

## Autonome Stabilisierung eines Quadropters

# Diplomarbeit

zur Erlangung des Grades Diplom Informatiker  
im Studiengang Informatik

vorgelegt von

**Simeon Maxein**

Erstgutachter: Prof. Dr. Dietrich Paulus  
Institut für Computervisualistik, Fachbereich 4

Zweitgutachter: Dr. Merten Joost  
Institut für integrierte Naturwissenschaften, Fachbereich 3

Koblenz, am 18. Mai 2012



# Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

- Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.
- Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

Koblenz, den .....



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>3</b>
<b>2. Grundlagen der Quadrocoptersteuerung</b>	<b>5</b>
2.1. Definition verwendeter Koordinatensysteme . . . . .	5
2.2. Dynamisches Verhalten eines Quadrocopters . . . . .	6
2.2.1. Beschreibung des Modells . . . . .	7
2.2.2. Rotorkräfte . . . . .	8
2.2.3. Schwerkraft . . . . .	15
2.2.4. Luftwiderstand . . . . .	15
2.2.5. Bewegungsgleichungen . . . . .	16
2.3. Sensoren zur Lagebestimmung . . . . .	25
2.3.1. Lagekreisel . . . . .	26
2.3.2. MEMS-Beschleunigungssensoren . . . . .	26
2.3.3. MEMS-Gyrosensoren . . . . .	27
2.4. Zustandsschätzung . . . . .	28
2.4.1. Kalman-Filter . . . . .	29
2.4.2. Partikelfilter . . . . .	32
2.4.3. Explizites komplementäres Filter . . . . .	33
2.5. Quaternionen . . . . .	34
2.5.1. Grundlagen . . . . .	34
2.5.2. Darstellung von Rotationen durch Quaternionen . . . . .	35
2.5.3. Rechenbeispiel . . . . .	37
2.5.4. Vergleich mit Rotationsmatrizen . . . . .	38
2.6. PID-Regler . . . . .	39
<b>3. Entwicklung einer Steuerplatine</b>	<b>43</b>
3.1. Aufbau der Platine . . . . .	44
3.2. Mikrocontroller: ATmega168 . . . . .	47
3.2.1. Verwendete Hardwaremodule . . . . .	47
3.3. Gyrosensor: Wii MotionPlus . . . . .	49
3.3.1. Aufbau und Funktionsweise . . . . .	51
3.3.2. Externe Schnittstellen . . . . .	52
3.3.3. Der I <sup>2</sup> C-Bus . . . . .	53

3.3.4.	Elektrische und mechanische Verbindung . . . . .	53
3.3.5.	Initialisierung und Aktivierung . . . . .	55
3.3.6.	Abfrage der Messwerte . . . . .	56
3.4.	Beschleunigungssensor: MMA7260QT . . . . .	58
3.4.1.	Elektrische und mechanische Verbindung . . . . .	58
3.4.2.	Abfrage der Beschleunigungswerte . . . . .	61
3.5.	Fernsteuerung: robbe Futaba F-14 . . . . .	64
3.5.1.	Anschluss und Auswertung des Empfängers . . . . .	64
3.6.	Antrieb . . . . .	66
3.6.1.	Antriebsmodule des X-Ufos . . . . .	66
3.6.2.	Leistungs-MOSFETs . . . . .	69
3.6.3.	Pulsweitenmodulation . . . . .	72
3.6.4.	Antriebssoftware . . . . .	77
3.7.	Zustandsfilter . . . . .	80
3.7.1.	Schätzung nur nach Winkelgeschwindigkeiten . . . . .	81
3.7.2.	Schätzung nach Winkelgeschwindigkeiten und gemessenen Richtungen . . . . .	82
3.7.3.	Driftausgleich . . . . .	84
3.7.4.	Besonderheiten der Implementierung . . . . .	84
3.8.	Ausrichtungsregler . . . . .	85
<b>4.</b>	<b>Fazit</b>	<b>87</b>
4.1.	Zusammenfassung . . . . .	87
4.2.	Vergleich mit verwandten Projekten . . . . .	88
4.2.1.	MikroKopter . . . . .	88
4.2.2.	MultiWii . . . . .	88
4.2.3.	AeroQuad . . . . .	89
4.3.	Ausblick . . . . .	89
<b>A.</b>	<b>Schaltplan und Platinenlayout</b>	<b>91</b>
	<b>Literaturverzeichnis</b>	<b>95</b>

