird. [MPU6050 Description] [MPU9150 Description]

### 3.1.4 Hauptprozessor

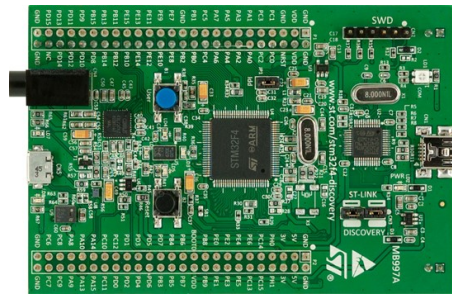
Der Hauptprozessor soll zur Auswertung der Lageinformationen, Regelung und Ansteuerung der Motorregler dienen. In Betracht gezogen wurden zwei unterschiedliche Systeme, die beide erworben und getestet wurden.



**Bild 3.3:** Raspberry Pi<sup>5</sup>

Zum einen handelt es sich um ein Raspberry Pi (Bild 3.3), welches einen günstigen Einplatinencomputer in Kreditkartenformat darstellt. Die vorhandenen Schnittstellen sind sehr vielfältig und umfassen auch eine I2C-Schnittstelle zur Ansteuerung der Motoren. Der Prozessor basiert auf der ARM-Architektur und taktet mit 700 MHz. Als Betriebssystem kommen verschiedene Linux-Distributionen oder auch andere Derivate zum Einsatz, die über eine eingeschobene SD-Karte gebootet werden. Zudem bietet Matlab eine direkte Simulink Anbindung, sodass in Simulink erstellte Modelle direkt auf das Raspberry geladen werden können. Eine Programmierung in z. B. C ist in diesem Fall nicht notwendig.

<sup>5</sup>Bildquelle: <http://upload.wikimedia.org/wikipedia/commons/3/30/RaspberryPiModelBRev2.by.Philipp.Bohk.jpg> (Stand: 11. 02. 2014)

Bild 3.4: STM32F4-Discovery<sup>6</sup>

Zum anderen wird ein STM32F4-Discovery-Board (Bild 3.4) in Betracht gezogen. Dieses basiert auf einem 32 bit ARM-Cortex-M4F-Chipsatz mit einer Taktrate von bis zu 168 MHz. Dieses Board bietet ebenso wie das Raspberry viele Arten von Schnittstellen, darunter drei I2C-Schnittstellen. Die Programmierung dieses Mikrocontrollers erfolgt üblicherweise in C. Der Mikrocontroller besitzt eine Floating-Point-Unit (FPU), die eine sehr schnelle Gleitkommaberechnung ermöglicht. Auf der Internetseite <http://blog.stm32f4.eu/category/fpu/> ist beispielsweise ein Test zu finden, in dem die Zeiten mit und ohne Nutzung der FPU miteinander verglichen werden. Bei durchgeführten 100000 Berechnungen einer einfachen Addition wurde ohne FPU eine Zeit von 35 ms gemessen, wohingegen der Einsatz der FPU zu einer Rechenzeit von nur 4 ms führte. Der Geschwindigkeitszuwachs ist somit deutlich vorhanden.

Tabelle 3.1: Vergleich der Hauptprozessorboards

Merkmal	Raspberry Pi	STM32F4-Discovery
Schnittstellen	+	+
Simulink-Anbindung	+	-
Bootzeit	-	+
Echtzeitfähigkeit	-	+
Interrupts	-	+
Floatberechnung	-	+
Abmessungen	+	0
<b>Summe</b>	-1	+4

In Tabelle 3.1 werden die wichtigsten Kriterien beider Boards miteinander verglichen und mit Plus, Minus und Null gewichtet. Sehr wichtig bei der Wahl des Prozessors ist dessen Echtzeitfähigkeit. Die Lageregelung des Quadropters fordert eine harte Echtzeit, weshalb es wichtig ist, möglichst auf Interruptprogrammierung zurückgreifen zu

<sup>6</sup>Bildquelle: [http://www.st.com/st-web-ui/static/active/en/fragment/product\\_related/rpn\\_information/board\\_photo/stm32f4\\_discovery.jpg](http://www.st.com/st-web-ui/static/active/en/fragment/product_related/rpn_information/board_photo/stm32f4_discovery.jpg) (Stand: 11. 02. 2014)

können, um asynchrone Ereignisse zeitnah zu bearbeiten und auszuwerten. In diesem Fall ist das STM32F4-Discovery-Board klar im Vorteil, da sehr viele Interrupts zur Verfügung stehen, die zudem mit verschiedenen Prioritäten versehen werden können. Das Raspberry Pi bietet zwar auch die Möglichkeit der Interruptprogrammierung, ist für diesen Zweck jedoch weniger geeignet.

Eine große Enttäuschung stellte der Test der Simulink-Anbindung des Raspberrys dar. Die Einrichtung ist gut beschrieben, jedoch dauert das Laden eines Modells auf das Raspberry etwa eine Minute oder länger. Diese Ladezeiten summieren sich sehr schnell und führten schon bei den ersten Tests zu einer persönlichen Abneigung. Außerdem ist die Auswahl an Simulink-Blöcken zur Ansteuerung von Peripherie nur sehr eingeschränkt, sodass eigener in C-geschriebener Quellcode eingebunden werden muss. Genutzt wurde Matlab in der Version 2013a. Ein weiterer Nachteil ist die kaum realisierbare Möglichkeit, asynchrone Ereignisse in Simulink zu bearbeiten.

Die Entscheidung ist letztendlich auf das STM32F4-Discovery-Board gefallen, da dieses bei richtiger Programmierung garantiert eine Echtzeitfähigkeit bietet, sehr schnelle Floatberechnungen ermöglicht und innerhalb nur weniger Millisekunden nach dem Einschalten einsatzbereit ist. Zudem ist auch neuer Quellcode schnell compiliert und es vergeht nur eine Wartezeit von wenigen Sekunden bis zum ersten Test. Die Programmierung des Boards in C kann aufgrund der bis zu diesem Zeitpunkt schlechten Anbindung des Raspberrys an Simulink eher als Vorteil gesehen werden.

Die Argumente sprechen in diesem Fall also deutlich für sich und das STM32F4-Discovery stellt die bessere Wahl dar.

### 3.1.5 Fernsteuerung

Die Fernsteuerung dient zur Steuerung des Quadropters, weshalb diese mindestens fünf Kanäle besitzen muss. Diese fünf Kanäle dienen der Steuerung von Pitch, Roll, Yaw, der Höhe und dem Ein- und Ausschalten der Motoren.

Wichtig bei der Wahl der Fernsteuerung ist bei der Anwendung an einem Quadropters die Möglichkeit, den Empfänger so konfigurieren zu können, dass dieser ein PPM-Summensignal ausgibt. Dabei handelt es sich um ein Signal, das alle zu übermittelnden Kanäle der Fernsteuerung in einem zusammenfasst. Dadurch ist ein vereinfachtes Auslesen aller Kanäle gewährleistet, ohne viele Leitungen verlegen und anschließen zu müssen.

Eine Neuerung bei den Modellbaufernbedienungen stellt die 2,4 GHz-Technik dar. Diese gewährleistet eine hohe Störsicherheit und eine hervorragende Signalübertragung durch digitale Übertragungstechnik.



**Bild 3.5:** Fernsteuerung Graupner MX-16

Die Graupner MX-16 bietet all diese Vorzüge, besitzt acht Kanäle und lässt sich für diverse Anwendungsfälle konfigurieren. Zu erkennen ist diese in Bild [3.5](#).

### 3.1.6 Akkumulator

Die Energieversorgung des Quadropters soll so ausgelegt werden, dass mindestens eine Flugzeit von 10 min erreicht werden kann. Als Akkutyp wurde ein moderner Lithiumpolymerakku gewählt, da dieser im Gegensatz zu Bleigel-, NiMH- und NiCd-Akkus die höchste Energiedichte aufweist. In Kauf genommen werden muss dabei jedoch, dass dieser Akkutyp sehr pfleglich behandelt werden muss. Ein Überladen des Akkus muss unbedingt vermieden werden, da sich die Zellen bei Überladung schnell entzünden können. Aus diesem Grund müssen immer Ladeelektroniken verwendet werden, die jede Zellspannung einzeln überwachen und gegebenenfalls auch einen Angleich aller Zellspannungen (Balancing) beherrschen. Ebenso kritisch ist eine Tiefentladung des Akkus, die zu starken Kapazitätsverlusten der Zellen führt. Eine rechtzeitige Abschaltung bei Betrieb verhindert somit einen frühzeitigen Austausch der Zellen.

Als Zellenzahl wurde die maximal mögliche Anzahl gewählt, die durch die ausgewählten Motoren begrenzt wird. Je höher die Betriebsspannung gewählt wird, desto niedriger sind bei gleicher Leistungsaufnahme die anfallenden Ströme. Spannungsverluste auf den Leitungen werden dadurch verringert. Die Motorregler sind für eine Versorgungsspannung von bis zu 25 V ausgelegt. Die von den Motorreglern bereitgestellte 5 V-Versorgungsspannung ist durch einen Step-Down-Schaltregler realisiert, weshalb eine höher gewählte Versorgungsspannung nur zu geringfügig höheren Verlusten führt.

Deshalb ergeben sich in diesem Fall durch die Wahl der höchstmöglichen Versorgungsspannung nur Vorteile.

Der Akku wurde erst nach der Schubmessung in Kapitel 3.6.1 bestellt. Als Gesamtgewicht des Systems wurde weiterhin von geschätzten 1000 g ausgegangen. So muss jeder der vier Antriebe 250 g heben. Laut Tabelle 3.3 kann dabei etwa von einer Drehzahl von 4000 U/min ausgegangen werden. Bei dieser Drehzahl kann jeder Motor, entsprechend Tabelle 3.3, 238 g anheben, was in erster Näherung den geforderten 250 g entspricht. Der Stromverbrauch liegt hier bei gerundeten 2,2 A pro Motor. Insgesamt fließt also ein Strom von 8,8 A.

Die minimale Akkukapazität berechnet sich damit wie folgt:

$$Q_{\text{Akkumin}} = I_{\text{ges}} \cdot T = 8,8 \text{ A} \cdot 10 \text{ min} = 8800 \text{ mA} \cdot \frac{10 \text{ min} \cdot \text{h}}{60 \text{ min}} = 1467 \text{ mAh} \quad (3.1)$$

Bestellt wurde ein Lithiumpolymerakku mit drei Zellen, einer Nennspannung von 11,1 V und einer großzügig ausgelegten Kapazität von 2500 mAh.

## 3.2 Aufbau und Test der Motorregler

Aus meiner Bachelorarbeit ist bereits ein komplett aufgebauter und funktionsfähiger Motorregler vorhanden. Dieser muss jedoch mit den neuen Motoren getestet werden. Dies erfolgt vor dem Bau der drei fehlenden Motorregler, um eventuell vorhandene Fehler in der Hard- und Software vorher beheben zu können. Dadurch werden Fehlbestellungen von Komponenten und zeitlich aufwändige Lötarbeiten durch Veränderungen der Bauteile vermieden.

Für den Test wurde der angeschlossene Motor bis zu seiner maximalen Strombelastbarkeit von 15 A betrieben. Dabei zeigte sich, dass gerade bei höheren Drehzahlen von ca. 6000 – 7000 U/min zufällig auftretende Kommentierungsfehler auftraten. Da die sensorlose Rotorlagenerkennung des Motorreglers hardwareseitig über Operationsverstärker gelöst ist, kann in diesem Fall lediglich die Auslegung der Filter für die rückinduzierten Motorphasenspannungen angepasst werden. Eine größere Störfestigkeit ergibt sich durch Wahl größerer Kondensatoren an den jeweiligen Spannungsteilern, die im Schaltplan der Motorregler im Modul „Sensorless detection (K)“ auf Seite 2 zu finden sind. Die Kondensatoren CK4, CK5 und CK6 wurden von 47 nF auf 100 nF erhöht. Durch diese Änderung konnten die auftretenden Kommentierungsfehler vollständig behoben werden.



Während der Bachelorarbeit wurde der Motorregler für sehr große Ströme von bis zu 50 A ausgelegt. Die neuen Motoren haben jedoch eine maximale Strombelastbarkeit von 15 A. Um die relativ niedrige Auflösung von 10 Bit des ADCs des Microcontrollers etwas zu kompensieren, wurde die Stromverstärkung der Strommessung auf der Platine angepasst. Dazu wurde der Widerstand RI2, auf Seite 2 im Schaltplan ersichtlich, von  $2000\ \Omega$  auf  $4700\ \Omega$  erhöht. Es ergibt sich ein maximal messbarer Strom von 40,96 A anstatt zuvor 96,25 A. Die entsprechenden Änderungen wurden zudem in der Software übernommen.

Die Abtastzeit des Drehzahlreglers wurde von 10 ms auf 5 ms reduziert, um kleinere Totzeiten bei der Sollwertübernahme beim Betrieb als Drehzahlregler zu erreichen.

Zudem musste ein Fehler beseitigt werden, der zu einer fehlerhaften Motorarmierung bei kontinuierlichem Senden alle 10 ms von null als Sollwert führte.

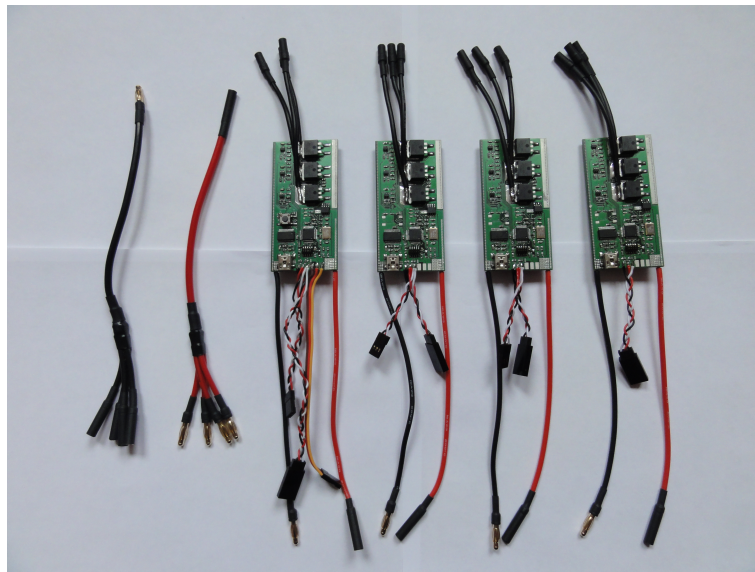
Ein weiteres Problem zeigte sich vor jedem Start der Motoren. Mit montiertem Propeller wird zunächst der eingestellte Startstrom über eine Rampe angeregt, bevor die manuelle Kommentierung beginnt. Bei diesem Anregen des Stroms stellte sich häufig eine Art Resonanz ein, sodass der Propeller bei Erreichen des Startstroms keine ruhige Position einnahm. Dieses Problem wurde durch eine weniger steile Startrampe für den Sollwertstrom unterbunden. Die Zeit für die Rampe wurde um den Faktor zehn erhöht.

Mit den neuen Motoren stellte sich zwar eine deutlich bessere Genauigkeit in der Drehzahlwertenerfassung ein, jedoch waren dennoch Drehzahlschwankungen bei konstanter Drehzahl von ca.  $\pm 100\ \text{U/min}$  zu erkennen, die sich durch die sensorlose Rotorlagenenerkennung ergeben. Da im späteren Verlauf die Drehzahlwerte verwendet werden sollen, wurde ein einfaches Filter für die Drehzahl implementiert. Zudem wurde die Software des Motorreglers dahingehend erweitert, dass ein Auslesen der aktuellen Drehzahl via I2C möglich ist.

Eine weitere Optimierung betrifft den I2C-Interrupt. Dabei wurde der Interrupt für die Erkennung einer Start-Stopp-Bedingung auf dem I2C-Bus entfernt, da mehrere Motorregler an einem I2C-Bussystem eingesetzt werden, die jeweils ihre eigenen Sollwerte gesendet bekommen. Bei jedem Senden, egal an welchen Motorregler, würde sonst immer der Start-Stop-Interrupt bei jedem Motorregler aufgerufen werden. Die komplette Abarbeitung eines neuen Sollwerts erfolgt nun im I2C-Interrupt. Dazu wurde ein Bit im Interruptregister genutzt, das das Empfangen des ersten gesendeten Bytes an die eigene Adresse erkennt.

Um eine schnelle Abschaltung im Fehlerfall, zum Beispiel bei einem Leitungsbruch in der I2C-Leitung, zu erreichen, wurde das Intervall für das Senden eines neuen Sollwerts auf 100 ms verkürzt. Wird innerhalb von 100 ms kein neuer Sollwert an den Motorregler gesendet, erfolgt umgehend die Abschaltung des Motors und der Motorregler muss durch das Senden von null neu armiert werden.





**Bild 3.6:** Fertig aufgebaute Motorregler

Nach diesen eingeführten Änderungen wurden die verbleibenden drei Motorregler aufgebaut und die neue Firmware aufgespielt. Leider werden die eingesetzten Mikrocontroller von Texas Instruments nicht mehr produziert, obwohl diese auf der noch aktuellen Cortex-M3-Architektur basieren und zur Zeit der Entwicklung des Motorreglers in meiner Bachelorarbeit noch neu im Sortiment waren. Dadurch mussten die Mikrocontroller mit hohem Kostenaufwand aus den USA bestellt werden. Dieses Verhalten von Texas Instruments, Produkte schon kurz nach der Markteinführung nicht mehr zu produzieren, führt in weiterer Folge zu einer Meidung dieser Firma bei meiner persönlichen Bauteilwahl. Ein vergleichbares Nachfolgeprodukt ist nicht vorhanden und ein Mikrocontroller lässt sich nicht wie ein Transistor schnell durch ein anderes Bauteil ersetzen. Die fertigen vier Motorregler mit konfektionierten Anschlussleitungen, die im Quadrocopter Verwendung finden, sind in Bild 3.6 ersichtlich.

### 3.3 Konstruktion des 3D-Modells des Quadrocopters

Vor dem Zusammenbau des Quadrocopters wurden alle Einzelteile für eine exakte Nachbildung nachgemessen, mit einer Digitalwaage gewogen, teilweise eingescannt und anschließend in Solidworks, einem 3D-CAD-Programm, konstruiert. Dieses 3D-Modell dient zum einen dazu, eigene modifizierte Komponenten konstruieren und herstellen zu können und zum anderen kann eine exakte Bestimmung der Trägheitsmomente

des Quadropters erfolgen. Ein weiterer Vorteil besteht darin, durch einen Absturz beschädigte Teile schnell durch selbst hergestellte ersetzen zu können.

Die neu konstruierte untere Trägerplatte wurde so ausgelegt, dass der Akku mit Klettverschluss unter dem Modell montiert werden kann. Zudem wurden Bohrungen zur Montage des Empfängers und des SD-Kartenmoduls vorgesehen. Weiter wurde die obere Trägerplatte neu konstruiert, um eine Aufnahme des Evaluationsboards zu ermöglichen. Die beiden neuen Trägerplatten wurden mittels einer CNC-Fräse zugeschnitten.



**Bild 3.7:** Vergleich zwischen 3D-Modell (links) und realem Modell (rechts)

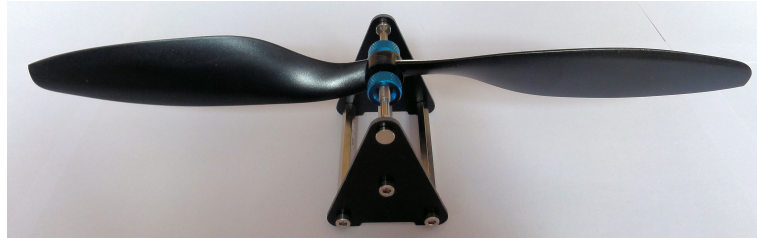
In Bild 3.7 ist ein Vergleich zwischen dem konstruierten 3D-Modell und dem realen Modell zu erkennen.

## 3.4 Aufbau und Inbetriebnahme des Quadropters

Für ein zufriedenstellend funktionierendes Modell müssen auch beim Aufbau einige Schritte beachtet werden, die unter anderem großen Einfluss auf die spätere Lageregelung haben können. Zudem wird ein erster Flugtest des aufgebauten Gesamtsystems durchgeführt, um eventuelle Fehler zu beseitigen und das Modell auf Funktion hin zu testen.

### 3.4.1 Wuchten der Propeller

Einen äußerst wichtigen Schritt vor dem Aufbau des Quadropters stellt das Wuchten der Propeller dar. Die vorhandenen Kunststoffpropeller sind im Auslieferungszustand durch die Fertigungstoleranzen nicht gewuchtet. Das Wuchten dient dazu, dass die Propeller, die sich mit bis zu 7000 U/min drehen, möglichst wenig Vibrationen erzeugen. Gerade Vibrationen führen zu Störungen bei der Lageerfassung mittels Gyroskop und Beschleunigungssensor, die durch diese einfache Maßnahme des Wuchtens hervorragend eliminiert bzw. minimiert werden können. Es ist somit ein unabdingbarer Schritt.



**Bild 3.8:** Propellerwaage mit gewuchtetem Propeller

Das Wuchten geschieht mittels einer Propellerwaage. Dazu wird der Propeller auf eine Welle geschraubt und in ein Gestell gehängt, das eine magnetische Lagerung für einen nahezu reibungsfreien Lauf aufweist. In Bild 3.8 ist die verwendete Propellerwaage zu erkennen. Wird der Propeller in diese Waage eingehängt, neigt sich dieser bei einer vorhandenen Unwucht zu der Seite, auf der das höhere Gewicht vorhanden ist. Im Idealfall sollte sich der Propeller zu keiner Seite drehen. Das Wuchten kann dann einerseits durch Materialabnahme (Abschaben mit einem Messer) oder durch Aufkleben von Klebefilmstreifen erfolgen, bis sich ein zufriedenstellendes Ergebnis einstellt und sich der Propeller zu keiner Seite mehr neigt. Anschließend kann dieser auf der Motorwelle montiert werden.

### 3.4.2 Aufbau des Quadropters

Nach dem Wuchten der Propeller konnte der Quadropter komplett aufgebaut werden. Der Aufbau ging dabei leicht vonstatten, da das Modell bereits komplett virtuell in Solidworks zusammengebaut wurde. Die vier Motorregler wurden jeweils auf jedem Arm mit Kabelbindern montiert. Zudem wurden diese zum Schutz in Schrumpfschlauch verpackt. Geachtet wurde beim Aufbau auf einen ausreichend großen Leiterquerschnitt für die Zuleitungen zu den Motorreglern, da alle vier im Maximalfall bis zu 60 A benötigen. So wird erhöhten Spannungsverlusten auf den Leitungen entgegengewirkt und Massefehler sollten ausbleiben. Alle angeschlossenen Komponenten wurden über geeignete Steckverbinder miteinander verbunden, sodass ein einfacher Ein- und Ausbau aller Komponenten ermöglicht wird.

Der Akku wird wie geplant unter dem Modell mit Klettverschluss platziert, der Empfänger und das SD-Kartenmodul finden auf der unteren Trägerplatte Platz und das Evaluationsboard ist auf der oberen Trägerplatte aufgesteckt. Die 5 V-Versorgung aller Komponenten erfolgt komplett über einen Motorregler, der eine 5 V-Spannungsversorgung mit einem Strom von bis zu 3 A bereitstellt.

### 3.4.3 Auslesen der Fernsteuersignale

Um das Flugmodell steuern zu können, müssen die Fernsteuersignale vom Discovery-Board ausgewertet werden. Die Signale werden vom Empfänger als PPM-Summensignal bereitgestellt. Dabei handelt es sich um ein zusammengefasstes Signal von allen Kanälen. Die Einzelwerte der Kanäle werden dabei nacheinander als Rechteckspannungen mit unterschiedlichen Breiten ausgesandt. Es wird in diesem Zusammenhang von PPM bzw. Puls-Pausen-Modulation gesprochen. [[mikrokoetter Summensignal](#)]

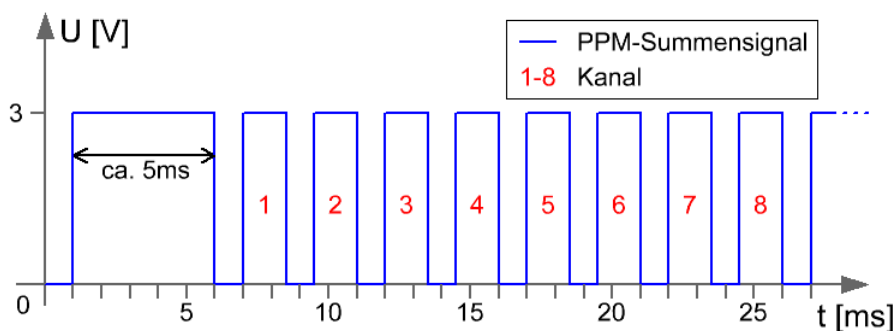
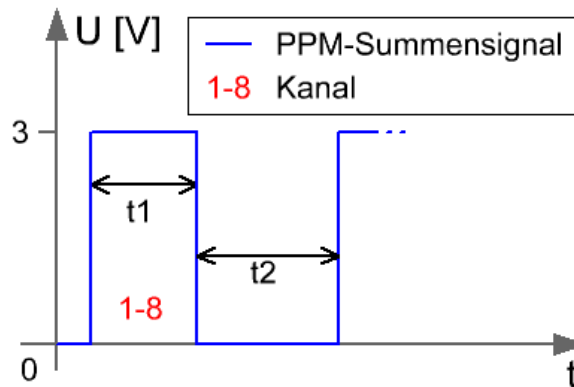


Bild 3.9: Verlauf des PPM-Summensignals

Der Empfänger Ausgang wurde mit einem analogen Oszilloskop nachgemessen. Der Verlauf wird jedoch zur Veranschaulichung und der besseren Darstellbarkeit nachgezeichnet. Bild 3.9 zeigt den Aufbau eines derartigen PPM-Summensignals, das vom Empfänger gesendet wird. Die einzelnen Kanäle sind nummeriert. Die Zeit zwischen zwei aufeinanderfolgenden Summensignalen beträgt ca. 5 ms, abhängig von den Signalzeiten aller Einzelkanäle.



**Bild 3.10:** Vergrößerte Ansicht des PPM-Summensignals

In Bild 3.10 ist eine vergrößerte Ansicht eines gesendeten Kanals zu erkennen. Die gemessenen Zeiten für  $t_1$  und  $t_2$ , entsprechend der Hebelstellung der Fernbedienung, werden in Tabelle 3.2 dargestellt.

**Tabelle 3.2:** PPM-Summensignalzeiten von einem Kanal

Hebelstellung [%]	$t_1$ [ms]	$t_2$ [ms]
0	1,0	0,5
+100	1,4	0,5
-100	0,6	0,5

Aus der Tabelle 3.2 wird deutlich, dass die Hebelstellung der Fernbedienung über die Zeit  $t_1$  übermittelt wird. Der Abstand zum darauffolgenden Signal beträgt konstant 0,5 ms, was auch die variable Pause zwischen zwei aufeinanderfolgenden Summensignalen erklärt.

Im Quellcode in der Datei „ppm.c“ erfolgt die Messung des Signals derart, dass die Zeiten zwischen zwei positiven Flanken mittels eines Interrupts im Mikrocontroller gemessen werden. Dies entspricht somit der Summation der Zeiten  $t_1$  und  $t_2$  aus Bild 3.10. Die gemessenen Zeiten aller Kanäle werden auf die Zeiten von  $t_1$  zurückgerechnet, in einem Array abgespeichert und anschließend in entsprechende Werte in Prozent umgerechnet.

### 3.4.4 Auslesen des Lagesensors

Der von der Firma InvSense beworbene integrierte Prozessor zur Sensorfusion (DMP™) ist leider nicht quelloffen. Dieser ist nicht dafür bestimmt, von Privatpersonen programmiert zu werden. Im Datenblatt finden sich keinerlei Informationen zur Programmierung. [[MPU6050 Datasheet](#)]

Es besteht somit normalerweise lediglich die Möglichkeit, die Rohdaten der Beschleunigungs- und Gyroskopsensoren auszulesen. Ein paar findige Programmierer haben jedoch die Daten von einem Evaluationsboard aufzeichnen können (reverse-engineering), mit dem der Prozessor programmiert wird. Dadurch ist es zumindest möglich, die interne Sensorfusion zu nutzen, ohne jedoch eigene Anpassungen vornehmen zu können.

Die Sensorfusion soll nicht Bestandteil dieser Arbeit sein, weshalb auf Quellcode zurückgegriffen wurde, der den integrierten Prozessor nutzt. Eine Anpassung des Codes zur Verwendung mit dem STM32F4 ist dennoch notwendig.

Der DMP™ fusioniert die Sensordaten, welche in Quaternions abgespeichert werden. Anschließend werden die Quaternions sowie die Rohdaten in einem FIFO abgelegt und es findet über eine extra Datenleitung eine Signalisierung statt, dass neue Daten ausgelesen werden können. Das Auslesen des FIFOs erfolgt zeitnah mittels eines Interrupts des STM32F4 über eine I2C-Schnittstelle. [[MPU6050 DMP](#)]

Anzumerken ist noch, dass die im DMP™ laufende Software eine sehr ausgeklügelte Driftkompensation beinhaltet, die in der Lage ist, während der Laufzeit die Offsets aller drei Gyroskope anzupassen.

### 3.4.5 Datenaufzeichnung auf der SD-Karte

Für die Aufzeichnung aller relevanten Flugdaten findet eine SD-Karte Verwendung. An dieser Stelle wurde ein Arduino-SD-Kartenleser erworben, der jedoch lediglich für den Betrieb mit einer SPI-Schnittstelle ausgelegt ist. Für STM32F4-Mikrocontroller ist allerdings bereits eine fertige Bibliothek zur Ansteuerung vorhanden. [[STM32F4 SDIO](#)]

Um diese Bibliothek nutzen zu können, die auch ein entsprechendes FATFS-Dateisystem beinhaltet, wurde das bestellte Kartenlesemodul hardwareseitig angepasst, sodass der Betrieb über die SDIO-Schnittstelle ermöglicht wird. Diese Schnittstelle zeichnet sich zudem durch eine parallele und somit schnellere Datenübertragung aus.

Nach dieser hardwareseitigen Änderung war es ohne größere Schwierigkeiten möglich, TXT-Dateien auf eine beliebige SD-Karte zu schreiben.



### 3.4.6 Funktionstest des Gesamtsystems

Um das Zusammenspiel aller verbauten Systemkomponenten zu testen, wird ein erster Flugtest des Modells durchgeführt. Dazu werden die Motorregler im Drehzahlregelbetrieb verwendet. Die vier Drehzahlregler werden alle gleich parametrisiert. Die Entkopplung der Regelgrößen erfolgt in diesem Fall manuell über das Wissen der Funktion des Quadropters. Es handelt sich eher um eine schnelle Hobbylösung als um eine wissenschaftliche Herangehensweise, weshalb auf die genaue Vorgehensweise nicht weiter eingegangen werden soll.

Der große Unterschied zu der in dieser Arbeit entworfenen Regler besteht darin, dass die Motorregler für den ersten Test im Drehzahlregelbetrieb genutzt werden. Durch empirisches Einstellen konnten Parameter für die vier PID-Regler zur Regelung des Auftriebs, von Pitch, Roll und Yaw gefunden werden.

Die Flugfähigkeit dieses Systems ist zwar gegeben, dieses reagiert allerdings äußerst anfällig bei kleineren Windböen, die fast zum Absturz führen. Auf weitere nachträgliche Optimierungen wird verzichtet, da dies nicht das Ziel des Versuchs ist.



**Bild 3.11:** Erstflug des Quadropters

Es konnte mittels dieser Methode die grundsätzliche Funktionsfähigkeit des Systems bewiesen werden. In Bild 3.11 ist der Quadropters bei seinem Erstflug zu sehen.

Bei den Tests trat vor allem ein Problem bei der Verwendung der SD-Karte zur Flugdatenaufzeichnung auf. Die SD-Karte sollte ursprünglich auf der unteren Trägerplatte montiert werden. Dies führte jedoch zu Störungen der einzelnen Bussysteme. Die genaue Ursache konnte nicht ermittelt werden. Die nahe Lage der SD-Karte zu den stromführenden Akkuleitungen kann jedoch nicht als optimal betrachtet werden. Auch eine Schirmung des SD-Kartenmoduls führte zu keiner Verbesserung, weshalb dieses kurzerhand neben dem STM32F4-Discoveryboard montiert wurde.

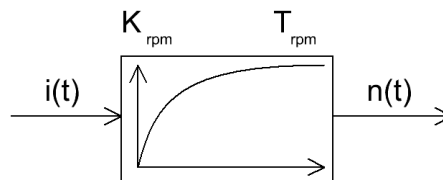
Bis auf die Probleme mit der SD-Karte traten keine weiteren Probleme auf. Durch die Datenaufzeichnung während des Fluges konnte ein mittlerer Stromverbrauch pro Motor im Schwebeflug von etwa 2500 mA ermittelt werden.

## 3.5 Modellbildung des Quadropters

In diesem Abschnitt findet die Modellbildung des Quadropters statt. Für die Beschreibung der Lage werden die englischen Begriffe Pitch, Roll, Yaw verwendet, da diese in gleicher Form im Quellcode Verwendung finden.

### 3.5.1 Rotordrehzahlen

Die vier Motorregler werden im Stromregelbetrieb genutzt. Die verwendeten Einstellungen finden sich im Anhang. Vereinfacht wird in diesem Fall davon ausgegangen, dass der vorgegebene Sollstrom verzögerungsfrei angeregt wird, da die Abtastzeit der Stromregelung bei 16 kHz liegt und der Stromregler entsprechend schnell eingestellt ist.



**Bild 3.12:** Nachbildung der Drehzahlregelstrecke

Eine Änderung der Rotordrehzahl wird somit über die Änderung des Eingangsstroms erreicht. Die Nachbildung des Verhaltens der Drehzahl auf eine Stromänderung wird aufgrund der vereinfachten Betrachtung durch ein PT1-Glied nachgebildet. Bild 3.12 veranschaulicht die Nachbildung der Drehzahlstrecke. Ein Strom  $i(t)$  führt zu einer Änderung der Drehzahl  $n(t)$  mit der Streckenverstärkung  $K_{rpm}$  und der Zeitkonstanten  $T_{rpm}$  des PT1-Glieds.

Die nachfolgenden Gleichungen 3.2, 3.3, 3.4 und 3.5 beschreiben dieses Verhalten für alle vier Motoren, ausgedrückt in Differentialgleichungen. Die Zustände  $x_1$  bis  $x_4$  entsprechen den Drehzahlen der Motoren M0 bis M3 in U/min und die Eingänge  $u_1$  bis  $u_4$  sind die Eingangsströme für die Motoren M0 bis M3 in mA.

$$\dot{x}_1(t) = -\frac{1}{T_{rpm}} \cdot x_1(t) + \frac{K_{rpm}}{T_{rpm}} \cdot u_1(t) \quad (3.2)$$

$$\dot{x}_2(t) = -\frac{1}{T_{rpm}} \cdot x_2(t) + \frac{K_{rpm}}{T_{rpm}} \cdot u_2(t) \quad (3.3)$$



$$\dot{x}_3(t) = -\frac{1}{T_{rpm}} \cdot x_3(t) + \frac{K_{rpm}}{T_{rpm}} \cdot u_3(t) \quad (3.4)$$

$$\dot{x}_4(t) = -\frac{1}{T_{rpm}} \cdot x_4(t) + \frac{K_{rpm}}{T_{rpm}} \cdot u_4(t) \quad (3.5)$$

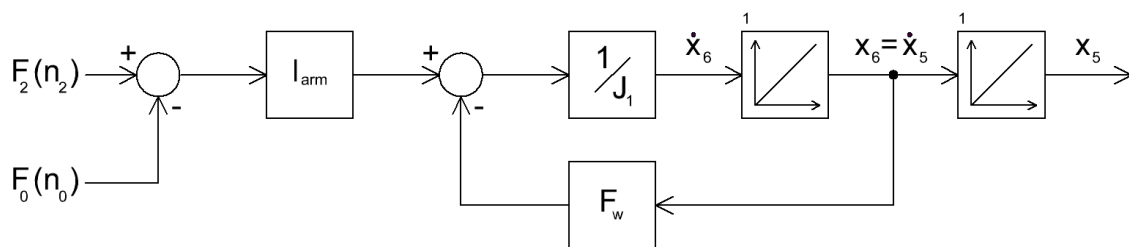
### 3.5.2 Pitch

Wie schon in Kapitel 2.2 erläutert, sind für die Änderung der Pitchlage die Schubkräfte der Propeller der Motoren M0 und M2 verantwortlich.

Die Schubkräfte stehen in Abhängigkeit von der Drehzahl der Motoren. Das Verhalten des Schubes in Abhängigkeit der Drehzahl soll durch eine quadratische Funktion abgebildet werden, wie sie in Gleichung 3.6 zu erkennen ist. Die Kraft  $F$  in N ergibt sich als Funktion von der Drehzahl  $n$  in U/min. Die Parameter  $a$ ,  $b$  und  $c$  müssen später noch ermittelt werden.

$$F(n) = a \cdot n^2 + b \cdot n + c \quad (3.6)$$

Der Wirkschaltplan ist in Bild 3.13 erkennbar. Die Kräfte  $F_0(n_0)$  und  $F_2(n_2)$  der zugehörigen Motoren M0 und M2 führen mit der Länge  $l_{arm}$  in m des Auslegers des Quadropters zu einem Drehmoment, welches mit Berücksichtigung des Trägheitsmoments  $J_1$  zu einer Drehung des Modells um die Pitchachse führt. Die Armlänge ist der Abstand vom Mittelpunkt des Quadropters zur Motorwelle. Der Zustand  $x_6$  beschreibt die Winkelgeschwindigkeit in rad/s und  $x_5$  gibt den Pitchwinkel in rad an.



**Bild 3.13:** Nachbildung der Strecke Pitch

Der Luftwiderstand  $F_w$  wird im Modell als einfacher Faktor eingeführt. In diesem Fall wird schon vorab eine Vereinfachung vorgenommen. Der Luftwiderstand hat üblicherweise eine quadratische Abhängigkeit zur Geschwindigkeit. Bei einer späteren Linearisierung des Systems bei einer Geschwindigkeit von Null würde sich jedoch der

Faktor  $F_w$  zu Null ergeben und würde somit nicht mehr berücksichtigt. Im Kapitel 4.2 soll allerdings eine Berücksichtigung des Luftwiderstands mit einbezogen werden.

Die Gleichungen 3.7 und 3.8 zeigen die Differentialgleichungen, die sich aus Bild 3.13 ergeben.

$$\dot{x}_5(t) = x_6(t) \quad (3.7)$$

$$\dot{x}_6(t) = (ax_3^2(t) - ax_1^2(t) + bx_3(t) - bx_1(t)) \cdot \frac{l_{arm}}{J_1} - \frac{F_w}{J_1} \cdot x_6(t) \quad (3.8)$$

### 3.5.3 Roll

Die Gleichungen für die Rolllage ergeben sich ebenso wie für die Pitchlage, mit dem Unterschied, dass die Motoren M1 und M3 für die Lageänderung verantwortlich sind. Die Gleichungen 3.9 und 3.10 veranschaulichen die daraus folgenden Differentialgleichungen.

$$\dot{x}_7(t) = x_8(t) \quad (3.9)$$

$$\dot{x}_8(t) = (ax_4^2(t) - ax_2^2(t) + bx_4(t) - bx_2(t)) \cdot \frac{l_{arm}}{J_1} - \frac{F_w}{J_1} \cdot x_8(t) \quad (3.10)$$

### 3.5.4 Yaw

Eine Lageänderung um den Yaw-Winkel ergibt sich aus den Drehmomenten aller vier Motoren. Die verwendeten bürstenlosen Motoren verhalten sich aufgrund der eingesetzten Rechteckkommutierung wie Gleichstrommotoren. Der Unterschied besteht lediglich in der elektronischen Kommutierung der Motorspulen. Bild 3.14 veranschaulicht das Ersatzschaltbild eines Gleichstrommotors.

Dabei gelten die folgenden drei Grundgleichungen 3.11, 3.12 und 3.13:

$$U_q = K \cdot \Phi \cdot \Omega \quad (3.11)$$

$$M_i = K \cdot \Phi \cdot I_A \quad (3.12)$$

$$\Phi = c \cdot N_E \cdot I_E \quad (3.13)$$

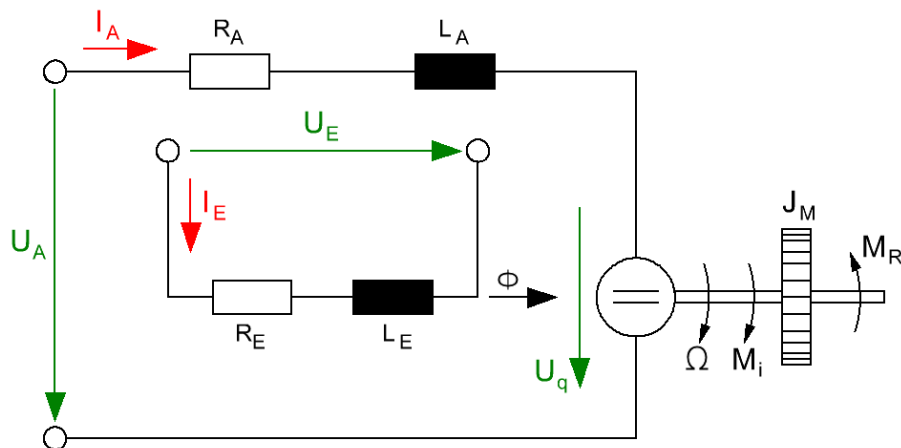


Bild 3.14: Ersatzschaltbild eines Gleichstrommotors

Der Erregerfluss  $\Phi$  kann durch die Verwendung von permanenterregten Motoren als konstant angenommen werden. Bei idealer Umwandlung von elektrischer in mechanische Energie kann das resultierende Drehmoment  $M_i$  aus dem eingepprägten Ankerstrom  $I_A$  nach Gleichung 3.12 berechnet werden. Lediglich die Motorkonstante  $K$  und der Hauptfluss  $\Phi$  des verwendeten Motors müssen ermittelt werden. [Röther, 2013, S. 13 ff.]

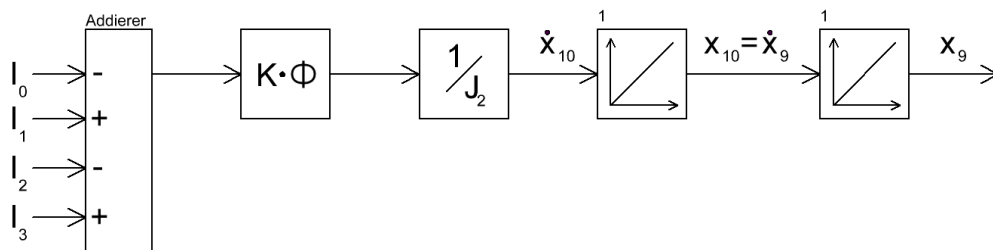


Bild 3.15: Nachbildung der Strecke Yaw

Das Wirkschaltbild ergibt sich damit wie in Bild 3.15 gezeigt. Eine Berücksichtigung des Luftwiderstandes findet in diesem Fall nicht statt. Aus diesem Wirkschaltbild ergeben sich die beiden Differentialgleichungen 3.14 und 3.15. Da die vier Eingangsgrößen  $u_1$  bis  $u_4$  in mA vorliegen, wird die Gleichung 3.15 noch durch 1000 dividiert, um die korrekte Berechnung der Drehmomente aus den Motorströmen zu gewährleisten.

$$\dot{x}_9(t) = x_{10}(t) \quad (3.14)$$

$$\dot{x}_{10}(t) = (u_1(t) + u_3(t) - u_2(t) - u_4(t)) \cdot \frac{K \cdot \Phi}{J_2 \cdot 1000} \quad (3.15)$$

### 3.5.5 Wahl der Ausgänge

Die Ausgänge werden so gewählt, dass diese in der Anzahl genau denen der vier Eingänge entsprechen. Dadurch wird eine exakte Berechnung des Vorfilters in Kapitel 4.1 gewährleistet, da bei gleicher Anzahl an Ein- und Ausgängen keine Pseudoinverse gebildet werden muss. Dies wird erreicht, indem die vier Drehzahlen, die für den Auftrieb verantwortlich sind, gemittelt und in einem Ausgang zusammengefasst werden. Dadurch kann nach wie vor der Auftrieb bestimmt werden, es wird jedoch nur ein Ausgang ( $y_1$ ) benötigt. Die anderen drei Ausgänge beschreiben die Lage des Quadropters für Pitch ( $y_2$ ), Roll ( $y_3$ ) und Yaw ( $y_4$ ).

$$y_1(t) = \frac{1}{4} \cdot (x_1(t) + x_2(t) + x_3(t) + x_4(t)) \quad (3.16)$$

$$y_2(t) = x_5(t) \quad (3.17)$$

$$y_3(t) = x_7(t) \quad (3.18)$$

$$y_4(t) = x_9(t) \quad (3.19)$$

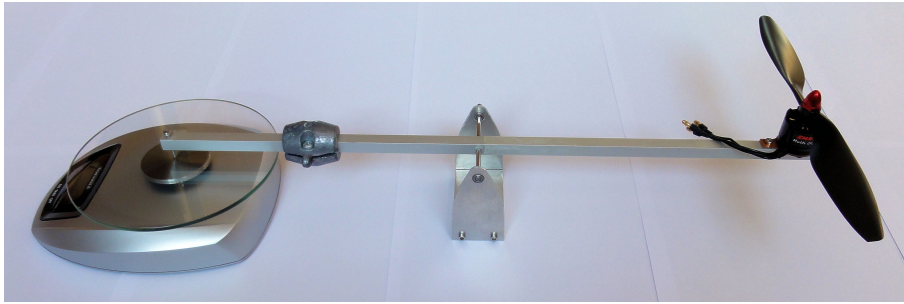
## 3.6 Ermittlung der Modellparameter

Entscheidend für den weiteren Reglerentwurf ist eine möglichst exakte Bestimmung der Streckenparameter.

### 3.6.1 Verhalten des Propellerschubes in Abhängigkeit der Drehzahl

Der Schub wird im Modell in Abhängigkeit von der Rotordrehzahl dargestellt, weshalb eine Messung erfolgt, die das Verhalten der Drehzahl auf den Propellerschub ermittelt. Eine Möglichkeit der Schubmessung besteht darin, mittels einer Waage die Gewichtskraft zu messen.

Dazu wurde eine Wippe in Solidworks konstruiert und anschließend aus Aluminium angefertigt. Die fertige Wippe mit montiertem Motor ist in Bild 3.16 zu sehen. Diese wurde so ausgelegt, dass die Abstände, vom Drehpunkt der Wippe aus gesehen, zur



**Bild 3.16:** Wippe für die Schubmessung

Motorwelle und zum Auflagepunkt auf der Waage gleich sind. So kann die Gewichtskraft eins zu eins, bedingt durch gleiche Hebelarmlängen, auf der Waage abgelesen werden. Zudem wurde Wert auf die Lagerung der Wippe gelegt. Um keine Verfälschungen in den Messergebnissen zu erhalten, wurden Kugellager eingesetzt. Dadurch ist eine nahezu reibungslose Bewegung möglich.