# Abstract

Quadrokopter sind Helikopter mit vier in einer Ebene angeordneten Rotoren. Kleine unbemannte Modelle, die oft nur eine Schubkraft von wenigen Newton erzeugen können, sind im Spielzeug- und Modellbaubereich beliebt, werden aber auch von Militär und Polizei als Drohne für Aufklärungs- und Überwachungsaufgaben eingesetzt. Diese Diplomarbeit befasst sich mit den theoretischen Grundlagen der Steuerung eines Quadrokopters und entwickelt darauf aufbauend eine kostengünstige Steuerplatine für einen Modellquadrokopter.

Die theoretischen Grundlagen enthalten eine Untersuchung der Dynamik eines frei fliegenden Quadrokopters, bei der Bewegungsgleichungen hergeleitet und mit den Ergebnissen verglichen werden, die in „Design and control of quadrotors with application to autonomous flying“ ([Bou07]) vorgestellt wurden. Weiterhin wird die Funktionsweise verschiedener Sensoren beschrieben, die zur Bestimmung der aktuellen räumlichen Ausrichtung geeignet sind, und es werden Verfahren besprochen, mit denen die Ausrichtung aus den Messwerten dieser Sensoren abgeschätzt werden kann. Zusätzlich wird in den Schiefkörper der Quaternionen eingeführt, in dem dreidimensionale Rotationen kompakt dargestellt und effizient verkettet werden können.

Daran anschließend wird die Entwicklung einer einfachen Steuerplatine beschrieben, die sowohl einen autonomen Schwebeflug als auch Fernsteuerung ermöglicht. Die Platine wurde auf einem *X-Ufo*-Quadrokopter der Firma Silverlit entwickelt und getestet, der daher ebenfalls vorgestellt wird. Die eingesetzten Bauteile und deren Zusammenspiel werden besprochen. Dabei ist insbesondere die *Wii MotionPlus* hervorzuheben, die als kostengünstiges Gyrosensormodul verwendet wird. Daneben werden verschiedene Aspekte der Steuersoftware erläutert: die Auswertung der Sensordaten, die Zustandsschätzung mit Hilfe des expliziten komplementären Filters nach Mahony et al. ([MHP08]), die Umsetzung des Ausrichtungsreglers sowie die Erzeugung der Steuersignale für die Motoren.

Sowohl die Steuersoftware als auch Schaltplan und Platinenlayout der Steuerplatine liegen dieser Arbeit auf einer CD bei. Schaltplan und Platinenlayout sind zusätzlich im Anhang der Arbeit abgedruckt.



# 1. Einleitung

Quadrokopter werden heute in sehr unterschiedlichen Bereichen eingesetzt. Militär und Polizei nutzen diese Geräte zur mobilen Überwachung und Aufklärung (vgl. [Zie08]), Modellflugbegeisterte für Kunstflug oder als Plattform für Luftaufnahmen. Auch in der Forschung zu autonomen mobilen Systemen werden oft Quadrokopter eingesetzt (z.B. [MLD11a], popularisiert durch das YouTube-Video [MLD11b]).

Aufgrund der Faszination, die von Fluggeräten allgemein ausgeht, eignen sich Quadrokopter auch als Anschauungsobjekt in Lehrveranstaltungen. Sie können z.B. zur Besprechung von Reglern eingesetzt werden, da ein Quadrokopter in der üblichen Bauweise nur mit Hilfe eines Ausrichtungsreglers stabil schweben kann. Dabei kann nicht nur der Ausgleich von Störeinflüssen durch den Regler demonstriert werden, sondern auch das Verhalten eines schlecht eingestellten Reglers, das sich durch Überschwingen und Aufschaukeln äußern kann.