Für die Messung wurde die Wippe mit einer Schraubzwinge auf einem Tisch in 70 cm Höhe montiert. Die Temperatur betrug gemessene  $22,5^{\circ}\text{C}$ , da auch die Luftdichte Einfluss auf die Messergebnisse nimmt. Als Spannungsquelle wurde ein 12 V Bleigelakku verwendet. Zusätzlich wurden neben der Messung mittels Digitalwaage der Eingangsstrom des Motorreglers und die Akkuspannung gemessen, wobei die Messgeräte nur bis zu einem Strom von unter 10 A angeschlossen waren.

Die Messergebnisse sind in Tabelle 3.3 ersichtlich. Das gemessene Gewicht der Waage wurde zudem in einen entsprechenden Schub in Newton umgerechnet, bei dem eine Erdanziehungskraft von  $9,81\text{ m/s}^2$  zugrundegelegt wurde.

Die Abhängigkeit der Schubkraft zur Drehzahl wurde mittels einer quadratischen Funktion in Matlab mit Hilfe der Funktion „polyfit“ angenähert. Das Ergebnis ist in Bild 3.17 zu erkennen. Die gemessenen Werte konnten durch die approximierete Funktion sehr gut abgebildet werden.

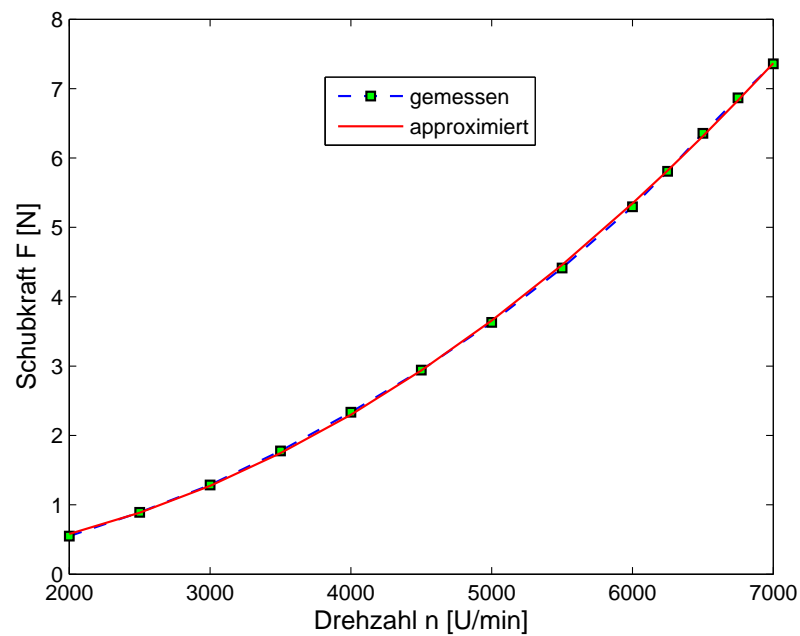
Die bestimmten Werte der Funktionsparameter sind in Gleichung 3.20 ersichtlich.

$$F(n) = a \cdot n^2 + b \cdot n + c = 1,6607 \cdot 10^{-7} \cdot n^2 - 1,3765 \cdot 10^{-4} \cdot n + 0,1924 \quad (3.20)$$

Aus der Gleichung kann die Drehzahl ermittelt werden, die jeder Motor aufbringen muss, um den Quadrocopter im Schwebeflug zu halten. Das Abfluggewicht des Quadrocopters wurde auf 1147 g bestimmt. Für den Schwebeflug muss dafür jeder Motor eine Schubkraft von 2,813 N bei einer Erdanziehungskraft von  $9,81\text{ m/s}^2$  aufbringen. Die erforderliche Drehzahl wurde mittels Matlab aus der Gleichung 3.20 auf 4408 U/min bestimmt, deren Wert sich jedoch auch grafisch aus Bild 3.17 ablesen ließe.

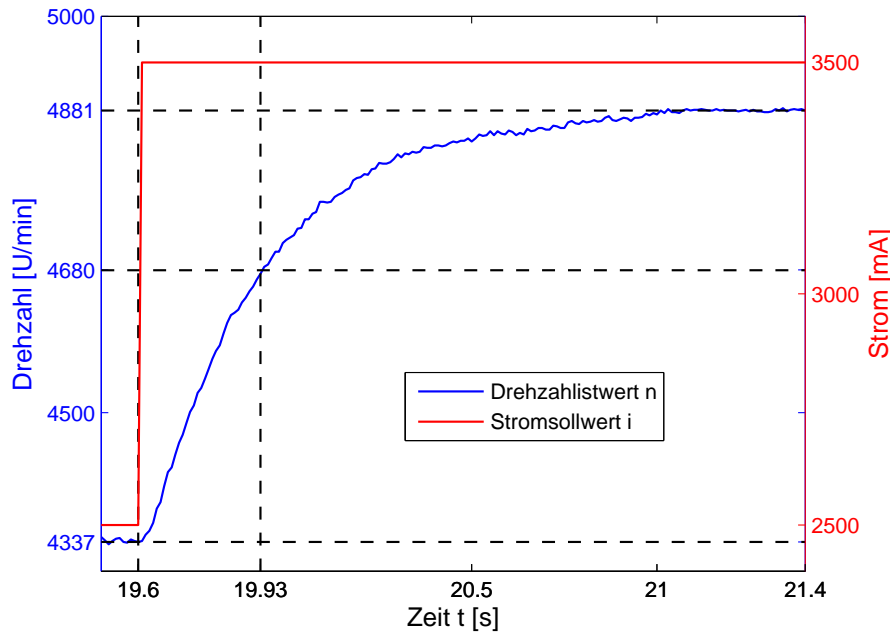
**Tabelle 3.3:** Schubmessung in Abhängigkeit der Drehzahl

Drehzahl [U/min]	Gewicht [g]	Schubkraft [N]	Strom [A]	Spannung [V]
2000	56	0,55	0,46	11,61
2500	91	0,89	0,72	11,56
3000	131	1,29	1,09	11,48
3500	181	1,78	1,56	11,42
4000	238	2,33	2,19	11,34
4500	300	2,94	3,05	11,25
5000	370	3,63	4,29	11,14
5500	450	4,41	5,50	11,08
6000	540	5,30	7,41	10,88
6250	592	5,81	-	-
6500	648	6,36	-	-
6750	700	6,87	-	-
7000	750	7,36	-	-

**Bild 3.17:** Approximierte Funktion der Schubkraft in Abhängigkeit von der Drehzahl

### 3.6.2 Verhalten der Rotordrehzahl auf eine Stromänderung

Bei dem ersten Flugversuch zeigte sich, dass der Stromverbrauch pro Motor im Schwebeflug bei etwa 2500 mA liegt. Um das Verhalten der Rotordrehzahl auf eine Stromänderung zu untersuchen, wird deshalb ein Stromsollwertsprung von 2500 mA auf 3500 mA durchgeführt.



**Bild 3.18:** Verhalten der Drehzahl auf einen Stromsollwertsprung

Bild 3.18 zeigt das Messergebnis der Untersuchung. Aus dieser Messung werden die Parameter  $K_{rpm}$  und  $T_{rpm}$  des PT1-Gliedes aus Kapitel 3.5.1 ermittelt. Dabei zeigt sich, dass die vereinfachte Annahme, die Verzögerung der Drehzahlannahme durch die Stromregelung zu vernachlässigen, ohne Bedenken angewandt werden kann. Der Verstärkungsfaktor  $K_{rpm}$  ergibt sich aus der Höhe des Stromsollwertsprungs  $\Delta i$  und der Drehzahländerung  $\Delta n$ .

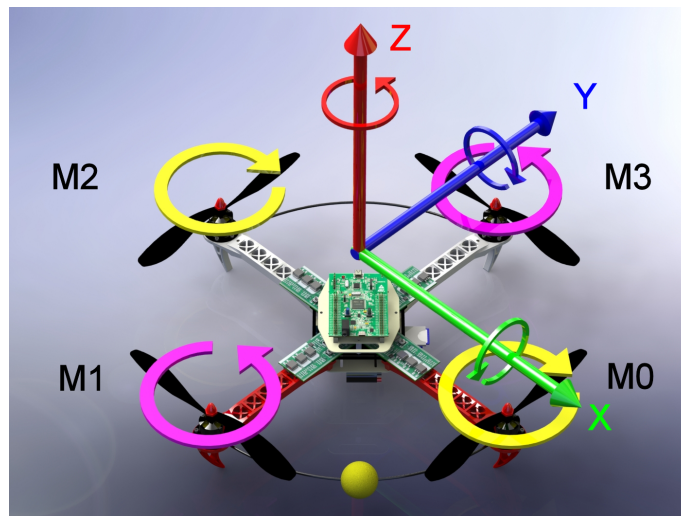
$$K_{rpm} = \frac{\Delta n}{\Delta i} = \frac{4881 \text{ U/min} - 4337 \text{ U/min}}{3500 \text{ mA} - 2500 \text{ mA}} = 0,544 \frac{\text{U/min}}{\text{mA}} \quad (3.21)$$

Die Zeitkonstante  $T_{rpm}$  des PT1-Gliedes wird aus der Zeitspanne ermittelt, die benötigt wird, um 63% des Drehzahlendwertes zu erreichen. Dies ist in Bild 3.18 bei einer Drehzahl von  $\Delta n \cdot 0,63 + 4337 \text{ U/min} = 4680 \text{ U/min}$  der Fall. [Vogl, 2012, S. 58-59]

$$T_{rpm} = 19,93 \text{ s} - 19,6 \text{ s} = 0,33 \text{ s} \quad (3.22)$$

### 3.6.3 Ermittlung der Hauptträgheitsmomente aus dem 3D-Modell

Für die Ermittlung der Hauptträgheitsmomente wird das konstruierte 3D-Modell herangezogen. Die Ausrichtung des Koordinatensystems des Quadropters ist in Bild 3.19 ersichtlich. Zu beachten ist in diesem Fall, dass dieses zum Zweck der besseren Darstellbarkeit auf der Z-Achse verschoben wurde. Der Ursprung der Z-Achse befindet sich im Schwerpunkt des Quadropters.



**Bild 3.19:** Koordinatensystemausrichtung Quadropters

In Solidworks werden die Trägheitsmomente für die Drehungen um die X-, Y- und Z-Achsen mit Hilfe des Tools Masseneigenschaften ermittelt. Dabei wird davon ausgegangen, dass die Trägheitsmomente für die Drehungen um die X- und Y-Achsen gleich sind.

Das Trägheitsmoment für die Drehung um X und Y bzw. Pitch und Roll wird mit  $J_1$  und die Drehung um Z bzw. Yaw mit  $J_2$  gekennzeichnet. Für  $J_1$  wurde ein Trägheitsmoment von  $4,96 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2$  und für  $J_2$  von  $9,66 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2$  ermittelt.



### 3.6.4 Ermittlung der Motordrehmomente

Die Motordrehmomente können, wie schon in Kapitel 3.5.4 erwähnt, aus den Motorströmen bestimmt werden. Die Gleichung 3.23 veranschaulicht nochmals deren Berechnung.

$$M_i = K \cdot \Phi \cdot I_A \quad (3.23)$$

Es müssen somit die Faktoren  $K$  und  $\Phi$  bestimmt werden. Der verwendete Motor hat eine Angabe der spezifischen Drehzahl von 935 kV. Diese gibt an, welche Leerlaufdrehzahl in U/min der Motor ohne Last bei einer Spannung von 1 V erreicht. Im Fall des idealen Leerlaufs ist die induzierte Spannung  $U_{gl}$  des Motors gleich der angelegten Ankerspannung  $U_{Al}$ . Mit dieser Kenntnis können die beiden Faktoren berechnet werden. Beachtet werden muss, dass die Leerlaufdrehzahl  $\Omega_l$  in rad/s angegeben wird.

$$U_{gl} = U_{Al} = K \cdot \Phi \cdot \Omega_l \quad (3.24)$$

$$\Rightarrow K \cdot \Phi = \frac{U_{Al}}{\Omega_l} = \frac{1 \text{ V} \cdot 60 \text{ s}}{935 \text{ min}^{-1} \cdot 2\pi} = 1,021 \cdot 10^{-2} \frac{\text{V}}{\text{s}} \quad (3.25)$$

## 3.7 Linearisierung des Modells

In der Zustandsraumdarstellung von Systemen können nur lineare Zusammenhänge dargestellt werden. Aus diesem Grund ist eine Systemlinearisierung notwendig. Das Modell wird dabei in einem festgelegten Arbeitspunkt linearisiert.

Der Arbeitspunkt wird bei dem Quadrocopter als Schwebeflug definiert, bei dem der Quadrocopter weder steigt noch sinkt und sich dessen Lage nicht verändert. Die notwendige Drehzahl pro Motor wurde schon in Kapitel 3.6.1 ermittelt und liegt bei 4408 U/min. Der Luftwiderstand  $F_w$  wird zu Null gewählt, da der Quadrocopter im Arbeitspunkt seine Lage nicht verändert.

In Matlab wurde ein Script erstellt, das eine automatische Generierung der Matrizen ermöglicht und die partiellen Ableitungen der Differentialgleichungen vornimmt. Alle relevanten Daten des Quadrocopters werden dabei mit einbezogen und können nach Belieben angepasst werden.

Die Matrizen des kontinuierlichen Mehrgrößensystems in Zustandsraumdarstellung sind in den Gleichungen 3.26, 3.27, 3.28 und 3.29 ersichtlich.

$$\mathbf{A} = \begin{pmatrix} -3,0303 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3,0303 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3,0303 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3,0303 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -0,0602 & 0 & 0,0602 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -0,0602 & 0 & 0,0602 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.26)$$

$$\mathbf{B} = \begin{pmatrix} 1,6485 & 0 & 0 & 0 \\ 0 & 1,6485 & 0 & 0 \\ 0 & 0 & 1,6485 & 0 \\ 0 & 0 & 0 & 1,6485 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0,0011 & -0,0011 & 0,0011 & -0,0011 \end{pmatrix} \quad (3.27)$$

$$\mathbf{C} = \begin{pmatrix} 0,25 & 0,25 & 0,25 & 0,25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.28)$$

$$\mathbf{D} = \mathbf{0} \quad (3.29)$$

In der Tabelle 3.4 werden die Eingänge, Zustände und Ausgänge inklusive deren Bedeutungen und Einheiten des Zustandsraummodells zusammengefasst.

**Tabelle 3.4:** Zusammenfassung der Eingangs-, Zustands- und Ausgangsgrößen des Zustandsraummodells

	<b>Bedeutung</b>	<b>Einheit</b>
<b>Eingänge</b>		
$u_1$	Motorstrom M0	mA
$u_2$	Motorstrom M1	mA
$u_3$	Motorstrom M2	mA
$u_4$	Motorstrom M3	mA
<b>Zustände</b>		
$x_1$	Motordrehzahl M0	U/min
$x_2$	Motordrehzahl M1	U/min
$x_3$	Motordrehzahl M2	U/min
$x_4$	Motordrehzahl M3	U/min
$x_5$	Winkel von Pitch	rad
$x_6$	Winkelgeschwindigkeit von Pitch	rad/s
$x_7$	Winkel von Roll	rad
$x_8$	Winkelgeschwindigkeit von Roll	rad/s
$x_9$	Winkel von Yaw	rad
$x_{10}$	Winkelgeschwindigkeit von Yaw	rad/s
<b>Ausgänge</b>		
$y_1$	Mittelwert aller vier Motordrehzahlen	U/min
$y_2$	Winkel von Pitch	rad
$y_3$	Winkel von Roll	rad
$y_4$	Winkel von Yaw	rad

### 3.7.1 Eigenwerte

Die Eigenwerte des Systems liegen in sechsfacher Anzahl bei Null und in vierfacher bei  $-3,0303$ .

Das System ist somit nicht asymptotisch stabil, da folgende Bedingung nicht erfüllt ist:

$$Re\{\lambda_i\} < 0 \quad , \quad \forall \in \{1, 2, \dots, n\} \quad (3.30)$$

Zudem kann das System nicht als stabil bezeichnet werden, da die Matrix  $\mathbf{A}$  nicht diagonalähnlich ist.

### 3.7.2 Steuer- und Beobachtbarkeit

Um das System überhaupt mittels einer Zustandsregelung stabilisieren zu können, muss dieses vollständig steuerbar sein. Die Steuerbarkeitsmatrix weist den vollen Rang auf und erfüllt somit dieses Kriterium.

Das System ist zudem vollständig beobachtbar, da die Beobachtbarkeitsmatrix vollen Rang aufweist. Dadurch ist es möglich, eventuell nicht messbare Zustände mit einem Beobachter zu schätzen.

Die beiden Kriterien für Steuer- und Beobachtbarkeit wurden in Matlab mittels der Befehle „ctrb“ und „obsv“ sowie „rank“ überprüft, da eine händische Rechnung aufgrund der Matrizengrößen kaum möglich bzw. sehr aufwändig ist.

## 4 LQR-Mehrgrößenregelung

In diesem Kapitel findet die Auslegung der Mehrgrößenregelung nach dem LQR-Verfahren statt. Anschließend wird eine Methodik entwickelt, vorhandene Störgrößen zu kompensieren.

### 4.1 Entwurf der LQR-Regelung

Für den Entwurf des Zustandsreglers nach dem LQR-Verfahren müssen die Wichtungsmatrizen  $\mathbf{Q}$  und  $\mathbf{R}$  bestimmt werden. Die Riccati-Gleichung wird mittels Matlab gelöst.

Für die Wichtungsmatrix  $\mathbf{R}$  wird eine 4x4-Einheitsmatrix gewählt. Die Einstellung der Einzelstrecken erfolgt anschließend anhand der Matrix  $\mathbf{Q}$ , bei der lediglich die Diagonalelemente bewertet werden. Eine Bewertung der Zustände  $x_6$ ,  $x_8$  und  $x_{10}$ , die den Drehraten von Pitch, Roll und Yaw entsprechen, ist nicht notwendig. Das LQR-Verfahren berücksichtigt diese Zustände in der Zustandsrückführung automatisch, da das System sonst nicht zu stabilisieren ist.

Die Matrix  $\mathbf{R}$  lautet somit wie folgt:

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

Die Matrix  $\mathbf{Q}$  wurde anhand von Sprungantworten für alle zu regelnden Größen empirisch eingestellt und anschließend am realen System getestet. Es wurde folgende Matrix eingestellt:

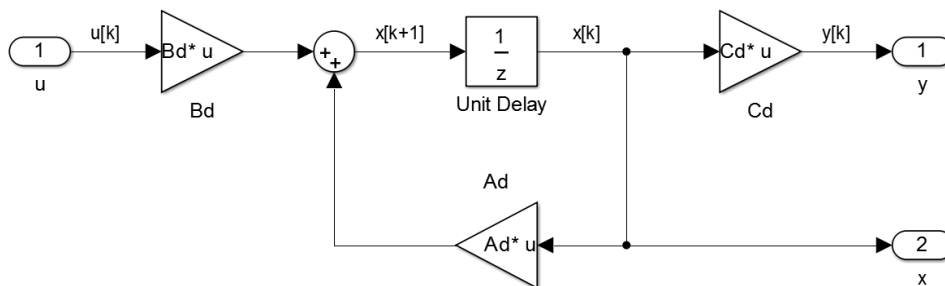
$$\mathbf{Q} = \begin{pmatrix} 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 \cdot 10^8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 \cdot 10^8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \cdot 10^8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.2)$$

Mit den ersten vier Diagonalelementen in  $\mathbf{Q}$  werden die Drehzahlregelungen der vier Motoren beeinflusst. Diese werden alle gleich gewichtet, um ein gleiches Verhalten aller Motoren zu erhalten. Gleiches gilt für die Zustände Pitch und Roll, die beide ein identisches Verhalten aufweisen sollen. Der Zustand für Yaw wurde unabhängig von allen anderen Größen parametrisiert. Die Zustände für Pitch, Roll und Yaw werden derart hoch gewichtet, da diese in der Einheit rad vorliegen und über die Rückführmatrix  $\mathbf{K}$  auf die Systemeingänge geführt werden, die in der Einheit mA vorgegeben werden.

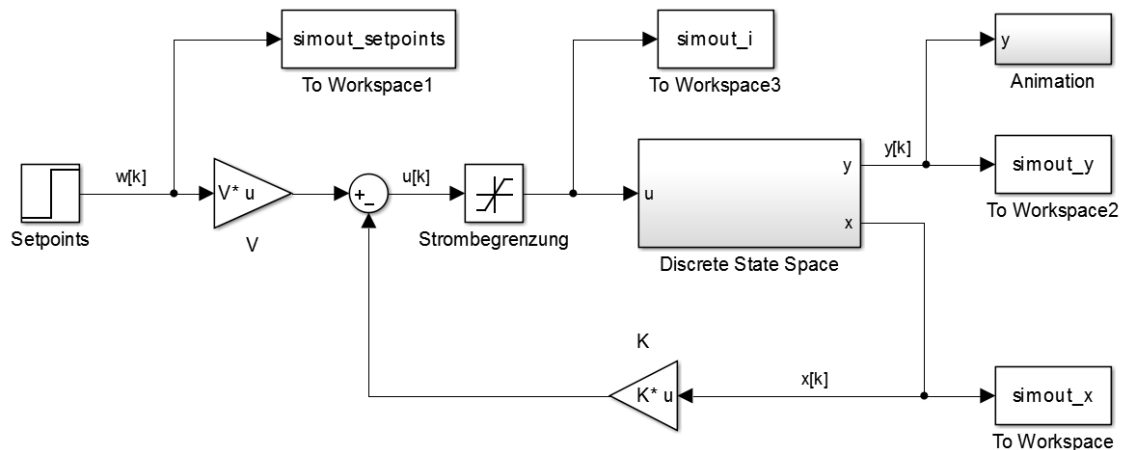
Über den Matlab-Befehl „lqr“ wird die Rückführmatrix  $\mathbf{K}$  am kontinuierlichen System berechnet.

Um am System eine Sollwertfolge ohne bleibende Regelabweichungen zu erreichen, wird zudem ein Vorfilter berechnet. Die Berechnung wird nach folgender Berechnungsvorschrift durchgeführt:

$$\mathbf{V} = -\left(\mathbf{C}(\mathbf{A} - \mathbf{BK})^{-1} \mathbf{B}\right)^{-1} \quad (4.3)$$

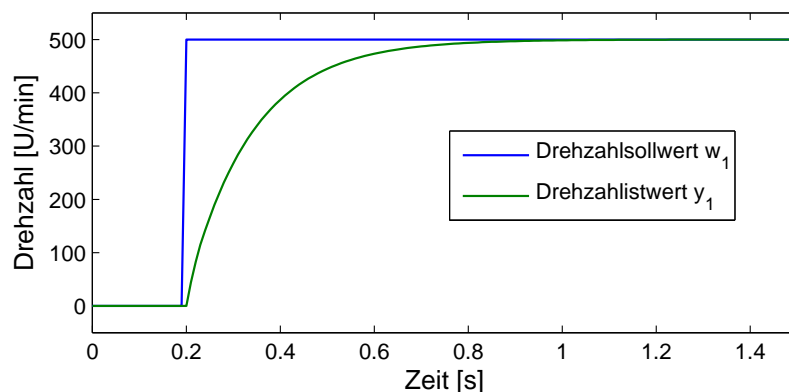


**Bild 4.1:** Simulinkmodell Subsystem Discrete State Space



**Bild 4.2:** Simulinkmodell zur Simulation der LQR-Regelung

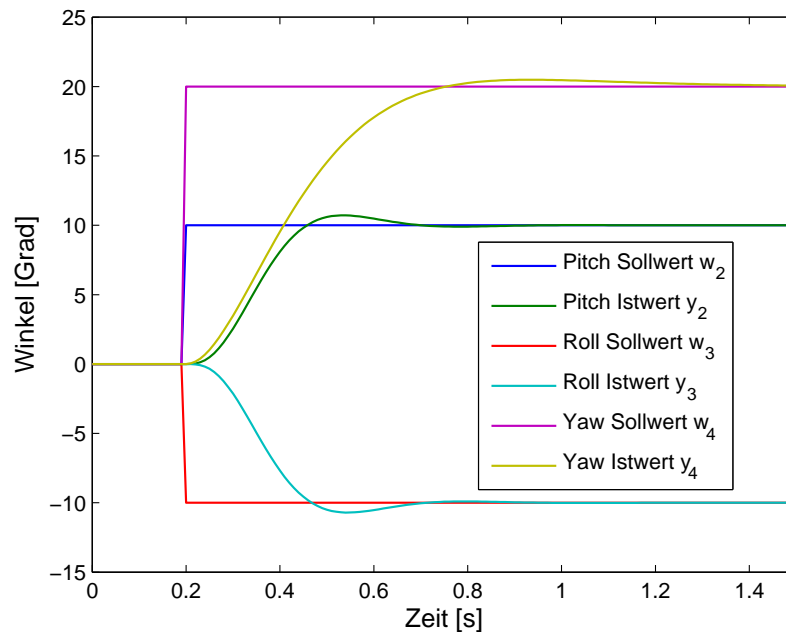
Für die Simulation der Sprungantworten des Systems wurde ein Matlab-Simulinkmodell entworfen, welches in Bild 4.2 zu erkennen ist. Das Subsystem „Discrete State Space“ wird in Bild 4.1 gezeigt. Die D-Matrix wurde im Subsystem vernachlässigt, da diese aus der Modellbildung gleich Null ist.



**Bild 4.3:** Simulierte Sprungantwort der Drehzahlen

Es ergeben sich mit der Rückführmatrix  $\mathbf{K}$  und dem Vorfilter  $\mathbf{V}$  die in Bild 4.3 und Bild 4.4 simulierten Sprungantworten für die zu regelnden Größen.

Ein Beobachterentwurf ist nicht nötig, da alle Zustände im Quadropter gemessen werden können. Der verbaute Lagesensor misst sowohl die Drehraten von Pitch, Roll und Yaw, als auch deren Winkel. Die Drehzahlen der vier Rotoren können von den Motorreglern ausgelesen werden, sodass alle zehn Zustände auch real zur Verfügung stehen.



**Bild 4.4:** Simulierte Sprungantwort Pitch, Roll, Yaw

Ein Problem der Zustandsregelung ist, dass Störgrößen nicht komplett ausgeglichen werden können. Dies ist vom Verständnis her mit einem P-Regler an einem SISO-System vergleichbar. Durch eine härtere Einstellung der Reglerparameter können zwar Störgrößeneinflüsse minimiert werden, dies ist allerdings nur bis zu einer gewissen Grenze möglich. Unendlich hohe Reglereinstellungen sind beispielsweise nur in der Theorie möglich, scheitern jedoch in der Praxis.

Gerade die Strecken Pitch und Roll sind sehr von Störungen betroffen. Zu nennen sind hier Wind und Turbulenzen, verursacht durch die eigenen Rotoren. Die Störungen, die bei der Strecke Yaw auftreten, sind nahezu nicht erkennbar.

In Bild 4.5 ist ein Ausschnitt aus einer Aufnahme eines realen Fluges mit dem Quadrocopter zu sehen. Die gleichen Sollwerte wurden zudem am geregelten diskreten Modell in Matlab simuliert, um einen Vergleich zwischen Realität und der Simulation zu erhalten. Es wurde zwar bei nahezu keinerlei Wind geflogen, dennoch sind die auftretenden Störungen gut erkennbar. Die rote Kennlinie liegt permanent etwa zehn Grad unter der simulierten grünen. Zudem weist das reale Verhalten der Strecke Pitch eine etwas größere Verzögerung als das der simulierten auf. Dies lässt sich durch den in der Realität vorhandenen Luftwiderstand erklären, der bei der Modellbildung aufgrund der Linearisierung nicht berücksichtigt wird (siehe Kapitel 3.7).



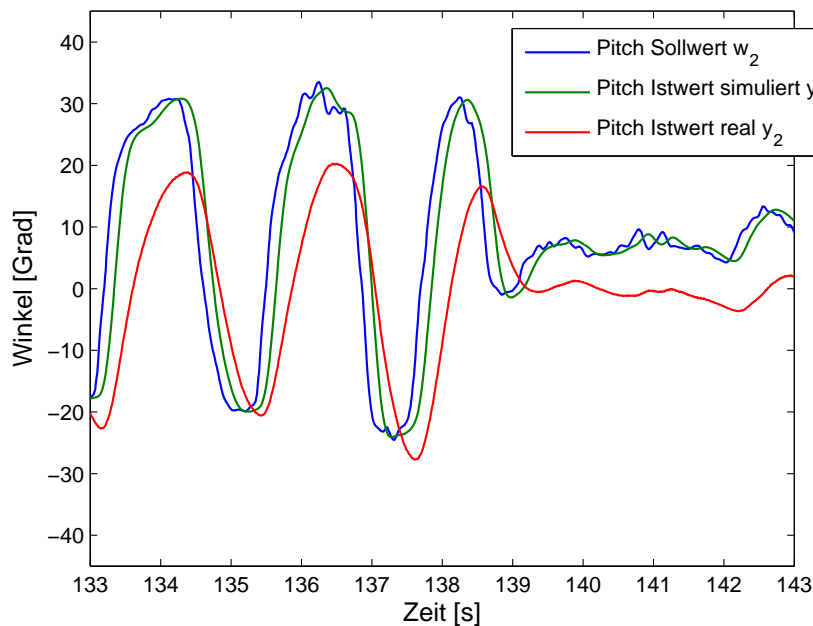


Bild 4.5: LQR-Regelung Soll-, Istwertvergleich

## 4.2 Störgrößenkompensation

Die auftretenden Störungen bei den Strecken Pitch und Roll sollen durch eine eigens entwickelte Herangehensweise beseitigt werden. Dazu soll intern im Mikrocontroller ein geregeltes Modell des Quadropters mitlaufen, das die gleichen Sollwerte erhält, wie auch der reale Quadropters.

Zu erkennen ist diese Idee in Bild 4.6 in diskreter Form. Der zustandsregelte Quadropters erhält im Idealfall die gleichen Stellwerte  $w[k]$  wie das intern mitlaufende simulierte geregelte Modell, solange keine Störgrößen auftreten und sich der Quadropters und das Simulationsmodell exakt gleich verhalten. Treten Abweichungen  $e[k]$  zwischen realem Modell  $y_r[k]$  und simuliertem  $y_m[k]$  auf, werden diese über einen diskret realisierten Integrator mit einstellbarer Verstärkung  $K$  aufintegriert. Der Ausgang des Integrators  $w_e[k]$  wird anschließend additiv dem realen Quadropters als zusätzlicher Sollwert hinzugefügt.

Der Vorteil dieser Art der Störgrößenbeseitigung besteht darin, dass eine Integration nur dann stattfindet, wenn sich die Ausgangswerte  $y_r[k]$  und  $y_m[k]$  unterscheiden. Entspricht das Verhalten der Simulation dem des Realen, ändert sich im Idealfall durch diese Störgrößenbeseitigung nichts an der Dynamik der Regelung.

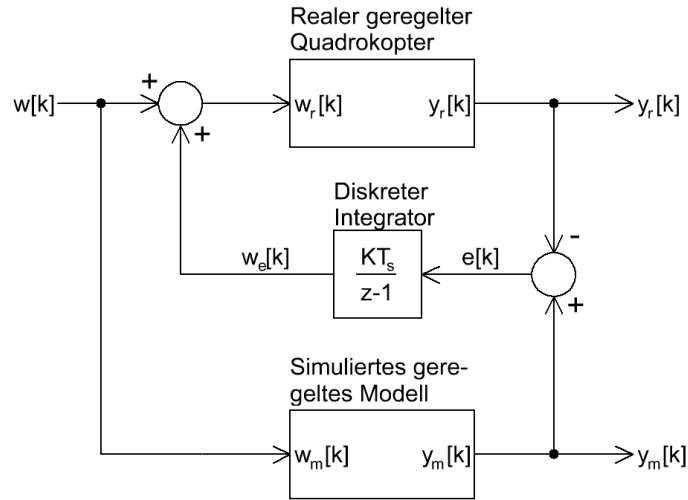


Bild 4.6: Störgrößenkompensation

Voraussetzung dafür ist jedoch, dass das Verhalten der Simulation sehr genau dem der Realität entspricht. Aus diesem Grund wird der Luftwiderstand  $F_w$  für das intern mitlaufende Modell mit einbezogen. Es handelt sich in diesem Fall um einen empirisch ermittelten mittleren Luftwiderstandswert, der auf  $F_w = 0,15$  bestimmt wurde. Für die Ermittlung wurden mehrere Flugtests herangezogen und mit der Simulation verglichen. Die Differentialgleichungen werden somit mit diesem Luftwiderstandswert für die Zustandsraumdarstellung neu linearisiert und diskretisiert. Die diskreten Zustandsraummatrizen mit dem eingeführten Luftwiderstand  $F_w$  werden mit den Indizes  $dF_w$  gekennzeichnet.

Es werden folgende Matrizen definiert:

$$\mathbf{A}_m = \mathbf{A}_{dF_w} - \mathbf{B}_{dF_w} \mathbf{K} \quad (4.4)$$

$$\mathbf{B}_m = \mathbf{B}_{dF_w} \mathbf{V} \quad (4.5)$$

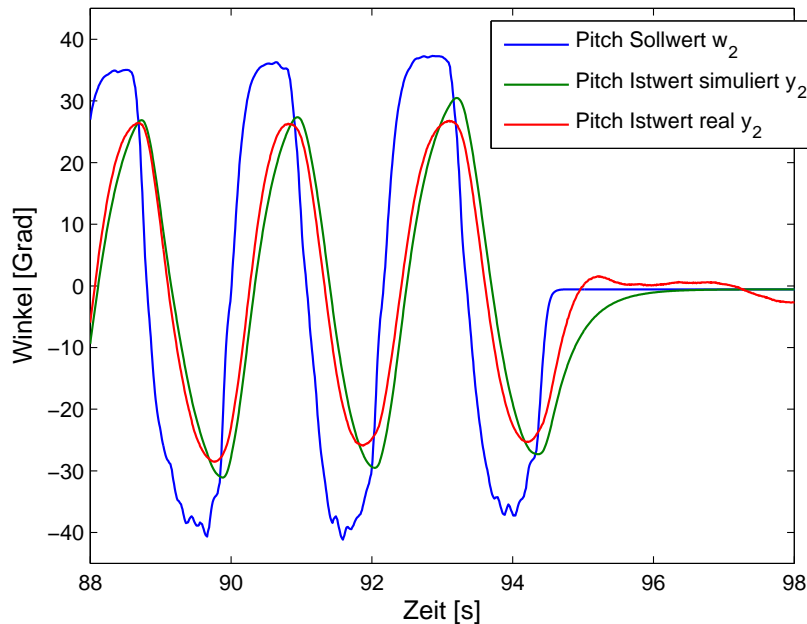
$$\mathbf{C}_m = \mathbf{C} \quad (4.6)$$

Das simulierte geregelte System hat damit folgendes Verhalten:

$$\mathbf{x}_m[k+1] = \mathbf{A}_m \mathbf{x}_m[k] + \mathbf{B}_m \mathbf{w}[k] \quad (4.7)$$

$$\mathbf{y}_m[k] = \mathbf{C}_m \mathbf{x}_m[k] \quad (4.8)$$

Die Matrizen  $\mathbf{A}_m$ ,  $\mathbf{B}_m$  und  $\mathbf{C}_m$  des Diskreten Zustandsraummodells werden mittels Matlab berechnet. Der Integrator in Bild 4.6 integriert nur die Differenzen von Pitch und Roll, da eine Korrektur der anderen Ausgangswerte nicht stattfinden soll. Die Verstärkung des Integrators wurde auf drei eingestellt. Es wird zudem eine Integrationsgrenze für die Stellwerte des Integrators  $w_e[k]$  auf  $\pm 10$  Grad vorgenommen.



**Bild 4.7:** LQR-Regelung Soll-, Istwertvergleich mit Störgrößenkompensation

Das Verhalten der Regelung von Pitch und Roll konnte dabei mittels dieser Methode derart verbessert werden, dass bei wenig Wind keine manuellen Korrekturen an der Fernsteuerung notwendig sind, um den Quadrocopter im Schwebeflug zu halten. Eine Beeinflussung der Dynamik der Regelung ist nicht zu beobachten. Die Flugdaten und der Vergleich zur Simulation sind in Bild 4.7 zu erkennen. Beide Aufzeichnungen, die in Bild 4.5 und Bild 4.7 zu sehen sind, wurden am gleichen Tag innerhalb einer Stunde bei gleichbleibenden Wetter- bzw. Windverhältnissen aufgenommen. Dadurch ist ein direkter Vergleich zwischen diesen Aufnahmen möglich.

## 5 Dezentrale Mehrgrößenregelung

In diesem Kapitel erfolgt der Entwurf einer dezentralen Regelung der Mehrgrößenregelstrecke. Dazu muss vorerst ein Entkopplungsglied entworfen werden. Anschließend werden Regelungen für die entstehenden entkoppelten Einzelstrecken entworfen und anhand von Einheitssprungantworten simuliert. Die Dynamiken der geregelten Strecken sollen dabei in etwa denen der LQR-Regelung entsprechen.

### 5.1 Entwurf des Entkopplungsgliedes

Entsprechend der Beschreibung in Kapitel 2.4.2 wird für die Regelung mit mehreren einzelnen Reglern zunächst ein Entkopplungsglied entworfen.

Die Berechnungsvorschrift wird nachfolgend nochmals aufgezeigt:

$$\mathbf{L}(s) = \mathbf{G}(s)^{-1} \text{diag} \tilde{\mathbf{G}}_i(s) \quad (5.1)$$

Die Übertragungsfunktionsmatrix  $\mathbf{G}(s)$  wurde aus dem kontinuierlichen Zustandsraummodell ermittelt:

$$\mathbf{G}(s) = \begin{pmatrix} \frac{0,4121}{s+3,03} & \frac{0,4121}{s+3,03} & \frac{0,4121}{s+3,03} & \frac{0,4121}{s+3,03} \\ \frac{-0,0992}{s^3+3,03s^2} & 0 & \frac{0,0992}{s^3+3,03s^2} & 0 \\ 0 & \frac{-0,0992}{s^3+3,03s^2} & 0 & \frac{0,0992}{s^3+3,03s^2} \\ \frac{0,001057}{s^2} & \frac{-0,001057}{s^2} & \frac{0,001057}{s^2} & \frac{-0,001057}{s^2} \end{pmatrix} \quad (5.2)$$

Die Inverse von  $\mathbf{G}(s)$  lautet:

$$\mathbf{G}(s)^{-1} = \begin{pmatrix} 0,6066s + 1,838 & -5,04s^3 - 15,27s^2 & 0 & 236,5s^2 \\ 0,6066s + 1,838 & 0 & -5,04s^3 - 15,27s^2 & -236,5s^2 \\ 0,6066s + 1,838 & 5,04s^3 + 15,27s^2 & 0 & 236,5s^2 \\ 0,6066s + 1,838 & 0 & 5,04s^3 + 15,27s^2 & -236,5s^2 \end{pmatrix} \quad (5.3)$$

Die Matrix  $\text{diag } \tilde{\mathbf{G}}_i(s)$  wurde wie folgt ausgelegt:

$$\text{diag } \tilde{\mathbf{G}}_i(s) = \begin{pmatrix} \frac{1}{s+1} & 0 & 0 & 0 \\ 0 & \frac{1}{s^3+s^2} & 0 & 0 \\ 0 & 0 & \frac{1}{s^3+s^2} & 0 \\ 0 & 0 & 0 & \frac{1}{s^2} \end{pmatrix} \quad (5.4)$$

Dies führt zur dynamischen Entkopplungsmatrix  $\mathbf{L}(s)$ :

$$\mathbf{L}(s) = \begin{pmatrix} \frac{0,6066s+1,838}{s+1} & \frac{-5,04s-15,27}{s+1} & 0 & 236,5 \\ \frac{0,6066s+1,838}{s+1} & 0 & \frac{-5,04s-15,27}{s+1} & -236,5 \\ \frac{0,6066s+1,838}{s+1} & \frac{5,04s+15,27}{s+1} & 0 & 236,5 \\ \frac{0,6066s+1,838}{s+1} & 0 & \frac{5,04s+15,27}{s+1} & -236,5 \end{pmatrix} \quad (5.5)$$

Zur einfacheren Implementierung soll lediglich eine statische Entkopplung erfolgen. Aus diesem Grund wird die statische Verstärkung von  $\mathbf{L}(s)$  ermittelt:

$$\mathbf{L}_{stat} = \mathbf{L}(0) = \begin{pmatrix} 1,838 & -15,27 & 0 & 236,5 \\ 1,838 & 0 & -15,27 & -236,5 \\ 1,838 & 15,27 & 0 & 236,5 \\ 1,838 & 0 & 15,27 & -236,5 \end{pmatrix} \quad (5.6)$$

Mit dem statischen Entkopplungsglied ergibt sich nachfolgend aufgeführtes Systemverhalten:

$$\tilde{\mathbf{G}}(s) = \mathbf{G}(s) \cdot \mathbf{L}_{stat} = \begin{pmatrix} \frac{3,03}{s+3,03} & 0 & 0 & 0 \\ 0 & \frac{3,03}{s^2(s+3,03)} & 0 & 0 \\ 0 & 0 & \frac{3,03}{s^2(s+3,03)} & 0 \\ 0 & 0 & 0 & \frac{1}{s^2} \end{pmatrix} \quad (5.7)$$

In diesem Fall führt auch das statische Entkopplungsglied zu einer vollständigen Entkopplung des Systems. Dieser Sachverhalt kommt durch die Verwendung der vier Motorregler, die als Stromregler fungieren, zustande.

Die nach der Entkopplung entstandenen Einzelstrecken können nachfolgend einzeln betrachtet und deren Regelungen ausgelegt werden. Es wird dabei von quasikontinuierlichen Regelungen ausgegangen, da die Abtastzeiten des Lagesensors und der Rotordrehzahlen mit 10 ms deutlich unter den Streckenzeitkonstanten liegen.

## 5.2 Regelung durch mehrere Eingrößenregler

### 5.2.1 Matlab PID-Regelalgorithmus

Ein Ansatz der Regelung der Einzelstrecken besteht darin, den Regelalgorithmus aus dem Matlab Simulink PID-Block in gleicher Weise auf dem Mikrocontroller zu implementieren. Der PID-Block bietet eine einfache und schnelle Möglichkeit der Reglereinstellung mittels einer Autotune-Funktion, die folglich verwendet werden kann. Andererseits können für die Einstellung auch manuelle Verfahren wie die Polvorgabe angewandt werden.

Die kontinuierliche Übertragungsfunktion dieses PID-Reglers stellt sich wie folgt dar:

$$G_r(s) = \frac{u(s)}{e(s)} = P + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}} \quad (5.8)$$

Der Faktor  $N$  stellt ein Filter für den D-Anteil dar, das dafür sorgt, dass sprungartige Änderungen am Reglereingang  $e$  zu einer begrenzten Sprungantwort am Ausgang  $u$  des Reglers führen.

Dieses Reglergesetz kann auch in die allgemein üblichere Form eines PIDT1-Reglers umgeformt werden.

$$G_r(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_p}{T_n s} + \frac{K_p T_v s}{1 + T_1 s} \quad (5.9)$$

Mit  $P = K_p$ ,  $I = \frac{K_p}{T_n}$ ,  $D = K_d = K_p T_v$  und  $N = \frac{1}{T_1}$ . Soweit im Folgenden nicht anders erwähnt, wird die Filterzeitkonstante des D-Anteils auf  $T_1 = T_v/10$  eingestellt.

Diskretisiert wird die kontinuierliche Matlab-PID-Übertragungsfunktion über den Vorwärtsdifferenziationssatz mit  $s \hat{=} \frac{z-1}{T_s}$ :

$$G_r(z) = P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}} \quad (5.10)$$

Die Reglerauslegung kann für erste Tests über die Autotune-Funktion in Simulink erfolgen, bei der die Einzelstrecken jeweils aufgebaut werden. Ausgelegt werden sollen die Regler allerdings anhand von Polvorgaben, die durch Güteforderungen definiert werden.

### 5.2.2 Drehzahlregelstrecke

Für die Regelung der Drehzahlen der vier Rotoren soll ein PI-Regler eingesetzt werden. Auf einen D-Anteil wird verzichtet, da das Drehzahlsignal trotz der Filterung noch Schwankungen aufweist und Priorität auf die Regelung der Lage gelegt wird. Es wird eine quasikontinuierliche Regelung entworfen.

Die Drehzahlregelstrecke ist in  $\tilde{\mathbf{G}}(s)$  aus dem ersten Diagonalelement der Matrix abzulesen und lautet:

$$G_{rpm}(s) = \frac{3,03}{s + 3,03} = \frac{1}{\frac{1}{3,03}s + 1} = \frac{1}{T_{rpm}s + 1} \quad (5.11)$$

Mit der Streckenzeitkonstante  $T_{rpm} = 0,33$  s.

Ein PI-Regler hat folgende Übertragungsfunktion:

$$G_{pi}(s) = K_p + \frac{K_p}{T_n s} = \frac{K_p s + \frac{K_p}{T_n}}{s} \quad (5.12)$$

Die Zusammenschaltung von Regler und Strecke führt zu folgender Führungsübertragungsfunktion:

$$G_{wrpm}(s) = \frac{G_{rpm}(s) \cdot G_{pi}(s)}{1 + G_{rpm}(s) \cdot G_{pi}(s)} = \frac{K_p s + \frac{K_p}{T_n}}{T_{rpm} s^2 + (K_p + 1)s + \frac{K_p}{T_n}} \quad (5.13)$$

Die Auslegung des Reglers soll über eine Vorgabe der Pole des Systems  $G_w(s)$  erfolgen. Dazu werden die Pole des Systems zunächst allgemein berechnet.

$$0 = s^2 + \frac{K_p + 1}{T_{rpm}} s + \frac{K_p}{T_n T_{rpm}} \quad (5.14)$$

Diese Gleichung wird mit Hilfe der PQ-Formel gelöst:

$$s_{1,2} = -\frac{K_p + 1}{2T_{rpm}} \pm j \sqrt{\frac{K_p}{T_n T_{rpm}} - \left(\frac{K_p + 1}{2T_{rpm}}\right)^2} \quad (5.15)$$

Angestrebt wird eine automatische Berechnung der Reglerparameter  $K_p$  und  $T_n$  anhand der Angaben der Beruhigungszeit  $T_{5\%}$  und der Überschwingweite  $\Delta h$  des geregelten Systems  $G_{wrpm}(s)$ . So können geänderte Güteforderungen an den Regelkreis schnell und effizient übernommen werden.

$$T_{5\%} \approx \frac{3}{\delta_e} \Rightarrow \delta_e = \frac{3}{T_{5\%}} \quad (5.16)$$

$$\Delta h = \exp\left(\frac{-\delta_e}{\omega_e}\pi\right) \Rightarrow \omega_e = -\frac{\delta_e\pi}{\ln(\Delta h)} \quad (5.17)$$

Die Güteforderungen der Beruhigungszeit und der Überschwingweite führen zu folgenden Polen:

$$s_{1,2} = -\delta_e \pm j\omega_e \quad (5.18)$$

Es wird ein Koeffizientenvergleich des Real- und Imaginärteils durchgeführt, um die Reglerparameter zu ermitteln.

Realteil:

$$\frac{3}{T_{5\%}} = \frac{K_p + 1}{2T_{rpm}} \quad (5.19)$$

$$\Rightarrow K_p = \frac{6T_{rpm}}{T_{5\%}} - 1 \quad (5.20)$$

Imaginärteil:

$$\frac{\delta_e\pi}{\ln(\Delta h)} = \sqrt{\frac{K_p}{T_n T_{rpm}} - \left(\frac{K_p + 1}{2T_{rpm}}\right)^2} \quad (5.21)$$

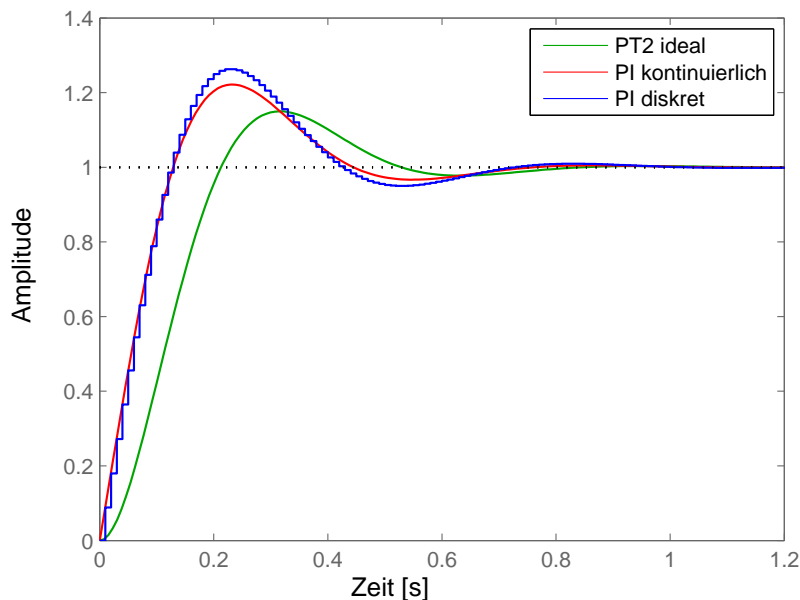
$$\left(\frac{\delta_e\pi}{\ln(\Delta h)}\right)^2 = \frac{K_p}{T_n T_{rpm}} - \left(\frac{K_p + 1}{2T_{rpm}}\right)^2 \quad (5.22)$$

$$\Rightarrow T_n = \frac{K_p}{T_{rpm} \left( \left(\frac{\delta_e\pi}{\ln(\Delta h)}\right)^2 + \left(\frac{K_p + 1}{2T_{rpm}}\right)^2 \right)} \quad (5.23)$$

Somit können nun durch die Vorgabe der Güteforderungen für die Regelung die Parameter des PI-Reglers automatisch berechnet werden.



Das geregelte System soll eine Beruhigungszeit von 0,5s und eine Überschwingweite von 15% besitzen. Es ergeben sich für den P-Anteil des Reglers  $K_p = 2,96$  und den I-Anteil  $T_n = 6,67 \cdot 10^{-3}$ .



**Bild 5.1:** Dezentrale Regelung der Drehzahlen (Sprungantwort)

In Bild 5.1 ist die Simulation der Sprungantwort auf das geregelte System im kontinuierlichen und diskreten Fall zu erkennen, wobei die Drehzahlstrecke mit einem Halteglied 0ter-Ordnung und einer Abtastzeit von 10 ms diskretisiert wurde. Es wurde außerdem das ideale, gewünschte Verhalten eines PT2-Gliedes, das sich aus den Güteforderungen ergibt, eingezeichnet. Die Überschwingweite von 15% wird zwar von den Reglern etwas überschritten, die Toleranz ist jedoch akzeptabel. Zudem können die Parameter nach Belieben angepasst werden.

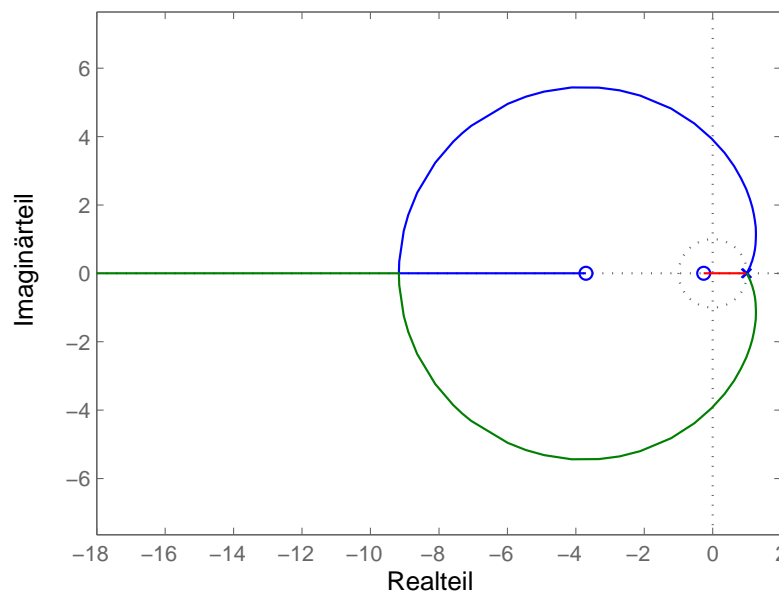
Am realen Modell funktionieren diese Reglereinstellungen einwandfrei und es zeigt sich eine sehr gute Drehzahlannahme.

### 5.2.3 Regelung von Pitch und Roll

Die Übertragungsfunktionen für Pitch und Roll ergeben sich aus dem zweiten und dritten Diagonalelement von  $\tilde{\mathbf{G}}(s)$  und zeigen beide das gleiche Verhalten.

$$G_{sp}(s) = \frac{3,03}{s^2(s + 3,03)} = \frac{1}{s^2(T_{rpm}s + 1)} \quad (5.24)$$

Die Regelung dieser Strecke  $G_{sp}(s)$  ( $sp$  = Strecke Pitch) stellt sich als sehr problematisch heraus, da die Strecke einen Doppelintegrator beinhaltet und zudem noch durch ein PT1-Glied verzögert wird. Auf einen I-Anteil in der Regelung sollte in diesem Fall verzichtet werden, um die Schwingneigung des Systems nicht noch zusätzlich zu fördern.



**Bild 5.2:** Wurzelortskurve der diskretisierten Strecke Pitch, geregelt durch einen P-Regler

Mit Hilfe einer Wurzelortskurve am diskretisierten System  $G_{sp}(z)$  mittels eines Halteglieds 0ter-Ordnung und einer Abtastzeit von 10 ms und geregelt durch einen P-Regler kann gezeigt werden, dass das System durch einen reinen P-Regler nicht stabilisierbar ist. In Bild 5.2 ist zu erkennen, dass die Pole für alle Verstärkungen des P-Reglers keine Lage innerhalb des Einheitskreises erreichen.

Ein PD-Regler kann in der Theorie so ausgelegt werden, dass dieser das System stabilisiert. Dazu wird in erster Näherung das PT1-Verhalten der Strecke vernachlässigt, sodass als Strecke ein Doppelintegrator geregelt wird.

Die Auslegung erfolgt wie bei der Drehzahlstrecke aus 5.2.2 mit dem Unterschied der Strecke als Doppelintegrator. Es wird für die Polzuweisung ein idealer PD-Regler angenommen.

$$G_s(s) = \frac{1}{s^2} \quad (5.25)$$

$$G_r(s) = K_p + K_d s \quad (5.26)$$

$$G_w(s) = \frac{K_p + K_d s}{s^2 + K_d s + K_p} \quad (5.27)$$

$$0 = s^2 + K_d s + K_p \quad (5.28)$$

$$s_{1,2} = -\frac{K_d}{2} \pm j\sqrt{K_p - \frac{K_d^2}{4}} \quad (5.29)$$

Der Koeffizientenvergleich führt zu folgenden Berechnungsvorschriften für die Reglerparameter:

Realteil:

$$\frac{3}{T_{5\%}} = \frac{K_d}{2} \Rightarrow K_d = \frac{6}{T_{5\%}} \quad (5.30)$$

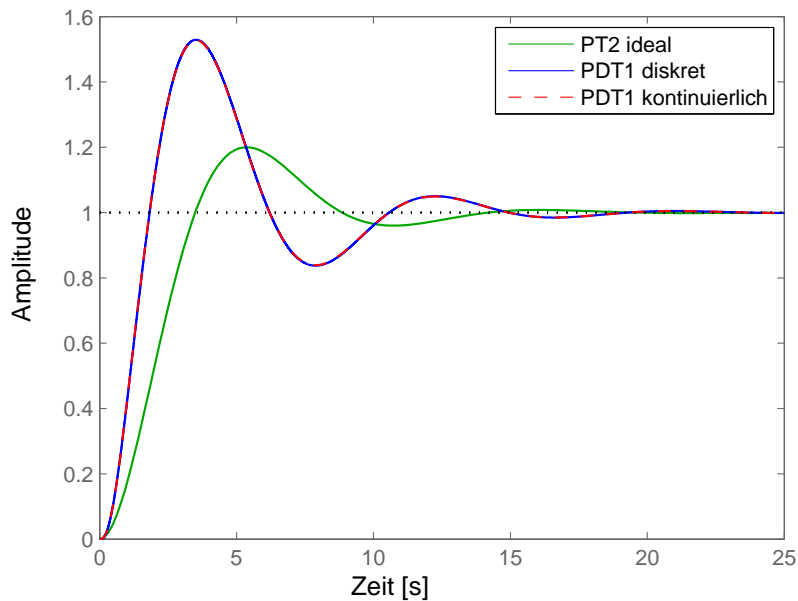
Imaginärteil:

$$\left( \frac{\delta_e \pi}{\ln(\Delta h)} \right)^2 = K_p - \frac{K_d^2}{4} \quad (5.31)$$

$$\Rightarrow K_p = \left( \frac{\delta_e \pi}{\ln(\Delta h)} \right)^2 + \frac{K_d^2}{4} \quad (5.32)$$

Leider lassen sich für den PD-Regler keine akzeptablen Reglereinstellungen finden. Lediglich Beruhigungszeiten von etwa 10 s und größer sind möglich. Das System lässt sich andernfalls nicht stabilisieren und beginnt zu schwingen.

Bild 5.3 zeigt das Verhalten der simulierten geregelten Strecke  $G_{sp}(s)$  mit einem PDT1-Regler in kontinuierlicher und diskretisierter Form und das ideal geforderte Verhalten (PT2 ideal). Als Güteforderung wurde eine Beruhigungszeit von 10 s und eine Überschwingweite von 20 % angenommen.



**Bild 5.3:** Dezentrale Regelung Pitch (Sprungantwort)

Am realen System führen derartig langsame Beruhigungszeiten zu keiner Stabilisierung. Auch ein empirisches Einstellen des Reglers sowie die Zuhilfenahme der Matlab-Autotune-Funktion führen zu keiner Stabilität.

Die Regelung dieses Systems wird deshalb als gescheitert angesehen und nicht weiter betrachtet. Auf die Auslegung der Strecke Yaw wird verzichtet, da der Test nur an einem System getestet werden kann, das uneingeschränkt flugfähig ist.

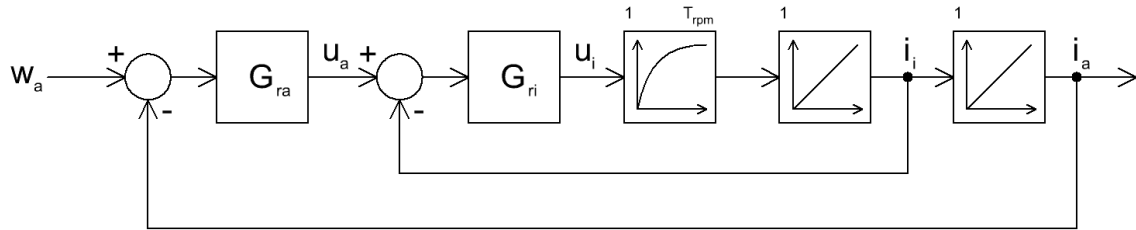
### 5.3 Regelung durch kaskadierte Eingrößenregler

Um die sehr problematische Regelung der Strecken Pitch und Roll zu verbessern, kann eine kaskadierte Regelung an dem entkoppelten System erfolgen. Durch die statische Entkopplung bleibt das Verhalten des Doppelintegrators wie in der Modellbildung in Kapitel 3.5.2 unverändert erhalten. Es kann somit ein innerer Regelkreis aufgebaut werden, der die Drehrate regelt. Ein äußerer Regelkreis regelt anschließend die Lage, wobei dessen Stellwerte dem unterlagerten Regelkreis übergeben werden.

Die Reglereinstellungen erfolgen bei einer Kaskadenregelung von innen nach außen, da die innere Regelstrecke eine höhere Dynamik aufweist.

Die Drehzahlregelung bleibt unangetastet und muss nicht neu ausgelegt werden.

### 5.3.1 Regelung von Pitch und Roll



**Bild 5.4:** Grundaufbau der kaskadierten Regelung

Zu sehen ist der Aufbau der kaskadierten Regelung von Pitch und Roll in Bild 5.4. Das Streckenverhalten für die Drehrate besteht aus einem PT1-Glied und einem Integrator, dessen Regelung durch den inneren Regler  $G_{ri}$  übernommen wird. Aus der Integration der Drehrate  $i_i$  ergibt sich die Lage  $i_a$ , die mit dem äußeren Regler  $G_{ra}$  geregelt wird.

#### Innerer Regelkreis

Für den inneren Regelkreis soll ein PDT1-Regler angewandt werden. Die Auslegung erfolgt auch in diesem Fall mit Hilfe der Polzuweisung über vorgegebene Gütekriterien.

Die innere Regelstrecke  $G_{spi}(s)$  ( $spi$  = Strecke Pitch innen) ergibt sich wie folgt:

$$G_{spi}(s) = \frac{1}{s(T_{rpm}s + 1)} \quad (5.33)$$

Mit einem idealen PD-Regler, wie schon aus Gleichung 5.26 bekannt, ergibt sich folgende Übertragungsfunktion für den inneren Regelkreis:

$$G_{wpi}(s) = \frac{K_d s + K_p}{s(T_{rpm}s + 1) + K_p + K_d s} \quad (5.34)$$

$$G_{wpi}(s) = \frac{K_d s + K_p}{T_{rpm}s^2 + (K_d + 1) + K_p} \quad (5.35)$$

Polstellen:

$$0 = s^2 + \frac{1 + K_d}{T_{rpm}}s + \frac{K_p}{T_{rpm}} \quad (5.36)$$

$$s_{1,2} = -\frac{1 + K_d}{2T_{rpm}} \pm j\sqrt{\frac{K_p}{T_{rpm}} - \left(\frac{1 + K_d}{2T_{rpm}}\right)^2} \quad (5.37)$$

Der Koeffizientenvergleich führt zu folgenden Berechnungsvorschriften:

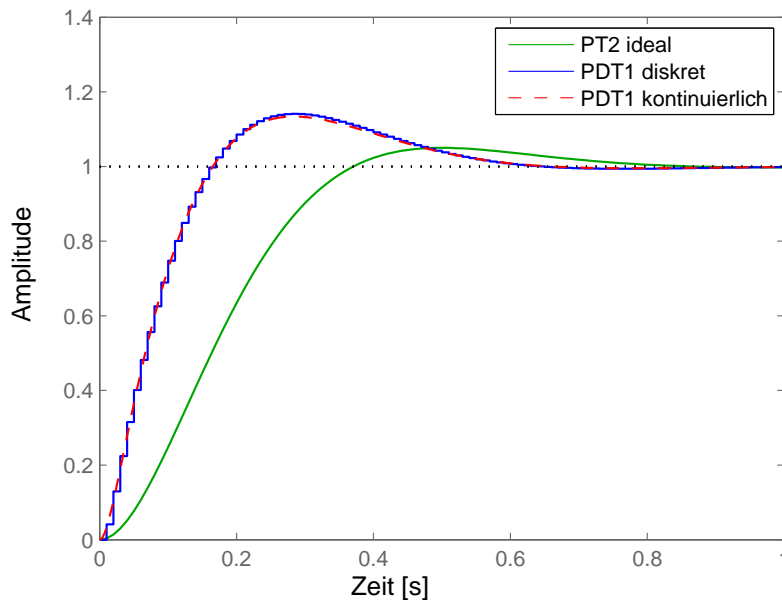
Realteil:

$$\frac{3}{T_{5\%}} = \frac{1 + K_d}{2T_{rpm}} \Rightarrow K_d = \frac{6T_{rpm}}{T_{5\%}} - 1 \quad (5.38)$$

Imaginärteil:

$$\left( \frac{\delta_e \pi}{\ln(\Delta h)} \right)^2 = \frac{K_p}{T_{rpm}} - \left( \frac{1 + K_d}{2T_{rpm}} \right)^2 \quad (5.39)$$

$$\Rightarrow K_p = \left( \left( \frac{\delta_e \pi}{\ln(\Delta h)} \right)^2 + \left( \frac{1 + K_d}{2T_{rpm}} \right)^2 \right) T_{rpm} \quad (5.40)$$



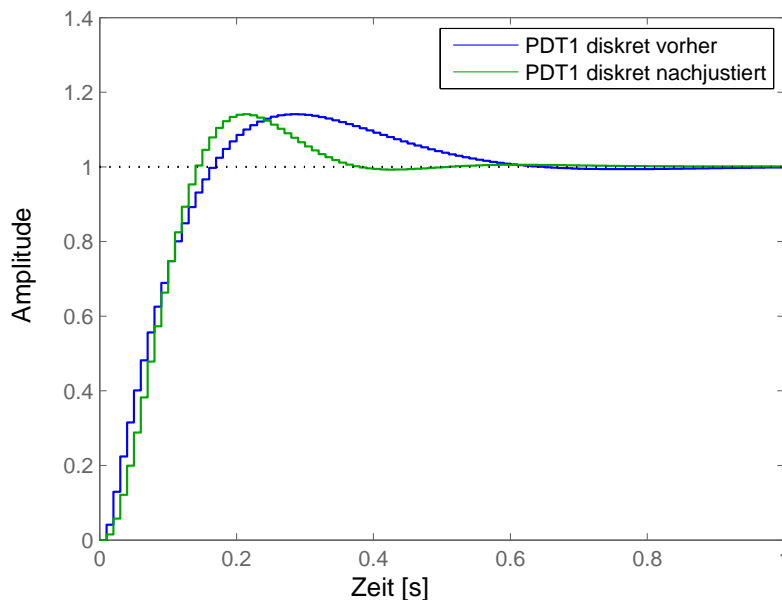
**Bild 5.5:** Dezentrale Regelung Pitch, innerer Regelkreis (Sprungantwort)

Angestrebt wird bei der inneren Regelung ein möglichst schnelles Einschwingen mit möglichst kleiner Überschwingweite, um eine gute Einstellbarkeit des äußeren Regelkreises zu gewährleisten. Als Gütekriterien wurden eine Einschwingzeit von 0,5 s und eine Überschwingweite von 5 % gefordert. Es ergeben sich für  $K_p = 24,95$  und  $K_d = 2,96$ . Das Ergebnis der Simulation ist in Bild 5.5 ersichtlich, wobei auch in diesem Fall das kontinuierliche, das mit 10 ms diskretisierte und das ideale Systemverhalten auf einen Eingangssprung dargestellt werden.

### Empirische Nachjustierung des inneren Regelkreises

Letztlich führt ein empirisches Nachstellen der Parameter zu einem noch geeigneteren Verhalten der Sprungantwort des inneren Regelkreises. Die Reglerauslegung über die Polzuweisung führt zu sehr großen Werten von  $K_p$ , die sich zwar als funktionsfähig am realen Modell herausstellen, jedoch eine erhöhte Gefahr zur Schwingneigung aufweisen.

Der Wert von  $K_p$  wurde auf 17 verringert,  $K_d$  auf 4,2 erhöht und die Filterzeitkonstante  $T_1$  von  $T_1 = T_v/10$  auf  $T_1 = T_v/5$  erhöht.



**Bild 5.6:** Dezentrale Regelung Pitch, innerer Regelkreis, nachjustiert (Sprungantwort)

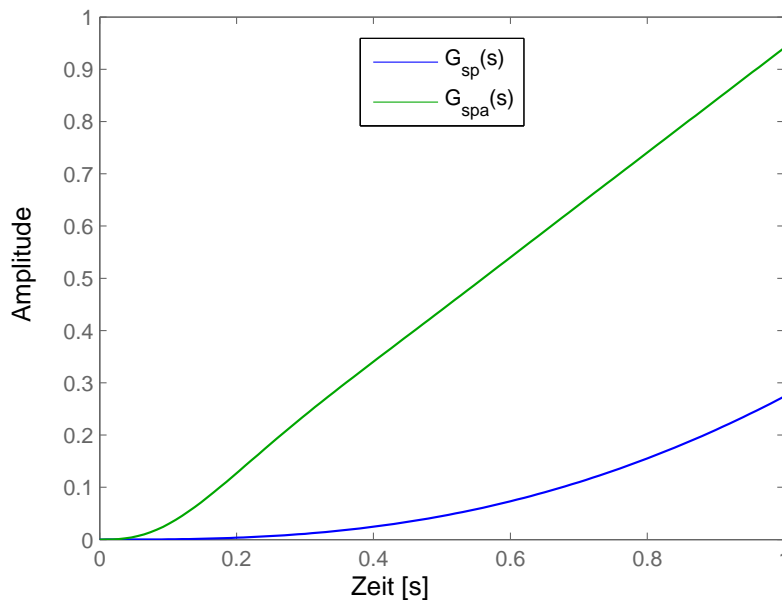
Mit diesen vorgenommenen Einstellungen ergibt sich das in Bild 5.6 dargestellte Verhalten. Es werden dabei das Verhalten der geregelten Strecke mit den diskretisierten PDT1-Reglern vor und nach der empirischen Nachstellung miteinander verglichen. Gut ersichtlich ist, dass trotz gleicher Überschwingweite nach der Justierung eine kleinere Einschwingzeit erreicht wird. Die Reglerauslegung über die Polzuweisung gibt jedoch einen guten Anhaltspunkt für die einzustellenden Parameter, um die gewünschte Dynamik zu erreichen.

## Äußerer Regelkreis

Das Übertragungsverhalten  $G_{spa}(s)$  ( $spa$  = Strecke Pitch außen) des äußeren Regelkreises ergibt sich aus der Reihenschaltung des inneren geregelten Systems und einem Integrator, wie es in Bild 5.4 veranschaulicht wurde.

Die Übertragungsfunktion der äußeren zu regelnden Strecke ergibt sich mit den Reglerparametern des inneren PDT1-Reglers zu:

$$G_{spa}(s) = \frac{309,1s + 1043}{s^4 + 23,27s^3 + 370,4s^2 + 1043s} \quad (5.41)$$



**Bild 5.7:** Sprungantwort der Strecke Pitch mit und ohne kaskadierter innerer Regelung

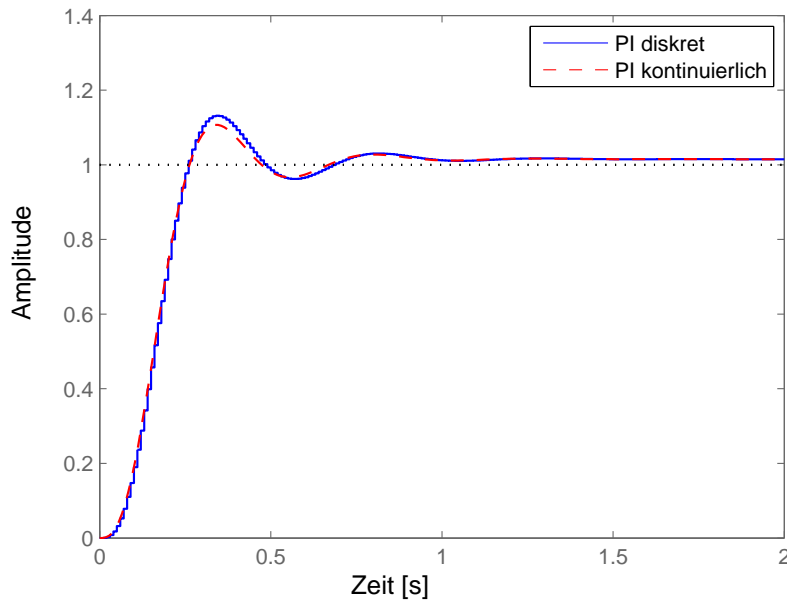
Bild 5.7 veranschaulicht die Sprungantwort des Systems  $G_{spa}(s)$  im Vergleich zum System  $G_{sp}(s)$  ohne inneren Kaskadenregler aus Gleichung 5.24. Es kann in erster Näherung von einem IT1-Verhalten ausgegangen werden. Deutlich zu erkennen ist, dass das Streckenverhalten durch den inneren Regler deutlich beschleunigt wird und zudem das Verhalten des Doppelintegrators entfällt. Dadurch lässt sich die Strecke sehr gut mit einem P- bzw. PI-Regler regeln.

Die Einstellung erfolgt in diesem Fall empirisch. Dazu wird zunächst ein P-Regler am realen System so eingestellt, bis das System beginnt zu schwingen. Der P-Anteil



wird von diesem Wert ausgehend wieder etwas reduziert. Anschließend wird ein kleiner I-Anteil hinzugefügt, um auftretende Störgrößen auszuregeln.

Es wurden durch diese empirische Einstellmethodik für den P-Anteil ein Wert von  $K_p = 6,5$  und für den I-Anteil ein Wert von  $K_i = 0,75$  ermittelt.



**Bild 5.8:** Äußere geregelte Strecke Pitch (Sprungantwort)

Bild 5.8 veranschaulicht die Simulation des Verhaltens der geregelten Strecke Pitch mit der kaskadierten Regelung. Die geringe Regelabweichung nach dem Einschwingvorgang nach etwa einer Sekunde ergibt sich durch den I-Anteil des äußeren Reglers, wird allerdings nach einiger Zeit kompensiert.

### 5.3.2 Regelung von Yaw

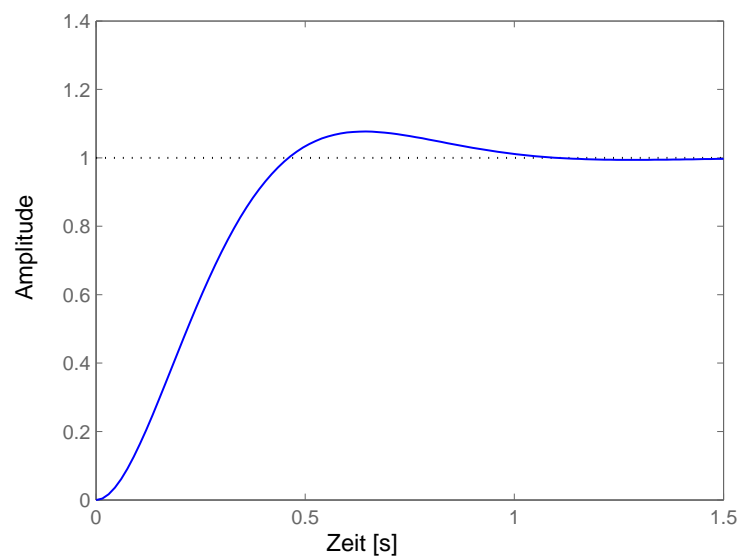
Auch die Regelung von Yaw soll kaskadiert erfolgen. Das Verhalten der Strecke Yaw ergibt sich aus dem letzten Diagonalelement von  $\tilde{\mathbf{G}}(s)$  und lautet:

$$G_{sy}(s) = \frac{1}{s^2} \quad (5.42)$$

Der Aufbau der kaskadierten Regelung dieser Strecke erfolgt, wie auch bei den Strecken Pitch und Roll, ersichtlich in Bild 5.4, mit dem Unterschied, dass das PT1-Glied in diesem Fall entfällt.

Die innere Strecke besteht somit aus einem reinen Integrator. Diese Strecke wird mittels eines P-Reglers geregelt. Auch die äußere Regelstrecke wird lediglich mit einem P-Regler geregelt. Dadurch ergibt sich eine sehr einfache Einstellbarkeit der Regler.

Beide Regler werden empirisch anhand von Matlab-Sprungantworten eingestellt. Es werden für den inneren Regler ein Wert von  $K_p = 8$  und den äußeren ein Wert von  $K_p = 5$  eingestellt.



**Bild 5.9:** Äußere geregelte Strecke Yaw (Sprungantwort)

Bild 5.9 zeigt die simulierte Sprungantwort des kaskadiert geregelten Systems in kontinuierlicher Form. Auf eine Darstellung des diskretisierten Systems wird in diesem Fall verzichtet.

## 6 Implementierung

Die Programmierung des STM32F4-Discoveryboards erfolgt in der Programmiersprache C. Als Programmierumgebung wird CoCoX CoIDE verwendet, die sehr empfehlenswert ist und bereits eine Unterstützung für die Programmierung des verwendeten Mikrocontrollers von STM beinhaltet. Das Programm bietet einen relativ einfachen Einstieg in die Programmierung dieser Art von Mikrocontrollern. [[CoIDE](#)]

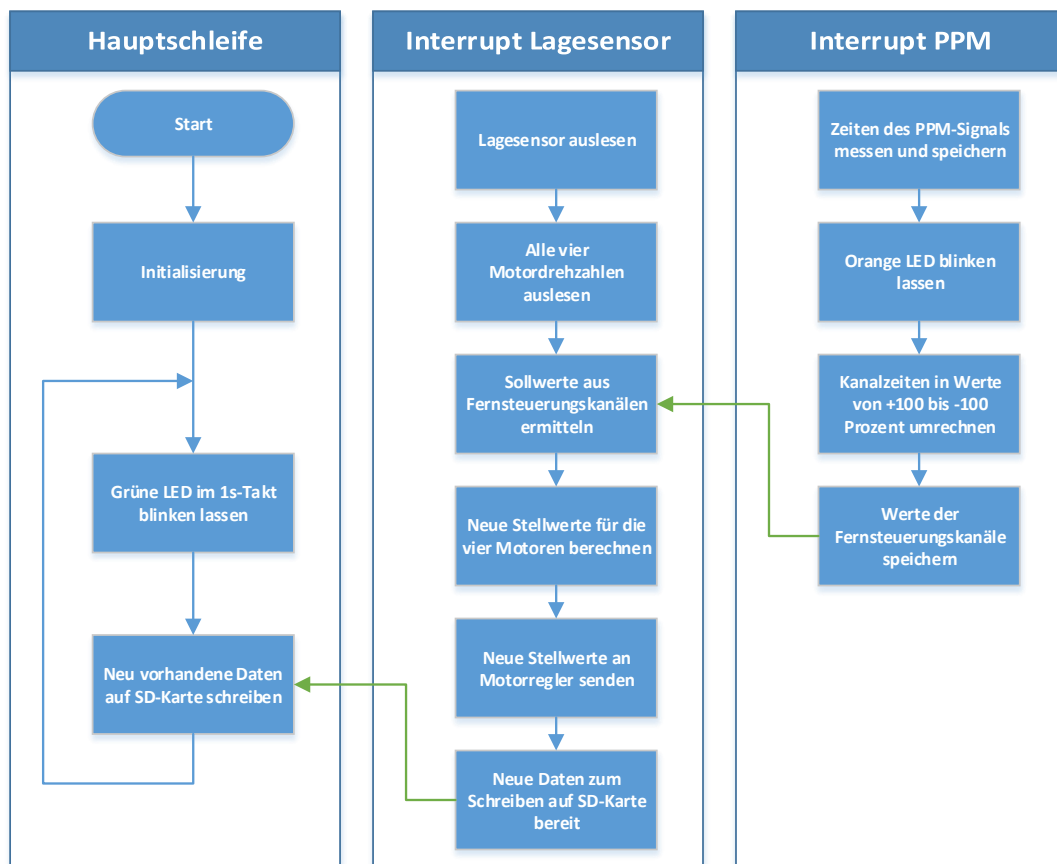
Die Implementierungen der verschiedenen Regelungsansätze unterscheiden sich lediglich durch eine andere Art der Stellgrößenberechnung für die vier Motorregler. Aus diesem Grund findet in diesem Kapitel eine gemeinsame Erläuterung der Implementierungen statt. Es wird dabei zunächst die Grundstruktur des Programms mit den verwendeten Interrupts aufgezeigt, um den Quellcode nachvollziehen zu können. Anschließend erfolgt die Erläuterung der Implementierungen der Zustandsregelung und der dezentralen Regelung.

Der Quellcode wird im Folgenden nicht näher erläutert. Dieser wurde ausreichend kommentiert und es wurden gut verständliche Variablennamen gewählt, sodass dieser aus der Kommentierung heraus verstanden werden kann.

### 6.1 Programmaufbau

Der in Bild [6.1](#) ersichtliche Programmablaufplan gibt einen groben Überblick über den grundsätzlichen Programmaufbau inklusive der Nutzung von Interrupts. Die grünen Pfeile veranschaulichen die Schnittstellen zwischen den Interrupts und der Hauptschleife.

Im PPM-Interrupt findet die Messung des PPM-Signals und dessen Auswertung statt. Eine auf dem Discoveryboard vorhandene orange LED signalisiert, dass ein PPM-Signal empfangen wird. Es findet eine kontinuierliche Überprüfung auf Korrektheit des PPM-Signals statt. Wird beispielsweise die Fernbedienung ausgeschaltet, werden sofort alle Motoren deaktiviert. Der PPM-Interrupt besitzt im Mikrocontroller eine höhere Priorität als der Interrupt des Lagesensors. Zudem werden die Interrupts preemptiv bearbeitet, sodass höher priorisierte Interrupts niedriger priorisierte unterbrechen können. Dies ist vor allem wichtig, da mittels des PPM-Interrupts die Signalzeiten des PPM-Signals



**Bild 6.1:** Schematischer Programmablaufplan

gemessen werden. Eine genaue Zeitmessung ist dabei nur möglich, wenn das Auslesen und Rücksetzen des verwendeten Timers zur Zeitmessung unmittelbar bei Auftreten von Signaländerungen erfolgt.

Der nieder priorere Interrupt des Lagesensors, zu finden in der Datei „mpu6050dmp6.c“, wird ausgeführt, sobald das MPU6050-IC über die Interruptleitung mitteilt, dass der FIFO neue Daten bereithält. Dies ist etwa alle 10 ms der Fall. Der FIFO wird anschließend im Interrupt ausgelesen. Zudem werden alle Istwerte des Systems, die für die Regelungen notwendig sind, ausgelesen. Dazu zählen außer der Lage des Quadropters die Drehzahlen aller vier Rotoren, die über die I2C-Schnittstelle der Motorregler ausgelesen werden. Anhand der Werte der Fernsteuerungskanäle, die im PPM-Interrupt ermittelt wurden, werden die neuen Sollwerte berechnet.

Über zwei Schalter an der Fernsteuerung kann gewählt werden, mit welcher Art der Regelung die neuen Stellwerte im Lagesensor-Interrupt berechnet werden sollen. Es können die normale Zustandsregelung, die Zustandsregelung mit Störgrößenkompensation oder die dezentrale kaskadierte Regelung ausgewählt werden. Je nach Auswahl erfolgt anhand der ermittelten Ist- und Sollwerte die Berechnung der neuen Stellwerte für die vier Motoren. Diese Stellwerte, welche den vier Motorströmen in mA entsprechen, werden nach der Berechnung über die I2C-Schnittstelle an die vier Motorregler übermittelt. Als letzter Schritt wird der Hauptschleife über eine Variable mitgeteilt, dass neue Daten für das Schreiben auf die SD-Karte vorhanden sind.

In der Hauptschleife erfolgt die Abarbeitung der nicht zeitkritischen Aufgaben. Die grüne LED auf dem Discoveryboard dient zur Signalisierung, dass die Hauptschleife noch bearbeitet wird und der Mikrocontroller nicht dauerhaft einen der Interrupts bearbeitet. Zudem erfolgt das Schreiben der Flugdaten auf die SD-Karte, sobald neue vorhanden sind.

## 6.2 LQR-Regelung

Für die LQR-Regelung wurden Strukturen und Funktionen geschrieben, die es ermöglichen, verschieden große Matrizen miteinander addieren, subtrahieren und multiplizieren zu können. Dies ist eine Grundvoraussetzung zur Berechnung der neuen Stellwerte und zur intern mitlaufenden Simulation des geregelten Modells zur Störgrößenkompensation.

In Matlab wurde ein Script erstellt, das die automatische Konvertierung der nötigen Matrizen in C-Quellcode übernimmt. Generiert werden dabei sowohl eine Header-Datei „`ssm_matrix.h`“ sowie eine C-Datei „`ssm_matrix.c`“. Dadurch können Änderungen der Reglerparameter schnell übernommen werden. Aufgrund der relativ großen Matrizen wäre ein manuelles Abschreiben sehr zeitaufwändig, weshalb sich dieser Schritt schon nach kurzer Zeit auszahlt.

Die Berechnungsvorschriften für die Zustandsregelung sind aus den vorhergehenden Kapiteln bekannt und werden nicht erneut behandelt. Für weitergehende Informationen sei auf den Quellcode hingewiesen. Die Programmierung der Matrizenbechnung erfolgt in „`matrix.c`“, die der Zustandsregelung in „`ssm.c`“.

## 6.3 Dezentrale Regelung

Für die dezentrale Regelung muss die Rekursionsgleichung aus dem Matlab PID-Regelalgorithmus ermittelt werden. Die Realisierung des statischen Entkopplungs-

glieders erfolgt mittels der von der Zustandsregelung bereits implementierten Matrizenberechnung. Auch für die Generierung des Quellcodes für das Entkopplungsglied wurde ein Script in Matlab erstellt.

Der Matlab PID-Regelalgorithmus wurde bereits in Gleichung 5.10 diskretisiert. Aus dieser Gleichung muss somit lediglich die entsprechende Rekursionsgleichung ermittelt werden, um diese anschließend in C programmieren zu können.

Das diskretisierte PID-Reglergesetz wird nochmals aufgezeigt:

$$G_r(z) = \frac{u(z)}{e(z)} = P + I \cdot T_s \frac{1}{z-1} + D \frac{N}{1 + N \cdot T_s \frac{1}{z-1}} \quad (6.1)$$

Zunächst wird die Gleichung so umgeformt, dass sich ein gemeinsamer Nenner ergibt. Der Index der Abtastzeit wird nachfolgend wegen der Übersichtlichkeit nicht mitgeschrieben, sodass  $T_s = T$  entspricht.

$$\frac{u(z)}{e(z)} = P + \frac{IT}{z-1} + \frac{DN(z-1)}{z-1+NT} \quad (6.2)$$

$$\frac{u(z)}{e(z)} = \frac{IT}{z-1} + \frac{Pz - P + PNT + DN(z-1)}{z-1+NT} \quad (6.3)$$

$$\frac{u(z)}{e(z)} = \frac{ITz - IT + INT^2 + (z-1)(Pz - P + PNT + DNz - DN)}{(z-1)(z-1+NT)} \quad (6.4)$$

Alle Klammerausdrücke werden ausmultipliziert.

$$\frac{u(z)}{e(z)} = \frac{ITz - IT + INT^2 + Pz^2 - Pz + PNTz + DNz^2 - DNz}{z^2 - z + NTz - z + 1 - NT} \quad (6.5)$$

$$\frac{u(z)}{e(z)} = \frac{-Pz + P - PNT - DNz + DN}{z^2 - z + NTz - z + 1 - NT}$$

$$\frac{u(z)}{e(z)} = \frac{(P + DN)z^2 + (P(NT - 2) + IT - 2DN)z}{z^2 + (NT - 2)z + (1 - NT)} \quad (6.6)$$

$$\frac{u(z)}{e(z)} = \frac{+(P(1 - NT) + IT(NT - 1) + DN)}{z^2 + (NT - 2)z + (1 - NT)}$$

Es werden nachfolgend aufgeführte Koeffizienten eingeführt:

$$\begin{aligned}c_1 &= 2 - NT \\c_2 &= NT - 1 \\c_3 &= P + DN \\c_4 &= P(NT - 2) + IT - 2DN \\c_5 &= P(1 - NT) + IT(NT - 1) + DN\end{aligned}$$

Mit den eingeführten Koeffizienten lautet die Gleichung wie folgt:

$$\frac{u(z)}{e(z)} = \frac{c_3 z^2 + c_4 z + c_5}{z^2 - c_1 z - c_2} \quad (6.7)$$

$$\frac{u(z)}{e(z)} = \frac{c_3 + c_4 z^{-1} + c_5 z^{-2}}{1 - c_1 z^{-1} - c_2 z^{-2}} \quad (6.8)$$

Die Rekursionsgleichung lautet:

$$u[k] = c_1 u[k - 1] + c_2 u[k - 2] + c_3 e[k] + c_4 e[k - 1] + c_5 e[k - 2] \quad (6.9)$$

Diese Rekursionsgleichung kann leicht in C-Code implementiert werden. Bei der Implementierung werden zudem einstellbare Stellgrößenbeschränkungen mit eingebunden, die gerade bei der Verwendung von integrativen Regleranteilen sehr wichtig sind. Um eine einfache Art der Verwendung mehrerer Regler im Code zu erlauben, erfolgt die Programmierung der Regler über Strukturen. Dadurch können mehrere Reglerinstanzen erzeugt werden, die alle auf die gleichen Funktionen zugreifen. Die gemeinsam genutzten Funktionen dienen der Initialisierung der definierten Parameter  $c_1$  bis  $c_5$ , der Berechnung der Rekursionsgleichung und einem Rücksetzen der Regler.

## 7 Vergleich der implementierten Regelungskonzepte

In diesem Kapitel werden die implementierten Regelungskonzepte miteinander verglichen. Es wird zunächst der Aufwand der Reglerauslegungen und der Implementierungen behandelt. Anschließend werden Referenzflüge bzw. Testszenarien definiert, die für die Regelungen durchgeführt werden. Die Flugtests beschränken sich dabei auf die Zustandsregelung mit Störgrößenkompensation und die entkoppelte kaskadierte Regelung. Zu guter Letzt folgt eine Beurteilung aus Sicht des Piloten über die Flugfähigkeit der Regelungen.

### 7.1 Aufwand der Reglerauslegungen und Implementierungen

Die Auslegung der Zustandsregelung über das LQR-Verfahren erweist sich als überaus einfach. Über die Gewichtung der Zustände kann das Systemverhalten für alle Systemgrößen unabhängig voneinander eingestellt werden. Schon der erste Flugversuch mit der LQR-Regelung erwies sich als funktionsfähig, was äußerst beeindruckend und nicht selbstverständlich ist.

Ganz anders sieht es im Fall der Auslegung der dezentralen Regelung aus. Der Entwurf des Entkopplungsgliedes ist in diesem Fall nicht ausschlaggebend und fällt relativ leicht. Problematisch ist vor allem, dass alle vier Regelungen einwandfrei funktionieren müssen, da der Quadrocopter ansonsten nicht flugfähig ist. Gerade die Regelung der Strecken Pitch und Roll erweist sich als sehr schwierig. Nur der Weg über den Einsatz einer kaskadierten Regelung führt zu einer Stabilisierung der beiden Strecken. Die Einstellung der Einzelregler ist extrem langwierig und führt bei zu starken Einstellungen der Reglerparameter schnell zu einem schwingfähigen System. Darüber hinaus stellt sich bei jeder Regelung der Einzelstrecken die Frage, welche Reglerart sich am ehesten eignet. So müssen für die kaskadierte dezentrale Regelung insgesamt fünf Regler ausgewählt und entsprechend ausgelegt werden.



Bei der Implementierung hingegen ist die Kehrseite der LQR-Regelung zu sehen. Es müssen Funktionen zur Matrizenberechnung und das intern im Mikrocontroller mitlaufende diskrete Zustandsraummodell zur Störgrößenkompensation implementiert werden. Dies erfordert entsprechend viel Zeit. Auf eine Matrizenberechnung könnte im Fall der dezentralen Regelung, aufgrund des einfachen Entkopplungsgliedes, verzichtet werden. Stattdessen könnte die Berechnung der Entkopplung manuell im Quellcode über eine Funktion erfolgen. Für die Implementierung muss somit lediglich die Rekursionsgleichung des Reglergesetzes ermittelt und implementiert werden.

Insgesamt kann der geringere Zeitaufwand für die Implementierung der dezentralen Regelung jedoch in keinster Weise den Aufwand für die Reglerauswahl und deren Einstellungen kompensieren. Die LQR-Regelung konnte inklusive der Störgrößenkompensation innerhalb von etwa zehn Werktagen ausgelegt und im Quellcode eingebunden werden. Die Zeit für den Entwurf der dezentralen Regelung kann lediglich auf mehrere Wochen bis Monate beziffert werden und erweist sich als sehr kompliziert und teils auch langwierig.

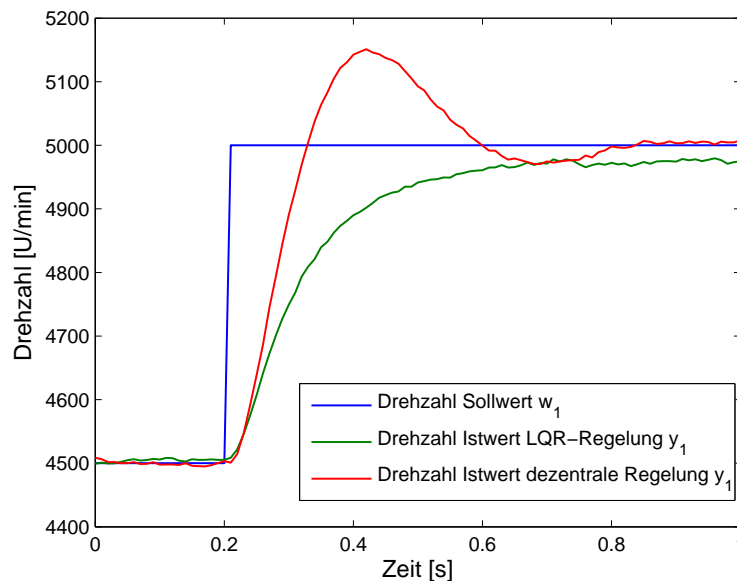
## 7.2 Testszenarios und Referenzflüge

Bedingt dadurch, dass kein Testgestell für den Quadrocopter vorhanden ist, auf welchem dieser fest montiert werden kann, müssen alle Tests am frei fliegenden Modell durchgeführt werden. Die einzige Ausnahme bildet der Test, wie gut neue Drehzahlsollwerte aller vier Rotoren angenommen werden, die für den Auftrieb verantwortlich sind. Der Vorteil der Freiflugtests besteht darin, dass reale Störgrößen bei der Betrachtung mit einfließen. Der große Nachteil dieser Methode ist allerdings, dass für die Beurteilung der Reglerdynamik am realen System keine einfachen Sollwertsprünge vorgenommen werden können. Vor allem aufgrund von auftretendem Wind, wenn auch nur in schwacher Form, wird das Modell trotz horizontaler Ausrichtung von seinem Testzentrum aus wegbewegt. Korrigierende Steuermaßnahmen durch den Piloten müssen somit in jedem Fall möglich sein, um eine permanente Kontrolle über den Quadrocopter zu behalten. Durch diesen Umstand müssen Testszenarios und Referenzflüge manuell vom Piloten durchgeführt werden. Diese werden jedoch möglichst so gewählt, dass eine Vergleichbarkeit möglich ist.

Alle Flüge, die im Folgenden miteinander verglichen werden, werden innerhalb kürzester Zeit hintereinander durchgeführt, um vergleichbare Wetter- und Windverhältnisse zu gewährleisten. Äußerst hilfreich ist dabei die Umschaltung der verschiedenen Reglervarianten über die Fernsteuerung.

### 7.2.1 Drehzahlannahme der vier Rotordrehzahlen

Für den Auftrieb sind die vier Rotordrehzahlen, die in der Modellbildung in Kapitel 3.5.5 im Ausgang  $y_1$  zusammengefasst wurden, verantwortlich. Es soll an dieser Stelle überprüft werden, wie gut ein neuer Drehzahlsollwert angeregt wird und ob dabei eine bleibende Regeldifferenz auftritt. Dazu wird der Quadrocopter auf dem Boden festgehalten und in etwa die zum Schweben notwendige Drehzahl von 4500 U/min als Sollwert eingestellt. Anschließend erfolgt ein Sollwertsprung auf 5000 U/min.



**Bild 7.1:** Vergleich der Drehzahlregelung zwischen LQR-Regelung und dezentraler Regelung

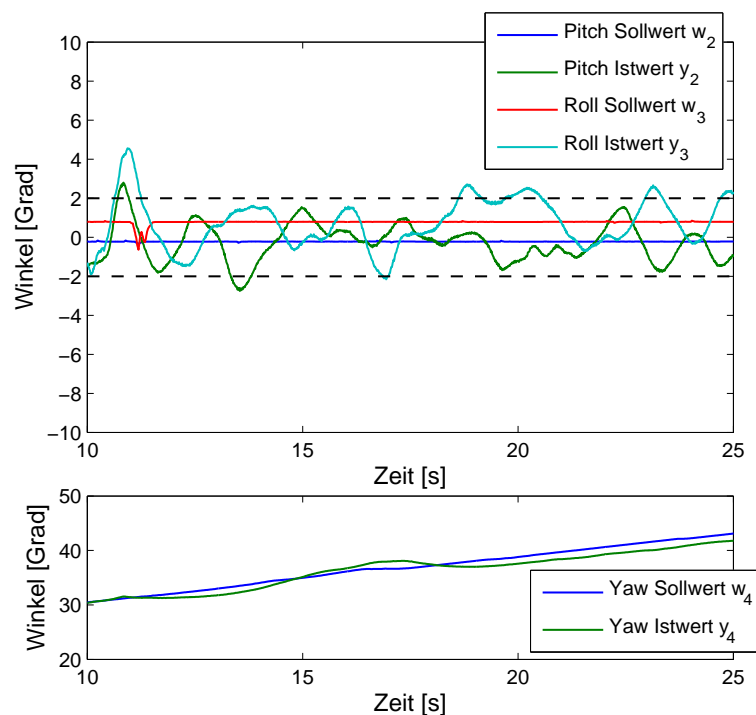
Das Ergebnis der Messung ist aus Bild 7.1 ersichtlich. Beide Regelungsarten haben in etwa 0,6 s nach dem Sollwertsprung ihren stationären Endwert erreicht. Die dezentrale Regelung der Strecke, realisiert mit dem PI-Regler, weist ein deutliches Überschwingen auf, welches jedoch den Vorgaben der Auslegung aus Kapitel 5.2.2 entspricht. Der stationäre Endwert entspricht, abgesehen von den Störungen, genau dem Sollwert. Die Regelung der Drehzahl mittels der LQR-Regelung weist kein Überschwingen auf, erreicht den stationären Endwert jedoch in der gleichen Zeit wie bei der dezentralen Regelung. Es ist jedoch eine bleibende Regeldifferenz zu erkennen. Dieses lässt sich dadurch begründen, dass die Drehzahlregelstrecke mit einem PT<sub>1</sub>-Verhalten nachgebildet wird, bei dem von einem linearen Systemverhalten ausgegangen wird. Das System verhält sich jedoch in Realität nicht komplett linear, weshalb die LQR-Regelung eine geringe, akzeptable Regeldifferenz aufweist.

Beide Regelungsarten zeigen eine gute Drehzahlannahme und eignen sich daher beide gleichermaßen gut. Die geringe Regeldifferenz der LQR-Regelung kann kaum als Nachteil angesehen werden, da diese bei der manuellen Steuerung über die Fernsteuerung nicht auffallen sollte.

### 7.2.2 Störgrößenkompensation im Schwebeflug

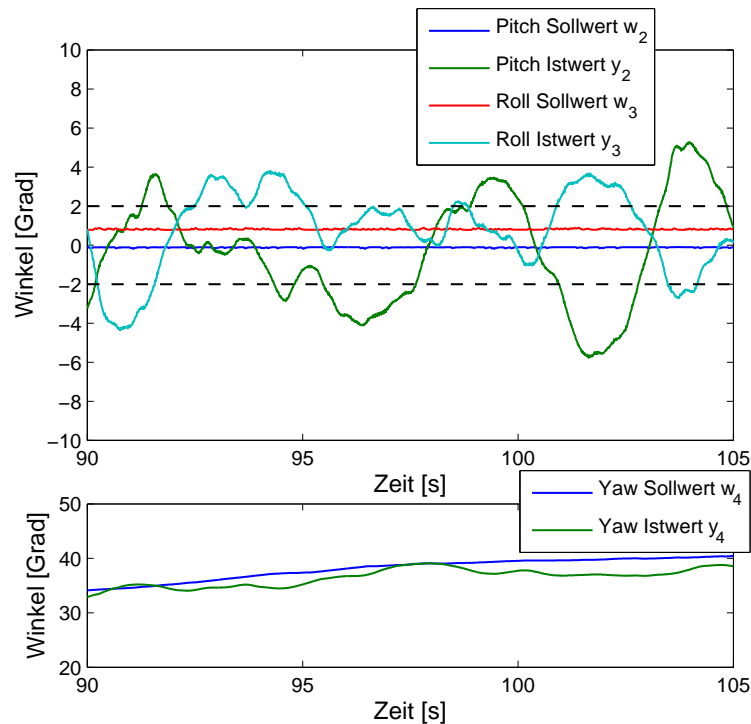
Ein sehr einfacher Test besteht darin, einen Schwebeflug des Quadropters für ca. 15 s lang durchzuführen, ohne Lagekorrekturen mit der Fernsteuerung vorzunehmen. Es werden lediglich eventuell notwendige Korrekturen aller Rotordrehzahlen vorgenommen, um die Flughöhe beizubehalten. Die Flughöhe wird auf 1,5 m definiert. In dieser Höhe ist kein sogenannter Bodeneffekt mehr vorhanden, bei dem der Quadropters durch seinen eigenen Auftrieb Turbulenzen erzeugt, die das System stören.

Anhand dieses Tests soll beurteilt werden, wie gut auftretende Störgrößen ausgeregelt werden. Betrachtet werden dabei die Systemausgänge Pitch, Roll und Yaw.



**Bild 7.2:** LQR-Regelung mit Störgrößenkompensation im Schwebeflug

Die Bilder 7.2 und 7.3 zeigen die Aufnahmen der beiden Regelungen im Schwebeflug, die beide innerhalb von zwei Minuten über die Umschaltung der Regelungen mittels



**Bild 7.3:** Dezentrale kaskadierte Regelung im Schwebeflug

der Fernbedienung aufgenommen wurden. Dadurch herrschen in diesem Fall nahezu gleiche Bedingungen für beide Regelungsarten. Es wurden zur besseren Vergleichbarkeit der Lagewinkel für Pitch und Roll in beiden Aufnahmen gestrichelte Linien bei  $\pm 2$  Grad eingezeichnet. Deutlich erkennbar ist, dass die LQR-Regelung (Bild 7.2) auftretende Störgrößen besser ausregelt, sodass die Lage nur in wenigen Fällen die markierte Grenze von  $\pm 2$  Grad über- bzw. unterschreitet. Anders sieht es bei der dezentralen kaskadierten Regelung aus. Die Abweichungen fallen sehr viel größer aus und über- bzw. unterschreiten teils sogar  $\pm 5$  Grad.

Die Regelung der Strecke Yaw führt bei der LQR-Regelung zu etwas geringeren Abweichungen als bei der dezentralen kaskadierten Regelung. Der Unterschied ist in diesem Fall allerdings weniger groß als bei Pitch und Roll.

Insgesamt kann zusammengefasst werden, dass die LQR-Regelung mit der implementierten Störgrößenkompensation eine sehr viel bessere Ausregelung von Störgrößen ermöglicht.

### 7.2.3 Untersuchung der vollständigen Entkopplung

Ein weiterer Test soll darin bestehen, zu untersuchen, ob die entworfenen Regelungen tatsächlich zu einer vollständigen Entkopplung des Systems führen, wenn unterschiedliche Sollwerte auf allen zu regelnden Größen vorgegeben werden. Gerade real vorhandene Stellgrößenbeschränkungen, die bei der Simulation eventuell nicht berücksichtigt werden, können die theoretisch mögliche vollständige Entkopplung beeinflussen.

Die Messergebnisse zweier Flüge mit den unterschiedlichen Regelungsarten sind in den Bildern 7.4 und 7.5 zu erkennen. Exakt gleiche Flüge konnten nicht durchgeführt werden, da das Modell durch vorhandenen Wind ohne korrigierende Steuermaßnahmen abdriften würde.

Gut ersichtlich ist, dass die LQR-Regelung (Bild 7.4) alle Regelgrößen voneinander entkoppelt. Es sind keinerlei Beeinflussungen der einzelnen Größen zueinander erkennbar.

Die dezentrale kaskadierte Regelung (Bild 7.5) ist hinsichtlich der Entkopplung etwas problematischer. Die Vorgabe von unterschiedlichen Drehzahl-, Pitch- und Roll-Sollwerten führt zu keinen erkennbaren Beeinflussungen der anderen Größen. Es zeigt sich allerdings bei der Sollwertvorgabe von Yaw das Problem, dass sich eine sehr große Störung im Lagewinkel Pitch ergibt. Der kritische Zeitpunkt wurde in Bild 7.5 mit einer rot gestrichelten vertikalen Linie bei etwa 209s verdeutlicht. Die Auswirkung ist derart stark, dass es ohne entsprechendes Gegensteuern zum Absturz des Quadropters gekommen wäre. Die Ursache der Störung konnte ermittelt werden. Durch die Sollwertvorgabe in diesem Bereich soll der Quadropters seinen Drehwinkel Yaw hin zum Negativen ändern. Dazu werden nach Kapitel 2.2, in dem die Funktionsweise des Quadropters erläutert wird, die Motoren M0 und M2 mit einem niedrigeren und M1 und M3 mit einem höheren Strom beaufschlagt. Die Motoren M0 und M2 erreichen dabei durch die Regelung von Yaw einen derart niedrigen Stromsollwert, dass die Drehzahlen in einen kritischen Bereich sinken. In diesem Bereich sind die unteren Stellgrößenbeschränkungen aktiv, die den minimalen Strom begrenzen, und die Schubkraftkennlinie ist weit außerhalb des Bereichs, in dem die Systemlinearisierung stattgefunden hat. Dadurch kann die Regelung von Pitch nicht mehr effektiv die Lage regeln, die zur Lageänderung die Motoren M0 und M2 benötigt. Abhilfe könnte einerseits eine Stellgrößenbeschränkung für die Regelung von Yaw bringen, andererseits könnte auch die Regelung von Yaw weniger stark eingestellt werden. Die Beseitigung dieses Problems wurde jedoch nicht näher untersucht.

Die Entkopplungsproblematik der dezentralen Regelung tritt bei der LQR-Regelung nicht auf, obwohl beide Regelungsarten von der Dynamik her in etwa gleich eingestellt wurden. In diesem Fall ist die LQR-Regelung für die Systementkopplung klar im Vorteil.

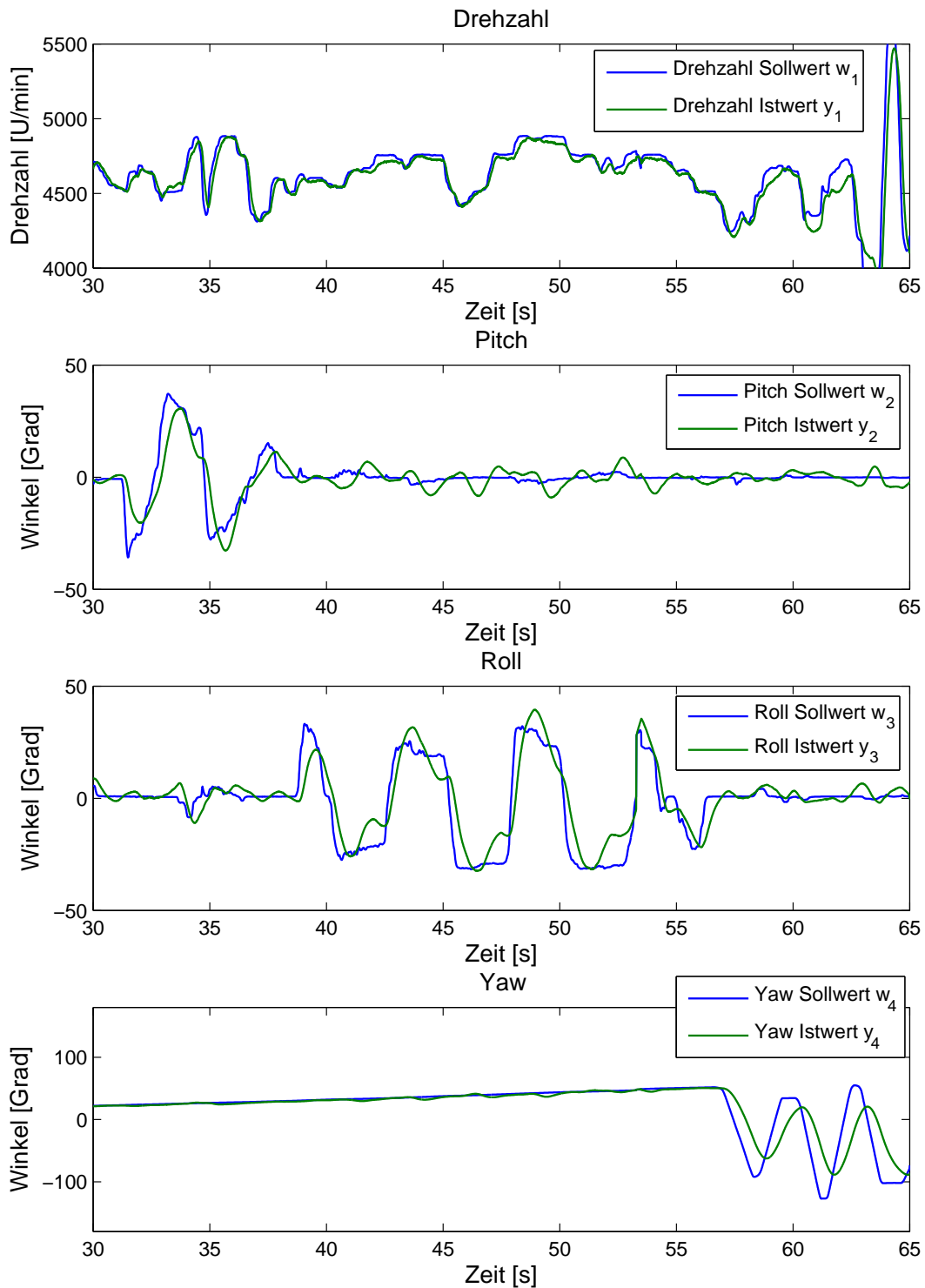
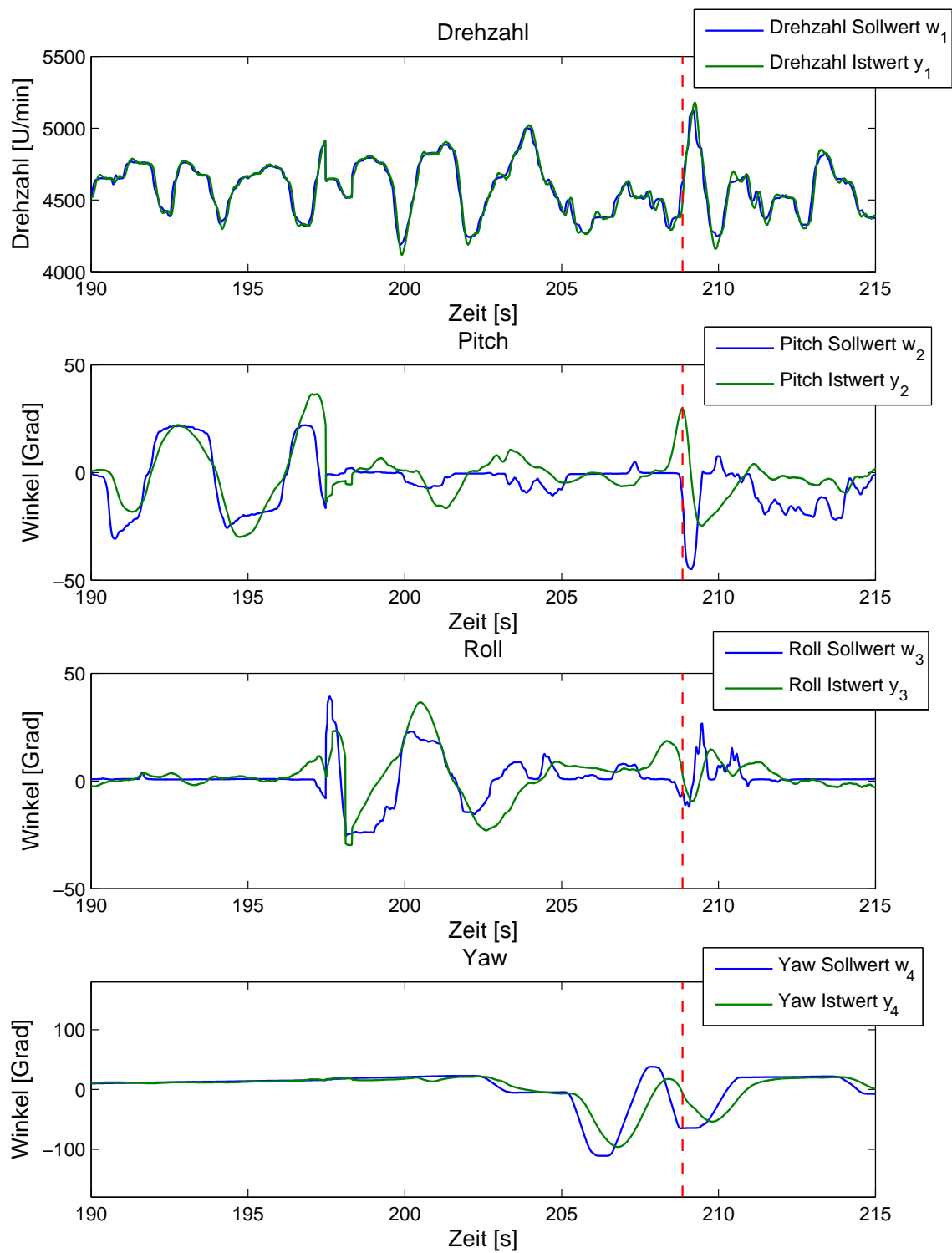


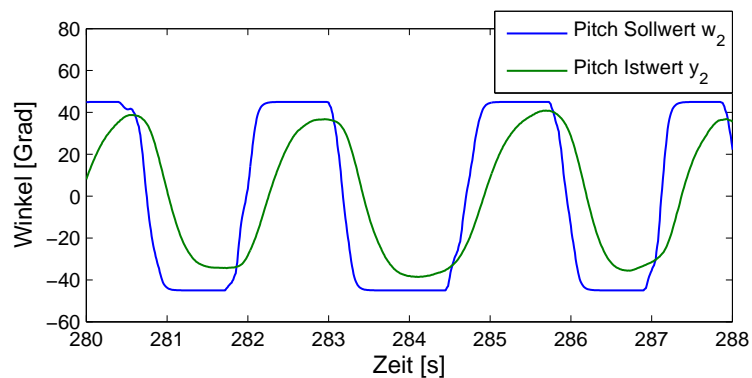
Bild 7.4: Flugaufzeichnung der LQR-Regelung mit Störgrößenkompensation



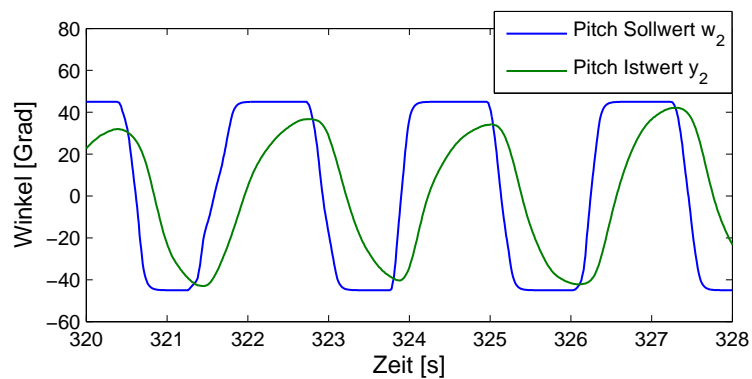
**Bild 7.5:** Flugaufzeichnung der dezentralen kaskadierten Regelung

### 7.2.4 Gezielte Steuereingaben zur Erlangung einer Systeminstabilität

Als die am schwierigsten zu regelnden Größen haben sich vor allem bei der dezentralen Mehrgrößenregelung das Verhalten von Pitch und Roll herausgestellt. Es soll in diesem Test versucht werden, durch gezielte Steuereingaben an der Fernbedienung für die Pitch-Lage eine Instabilität des Systems zu erreichen.



**Bild 7.6:** Erlangung einer Instabilität der LQR-Regelung von Pitch



**Bild 7.7:** Erlangung einer Instabilität der dezentralen kaskadierten Regelung von Pitch

Die Flugdaten sind in den Bildern 7.6 und 7.7 ersichtlich und zeigen nahezu identisches Verhalten. Beide Regelungsarten führen zu einer hervorragenden Systemstabilität, die sich durch gezielte Sollwertvorgaben nicht zur Instabilität hin bewegen lässt. Dies würde allerdings auch unweigerlich zum Absturz des Modells führen. Es lässt sich gut erkennen, dass beide Systeme hinsichtlich der Dynamik von Pitch, und folglich aufgrund gleicher Einstellungen auch Roll, nahezu gleiches Verhalten aufweisen.



## 7.3 Beurteilung des Flugverhaltens aus Pilotensicht

In diesem Abschnitt wird überprüft, wie der Pilot, der den Quadrocopter steuert, die unterschiedlichen Regelungsarten wahrnimmt. Es sei vorab darauf hingewiesen, dass es sich um ein persönliches Urteil handelt, das von jedem anders empfunden werden kann.

### 7.3.1 Steuerung des Auftriebs

Die Steuerung des Auftriebs funktioniert sowohl mit der LQR-Regelung als auch mit der dezentralen Regelung gleichermaßen gut. Es zeigt sich bei beiden Varianten eine sehr gute Drehzahlannahme aller vier Rotoren, die für den Auftrieb zuständig sind. Der einzig wahrnehmbare Unterschied ist, dass die dezentrale Regelung eine deutlich potentere Drehzahlannahme vermittelt. Dies lässt sich auch anhand des Vergleichs der Sprungantworten aus Bild 7.1 nachvollziehen. Das deutlich vorhandene Überschwingen bei der dezentralen Regelung vermittelt die spontanere Drehzahlannahme.

Die geringe Regeldifferenz bei der LQR-Regelung ist, wie erwartet, nicht wahrnehmbar.

### 7.3.2 Ausregelung von Störgrößen

Während des Fluges lässt sich erkennen, dass die LQR-Regelung die Störgrößen besser kompensiert als es die dezentrale Regelung vermag. Es sind somit bei der LQR-Regelung weniger Korrekturen für den Schwebeflug notwendig, die ein einfacheres Fliegen des Modells ermöglichen und dem Piloten mehr Vertrauen in die Lageregelung vermittelt.

### 7.3.3 Steuerung von Pitch und Roll

Die Sollwertannahme von Pitch und Roll funktioniert bei beiden Regelungsarten gleichermaßen gut und es ist kaum ein Unterschied wahrnehmbar. Die schlechtere Störgrößenkompensation der dezentralen Regelung ist allerdings auch bei der Steuerung von Pitch und Roll bemerkbar. Als Pilot entsteht dadurch das Gefühl, der Quadrocopter nehme die Steuerbefehle nicht richtig an.

### 7.3.4 Steuerung von Yaw

Hinsichtlich der Dynamik sind beide Regelungsarten bei der Regelung von Yaw gleichauf. Störgrößen sind nicht ersichtlich. Das größte Problem stellt allerdings die Beeinflussung der Sollwertvorgaben von Yaw auf die Pitch und Roll bei der dezentralen Regelung dar. Diese relativ unerwartet und plötzlich auftretenden Beeinflussungen sorgen für eine Unkontrollierbarkeit des Modells und können unter Umständen zu einem Absturz führen, wenn der Pilot nicht entsprechend gegensteuert.

### 7.3.5 Gesamteindruck

Die Tatsache, dass der Entwickler beider Regelungen zugleich auch der Pilot des Quadropters ist, führt dazu, dass die LQR-Regelung ein höheres Vertrauen genießt. Dies lässt sich vor allem dadurch begründen, dass gleich der erste LQR-Reglerentwurf zu einem sehr annehmbaren Flugverhalten führte. Es mussten somit lediglich „kleinere“ Optimierungen in der Parametrierung und der Störgrößenkompensation vorgenommen werden. Das Flugverhalten zeigt sich dabei als sehr vorhersehbar und es entsteht zu keiner Zeit der Eindruck, der Quadropters würde nicht auf Steuerbefehle reagieren bzw. diese nicht korrekt umsetzen. So konnte mit der LQR-Regelung sogar bei relativ viel Wind geflogen werden, bei dem zwar viel Flugerfahrung nötig ist, das Modell allerdings trotzdem noch fliegt. Auch wurde bereits eine Digitalkamera auf dem Quadropters befestigt, um eigene Luftaufnahmen zu erstellen.

Die dezentrale Regelung konnte erst nach diversen Einstellungen und Tests den Erstflug absolvieren. Die mäßige Störgrößenkompensation und die nicht vollständige Entkopplung führen zu einer Meidung dieser Regelungsart. Abstürze wären bei reiner Nutzung dieser Regelung, bei derzeitigem Entwicklungsstand, ansonsten vermutlich die Folge.

## 8 Fazit und Ausblick

Ziel dieser Arbeit war die Entwicklung, Implementierung und der Vergleich mindestens zweier verschiedener Regelungskonzepte zur Lageregelung eines Quadropters. Hierzu wurde ein Quadropter mit der zum Betrieb notwendigen Hardware und den vier eigenen Motorreglern aus der Bachelorarbeit aufgebaut. Es wurde ein regelungstechnisches Modell gebildet und dessen Parameter bestimmt.

Als Regelungen erwiesen sich die LQR-Mehrgrößenregelung und die dezentrale kaskadierte Regelung als funktions- und flugfähig. Die dezentrale Regelung ohne eine Kaskadierung hingegen führte zu keiner Systemstabilität. Aus dieser Erkenntnis kann darauf geschlossen werden, dass ein Quadropter nicht ohne die Hilfe einer Lageregelung von einem Piloten geflogen werden kann, da dieser nie in der Lage wäre, die Funktion einer kaskadierten Regelung zu übernehmen.

Der Entwurf und die Implementierung der LQR-Mehrgrößenregelung ließen sich wider Erwarten sehr gut und einfach einstellen und zeigten bei den ersten Flugversuchen eine hervorragende Systemstabilität. Die Auslegung der dezentralen Regelungen kann nur als sehr schwierig und aufwändig bezeichnet werden. Die Einstellmöglichkeiten sind derart vielfältig, dass eine optimale Parametrierung kaum möglich ist. Dennoch konnten sowohl die LQR-Regelung als auch die dezentrale kaskadierte Regelung derart parametrieren, dass diese ähnliche Dynamiken im Regelungsverhalten aufweisen. Jedoch erweist sich die Flugfähigkeit der LQR-Regelung durch die bessere Störgrößenbeseitigung und Entkopplung als deutlich besser steuerbar und vermittelt dem Piloten mehr Vertrauen in die Steuerbarkeit des Modells.

Der Einsatz der eigenen Motorregler aus der Bachelorarbeit ist ein voller Erfolg und es traten nach der Beseitigung einiger kleinerer Hard- und Softwarefehler keinerlei Probleme beim Betrieb auf. Auch der Einsatz von anderen Motoren mit dem Regler funktionierte ohne Probleme.

Der Test der Regelungen am frei fliegenden Modell ohne Zuhilfenahme eines Testgestells ist nicht unbedingt für jeden Entwickler empfehlenswert. Die Regelungen konnten durch diese Maßnahme jedoch sehr gut auf äußere Störgrößeneinflüsse hin untersucht und optimiert werden. Gerade auftretender Wind erweist sich für die Lage von Pitch und Roll als großer Störeinfluss. Es konnten entsprechende Tests durchgeführt werden, die eine Vergleichbarkeit der Regelungen ermöglichen. Ideal für den Vergleich

wären Sollwertsprünge für alle Regelgrößen gewesen, die jedoch aufgrund des Freiflugs nicht durchgeführt werden konnten. Ohne vorherige eigene Flugerfahrung mit Modellhelikoptern wäre dieses Unterfangen der Tests jedoch zum Scheitern verurteilt gewesen, da bei einer nicht perfekt funktionierenden Regelung intuitive und schnelle Reaktionen des Piloten an der Fernsteuerung gefordert sind. Zudem führen schon kleinste Programmierfehler zu unerwartetem Systemverhalten, sodass einmal auch die eigene Hand als „Notaus“ dienen musste.

Insgesamt kann die Untersuchung verschiedener Regelungskonzepte zur Lageregelung eines Quadropters in dieser Arbeit als voller Erfolg angesehen werden. Durch den Einsatz der eigens konstruierten Motorregler mit Stromregelung konnte zudem die Aussage aus dem in der Einleitung erwähnten IEEE-Artikel widerlegt werden, dass die Zustandsregelung langsamer ist und weniger Dynamik aufweist.

Entscheidende Verbesserungen könnten sich hinsichtlich der Regelung noch durch den Einsatz eines Kalmanfilters ergeben. Die Sensorfusion würde in diesem Fall im STM32F4-Mikrocontroller modellbasiert erfolgen. Möglicherweise ergeben sich dadurch höhere Abtastfrequenzen, die zu einer schnelleren Störgrößenkompensation führen könnten und die Regelungen könnten hinsichtlich ihrer Dynamiken noch schneller eingestellt werden.

# Literaturverzeichnis

- [Baránek und Solc 2012] BARÁNEK, Radek ; SOLC, Frantisek: *Modelling and Control of a Hexa-copter*, Brno University of Technology, IEEE-Artikel, 2012
- [Bouabdallah u. a. 2004] BOUABDALLAH, Samir; NOTH, André; SIEGWAN, Roland: *PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor*, Swiss Federal Institute of Technology, IEEE-Artikel, 2004
- [Bresciani 2008] BRESCIANI, Tommaso: *Modelling, Identification and Control of a Quadrotor Helicopter*, Lund University, Dissertation, 2008
- [Claussen 2009] CLAUSSEN, Prof. Dr.-Ing. Ulf: *Kompendium Regelungstechnik*, Hochschule für angewandte Wissenschaften Hamburg, Script, 3. Juni 2009
- [CoIDE ] : *CooCox CoIDE*. – URL [http://www.coocox.org/CooCox\\_CoIDE.htm](http://www.coocox.org/CooCox_CoIDE.htm). – Zugriffsdatum: 16. 03. 2014
- [Dremel 2008] DREMEL, Benedikt: *Konzeption und Realisierung einer Regelungs- und Steuerungssoftware für einen Quadcopter*, Friedrich-Alexander-Universität Erlangen-Nürnberg - Department Informatik, Studienarbeit, 2008
- [e-volo ] WOLF, Stephan; ZOSEL, Alexander: *Volocopter*. – URL <http://www.e-volo.com>. – Zugriffsdatum: 29. 01. 2014
- [Ebel und Bliefert 2003] EBEL, Hans F. ; BLIEFERT, Claus: *Diplom- und Doktorarbeit*. WILEY-VCH (Verlag), 2003. – ISBN 3-527-30754-0
- [Hoffmann u. a. 2010] HOFFMANN, Frank; GODDEMEIER, Niklas ; BERTRAM, Torsten: *Attitude estimation and control of a quadrocopter*, Technische Universität Dortmund, IEEE-Artikel, 2010
- [Holzhüter 2012] HOLZHÜTER, Prof. Dr. Thomas: *A few basic topics in Dynamical Systems*, Hochschule für angewandte Wissenschaften Hamburg, Script, 2012
- [Jensen 2006] JENSEN, Björn: *Arbeit mit Latex-Vorlage zur Erstellung wissenschaftlicher Arbeiten*. 2006. – URL <http://www.iee.haw-hamburg.de/Druckvorlagen.176.0.html>. – Zugriffsdatum: 15. 05. 2012

- [Jeremia u. a. 2012] JEREMIA, Stevie; KUANTAMA, Endrowednes; PANGARIBUAN, Julinda: *Design and Construction of Remote-Controlled Quad-copter Based on STC12C5624AD*, Universitas Pelita Harapan, IEEE-Artikel, 2012
- [Lee u. a. 2011] LEE, Keun Uk; YUN, Young Hun; CHANG, Wook; PARK, Jin Bae; CHOI, Yoon Ho: *Modeling and Altitude Control of Quad-rotor UAV*, Yonsei University, IEEE-Artikel, 2011
- [Lunze 2010a] LUNZE, Prof.Dr.-Ing. Jan: *Regelungstechnik 1*. Springer (Verlag), 2010. – ISBN 978-3-642-13807-2
- [Lunze 2010b] LUNZE, Prof.Dr.-Ing. Jan: *Regelungstechnik 2*. Springer (Verlag), 2010. – ISBN 978-3-642-10197-7
- [mikrokopter MK-Funktion ] BUSS, Holger; BUSKER, Ingo: *Funktionen des Mikro-Kopter*. – URL <http://www.mikrokopter.de/ucwiki/MK-Funktion>. – Zugriffsdatum: 29.01.2014
- [mikrokopter Summensignal ] BUSS, Holger; BUSKER, Ingo: *SummenSignal*. – URL <http://www.mikrokopter.de/ucwiki/SummenSignal>. – Zugriffsdatum: 29.01.2014
- [MPU6050 Datasheet ] INVSENSE: *MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2*. – URL <http://www.invensense.com/mems/gyro/documents/RM-MPU-6000A-00v4.2.pdf>. – Zugriffsdatum: 19.03.2014
- [MPU6050 Description ] INVSENSE: *MPU-6000/6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking™ Devices*. – URL <http://www.invensense.com/mems/gyro/mpu6050.html>. – Zugriffsdatum: 14.03.2014
- [MPU6050 DMP ] ROWBERG, Jeff: *MPU-6050 6-axis accelerometer/gyroscope*. – URL <http://www.i2cdevlib.com/devices/mpu6050#source>. – Zugriffsdatum: 10.03.2014
- [MPU9150 Description ] INVSENSE: *MPU-9150 Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking™ Device*. – URL <http://www.invensense.com/mems/gyro/mpu9150.html>. – Zugriffsdatum: 19.03.2014
- [Phang u. a. 2012] PHANG, Swee K. ; CAI, Chenxiao ; CHEN, Ben M. ; LEE, Tong H.: *Design and Mathematical Modeling of a 4-Standard-Propeller (4SP) Quadrotor*, National University of Singapore, IEEE-Artikel, 2012
- [Puls 2011] PULS, Tim: *Lokalisations- und Regelungsverfahren für einen 4-Rotor-Helikopter*., Carl von Ossietzky Universität Oldenburg, Dissertation, 2011

- [Razak u. a. 2012] RAZAK, Noorfadzli Abdul ; ARSHAD, Nor Hashim Mohd ; ADNAN, Ramli ; MISNAN, Mohamad Farid ; THAMRIN, Norashikin M. ; MAHMUD, Siti Fatimah: *A Study of Kalman's Filter in Embedded Controller for Real-Time Quadrocopter Roll and Pitch Measurement*, Universiti Teknologi Mara, IEEE-Artikel, 2012
- [Röther 2013] RÖTHER, Prof. Dr.-Ing. Michael: *A-M1-AS - Antriebstechnik für mobile Systeme*, Hochschule für angewandte Wissenschaften Hamburg, Script, 2013
- [Schoellig u. a. 2012] SCHOELLIG, Angela P. ; WILTSCHE, Clemens ; D'ANDREA, Raffaello: *Feed-Forward Parameter Identification for Precise Periodic Quadrocopter Motions*, ETH Zurich, IEEE-Artikel, 2012
- [Schulz und Graf 2013] SCHULZ, Prof. Dr. G. ; GRAF, Prof. Dr. K.: *Regelungstechnik 2*. Oldenbourg (Verlag), 2013. – ISBN 978-3-486-71281-0
- [Schulz 2010] SCHULZ, Prof. Dr. G.: *Regelungstechnik 1*. Oldenbourg (Verlag), 2010. – ISBN 978-3-486-59194-1
- [Silverlit ] V. o.: *Silverlit X-UFO*. – URL [http://multicopter.org/wiki/Silverlit\\_X-UFO](http://multicopter.org/wiki/Silverlit_X-UFO). – Zugriffsdatum: 29.01.2014
- [STM32F4 Discovery Manual ] STMICROELECTRONICS: *UM1472 User Manual - Discovery kit for STM32F407/417 lines*. – URL [http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user\\_manual/DM00039084.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/DM00039084.pdf). – Zugriffsdatum: 19.03.2014
- [STM32F4 Programming ] STMICROELECTRONICS: *PM0214 Programming manual - STM32F3xxx and STM32F4xxx Cortex-M4 programming manual*. – URL [http://www.st.com/st-web-ui/static/active/en/resource/technical/document/programming\\_manual/DM00046982.pdf](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/programming_manual/DM00046982.pdf). – Zugriffsdatum: 19.03.2014
- [STM32F4 Reference Manual ] STMICROELECTRONICS: *RM0090 Reference manual - STM32F405xx/07xx, STM32F415xx/17xx, STM32F42xxx and STM32F43xxx advanced ARM-based 32-bit MCUs*. – URL [http://www.st.com/st-web-ui/static/active/en/resource/technical/document/reference\\_manual/DM00031020.pdf?s\\_searchtype=keyword](http://www.st.com/st-web-ui/static/active/en/resource/technical/document/reference_manual/DM00031020.pdf?s_searchtype=keyword). – Zugriffsdatum: 19.03.2014
- [STM32F4 SDIO ] V. o.: *13-FATFS\_SDIO-Library (STM32F4)*. – URL [http://mikrocontroller.bplaced.net/wordpress/?page\\_id=621](http://mikrocontroller.bplaced.net/wordpress/?page_id=621). – Zugriffsdatum: 01.02.2014
- [STM32F407xx Datasheet ] STMICROELECTRONICS: *STM32F405xx STM32F407xx - ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera*. –

- URL <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf>. – Zugriffsdatum: 19.03.2014
- [Tagesschau Drohne 2013] V. o.: *Amazon will per Drohne liefern - Wenn der Postbote achtmal rotiert*. 2013. – URL <http://www.tagesschau.de/wirtschaft/amazon-drohnen100.html>. – Zugriffsdatum: 29.01.2014
- [Vogl 2012] VOGL, Patrick: *Realisierung einer Ansteuerung für geregelte bürstenlose Modellbaumotoren mit Busparametrierung auf einem ARM Cortex-M3-Prozessor*, Hochschule für angewandte Wissenschaften Hamburg, Bachelorthesis, 2012
- [Vojtech u. a. 2012] VOJTECH, Simak ; DUSAN, Nemeč ; JOZEF, Hrbček: *Calculation of Robot Position Utilizing Accelerometers in Non-inertial Frame of Reference*, University of Zilina, IEEE-Artikel, 2012
- [Wenck 2012/13] WENCK, Prof. Dr.-Ing. Florian: *Zustandsregelung*, Hochschule für angewandte Wissenschaften Hamburg, Script, 2012/13
- [Wenck 2013] WENCK, Prof. Dr.-Ing. Florian: *Digitale Regelungstechnik*, Hochschule für angewandte Wissenschaften Hamburg, Script, 2013
- [Wikipedia Quadrocopter ] V. o.: *Quadrocopter*. – URL <http://de.wikipedia.org/wiki/Quadrocopter>. – Zugriffsdatum: 29.01.2014
- [Wunschel u. a. 2012] WUNSCHEL, Daniel ; LANGE, Sven ; PROTZEL, Peter: *Motion Estimation for Autonomous Quadrocopter Indoor Flight*, Chemnitz University of Technology, IEEE-Artikel, 2012
- [Zhana u. a. 2012] ZHANA, Qiang ; WANG, Junqing ; XI, Xi: *Control System Design and Experiments of a Quadrotor*, Robotics Institute of Beihang University, IEEE-Artikel, 2012



# Variablenverzeichnis

<b>Symbol</b>	<b>Einheit</b>	<b>Bedeutung</b>
$i$	mA	Motorstrom
$n$	U/min	Motordrehzahl
$K_{rpm}$	$\frac{\text{U/min}}{\text{mA}}$	Streckenverstärkung PT1-Glied Drehzahlregelstrecke
$T_{rpm}$	s	Streckenzeitkonstante PT1-Glied Drehzahlregelstrecke
$M_i$	N	Inneres Drehmoment
$\Phi$	Wb	Magnetischer Hauptfluss
$K$	–	Motorkonstante
$I_A$	A	Ankerstrom
$N_E$	–	Windungszahl
$U_q$	V	Induzierte Spannung
$U_A$	V	Ankerspannung
$U_E$	V	Erregerspannung
$\Omega$	rad/s	Motordrehzahl
$R_A$	$\Omega$	Ankerwiderstand (inkl. Bürstenwiderstand)
$L_A$	H	Induktivität der Ankerwicklung
$R_E$	$\Omega$	Erregerwiderstand
$L_E$	H	Induktivität der Erregerwicklung
$J_M$	$\text{kg} \cdot \text{m}^2$	Trägheitsmoment der Maschine
$M_R$	Nm	Summe der Reibungsmomente
$F$	N	Schubkraft
$F_w$	N	Luftwiderstand

---

$l_{arm}$	m	Länge des Auslegers des Quadropters
$J_1$	$\text{kg} \cdot \text{m}^2$	Trägheitsmoment für Pitch, Roll
$J_2$	$\text{kg} \cdot \text{m}^2$	Trägheitsmoment für Yaw
$u_1$	mA	Motorstrom M0
$u_2$	mA	Motorstrom M1
$u_3$	mA	Motorstrom M2
$u_4$	mA	Motorstrom M3
$x_1$	U/min	Motordrehzahl M0
$x_2$	U/min	Motordrehzahl M1
$x_3$	U/min	Motordrehzahl M2
$x_4$	U/min	Motordrehzahl M3
$x_5$	rad	Winkel von Pitch
$x_6$	rad/s	Winkelgeschwindigkeit von Pitch
$x_7$	rad	Winkel von Roll
$x_8$	rad/s	Winkelgeschwindigkeit von Roll
$x_9$	rad	Winkel von Yaw
$x_{10}$	rad/s	Winkelgeschwindigkeit von Yaw
$y_1$	U/min	Mittelwert aller vier Rotordrehzahlen
$y_2$	rad	Winkel von Pitch
$y_3$	rad	Winkel von Roll
$y_4$	rad	Winkel von Yaw
$w_1$	U/min	Solldrehzahl für den Mittelwert aller Rotordrehzahlen
$w_2$	rad	Sollwinkel für Pitch
$w_3$	rad	Sollwinkel für Roll
$w_4$	rad	Sollwinkel für Yaw

# Glossar

<b>ARM</b>	Prozessorarchitektur, Markenname der Firma ARM
<b>C</b>	Programmiersprache
<b>CAD</b>	Computer-Aided-Design, rechnerunterstütztes Konstruieren
<b>CNC</b>	Computerized Numerical Control, computergestützte numerische Steuerung
<b>Cortex-M4</b>	Prozessorarchitektur für Mikroprozessoren der Firma ARM
<b>DMP™</b>	Digital-Motion-Processor, Prozessor der Firma InvSense zur Sensorfusionierung
<b>FATFS</b>	Open-Source-Dateisystem zur Einbindung in Embedded-Systems
<b>FIFO</b>	First In First Out, Speicherverfahren zur Datenablage
<b>FPU</b>	Floating-Point-Unit, spezieller Prozessor zur Gleitkommaberechnung
<b>I2C</b>	Inter-Integrated-Circuit, serieller Datenbus, entwickelt von Philips
<b>IC</b>	Integrated Circuit
<b>LED</b>	Light Emitting Diode, Leuchtdiode
<b>LQR</b>	Linear-quadratic-regulator
<b>Matlab</b>	Eine kommerzielle Software des Unternehmens The MathWorks zur Lösung mathematischer Probleme
<b>MIMO</b>	Multiple Input Multiple Output, Beschreibung für Mehrgrößensysteme
<b>PID</b>	proportional-integral-derivative

<b>PPM</b>	Puls-Pausen-Modulation
<b>SD-Karte</b>	Secure Digital Memory Card, sichere digitale Speicherkarte
<b>SDIO</b>	SD Input/Output, Schnittstelle zur Ansteuerung von SD-Karten
<b>SISO</b>	Single Input Single Output, Beschreibung für Eingrößensysteme
<b>SPI</b>	Serial Peripheral Interface, von der Firma Motorola entwickeltes Bussystem
<b>TXT</b>	Dateiendung von Textdateien unter Windows

# A Verwendete Messgeräte

Für die Messungen wurden folgende Geräte verwendet.

<b>Gerätetyp</b>	<b>Hersteller</b>	<b>Gerätename</b>	<b>Besitzer</b>
Analoges Oszilloskop	Philips	PM3232	Patrick Vogl
Oszilloskop Tastköpfe	Tektronix	TEK P6103	Patrick Vogl
Digitalmultimeter	Peaktech	3315 USB	Patrick Vogl
Digitalmultimeter	Hilix	DT-830B	Patrick Vogl
Optischer Drehzahlmesser	Lutron	DT2234C	Patrick Vogl
Küchenwaage	Soehnle	Siena	Patrick Vogl

## B Inhalt der CD

Der Quellcode sowie alle weiteren Dokumente, die nicht im Anhang zu finden sind, sind auf einer CD vorhanden, deren Einsicht bei Herrn Prof. Dr.-Ing. Florian Wenck möglich ist.

Der Inhalt der CD wird nachfolgend erläutert:

<b>Pfad</b>	<b>Beschreibung</b>
Datenblätter/	Alle verwendeten Datenblätter
Literatur/	Alle PDF-Dateien, die im Literaturverzeichnis angegeben sind
Matlab-Simulink/	Alle verwendeten Matlab-Simulationen
Motorregler/	Verwendete Motorreglereinstellungen
Quellcode/	Kompletter Quellcode in PDF-Form und als Projektordner
Schaltplan/	Aktualisierter Schaltplan des Motorreglers
Solidworks/	Erstellte 3D-CAD-Modelle in Solidworks
Masterthesis.pdf	Masterthesis in PDF-Form

# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach § 16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Tostedt, 31. März 2014

Ort, Datum

Unterschrift