Durch den Fortschritt der Technik, insbesondere im Bereich miniaturisierter Sensor-ICs, ist es heute kostengünstig möglich, die Steuerung für einen Quadrokopter selbst zu bauen. Die zentrale Aufgabe einer solchen Steuerung ist die Erfassung und Stabilisierung der Ausrichtung des Quadrokopters. Weitere Komponenten der Steuerung, z.B. ein Fernsteuerungsempfänger, können dabei eine Soll-Ausrichtung vorgeben.

Diese Arbeit befasst sich mit der Entwicklung einer Quadrokoptersteuerung, die eine solche Stabilisierung ermöglicht. Neben einer Behandlung theoretischer Grundlagen wird eine einfache Steuerplatine entwickelt, die mit Hilfe geeigneter Sensoren einen autonomen stabilen Schwebeflug realisiert und Steuervorgaben zu Ausrichtung und Schub umsetzen kann.

Die Platine kann als Grundlage für weitere Projekte verwendet werden, z.B. für die Untersuchung unterschiedlicher Regelstrategien, oder zur autonomen Durchführung von Flugmanövern. Dabei wurde das Erreichen idealer Flugeigenschaften als weniger wichtig behandelt als eine gute Verständlichkeit von Hard- und Software. Zusätzlich wurde Wert darauf gelegt, dass die vorgestellte Steuerplatine kostengünstig und einfach nachgebaut werden kann.



## 2. Grundlagen der Quadrokooptersteuerung

Dieses Kapitel befasst sich mit theoretischen Grundlagen zur Erfassung und Stabilisierung der Ausrichtung eines Quadrokoopters. Um eine konsistente Darstellung von Koordinaten und Richtungen zu ermöglichen, definiert Abschnitt 2.1 die in dieser Arbeit verwendeten Koordinatensysteme.

In Abschnitt 2.2 wird das Bewegungsverhalten des Quadrokoopters unter dem Einfluss verschiedener Kräfte genauer untersucht, um einen Einblick in das zu stabilisierende System zu erhalten. Abschnitt 2.3 gibt daraufhin eine Übersicht über die Funktionsweise verschiedener Sensoren, die zur Messung der Ausrichtung verwendet werden können. Die Messwerte von Sensoren sind allerdings oft ungenau und verrauscht. Abschnitt 2.4 beschreibt verschiedene Verfahren, die aus solchen ungenauen Messwerten die aktuelle Ausrichtung abschätzen können.

Bei der Umsetzung einer Steuersoftware müssen Berechnungen mit Rotationen und Ausrichtungen durchgeführt werden. Dazu werden oft Quaternionen eingesetzt, da diese im Vergleich zu Rotationsmatrizen weniger Speicher benötigen und eine effizientere Verkettung von Rotationen ermöglichen. Daher führt Abschnitt 2.5 in dieses Thema ein.

### 2.1. Definition verwendeter Koordinatensysteme

Mit dem Begriff „erdfestes Koordinatensystem“ wird in dieser Arbeit ein rechtshändiges Koordinatensystem bezeichnet, dessen Achsen der in der Luftfahrt üblichen NED (North-East-Down)-Konvention folgen (vgl. [Wik12f]). Die  $x$ -Achse zeigt dabei nach Norden, die  $y$ -Achse nach Osten, und die  $z$ -Achse in Richtung der Schwerkraft. Das „körperfeste Koordinatensystem“ des Quadrokoopters ist ebenfalls als rechtshändig definiert und folgt der Konvention für Luftfahrzeuge, bei der die  $x$ -Achse zur Vorderseite des Fluggeräts zeigt, die  $y$ -Achse zur rechten Seite und die  $z$ -Achse zur Unterseite. Um Verwirrung aufgrund der Symmetrie eines Quadrokoopters zu vermeiden, wurde einer der Rotoren des X-Ufos als vorderer Rotor ausgewählt und mit einer roten Markierung versehen.

Die  $x$ -Achse des Quadrokoopters wird auch als Rollachse bezeichnet, die  $y$ -Achse als Nickachse, und die  $z$ -Achse als Gierachse. Da ein rechtshändiges Koordinatensystem verwendet wird, dreht eine positive Nickbewegung die Vorderseite des Quadrokoopters nach oben. Eine positi-

## 2. Grundlagen der Quadroptersteuerung

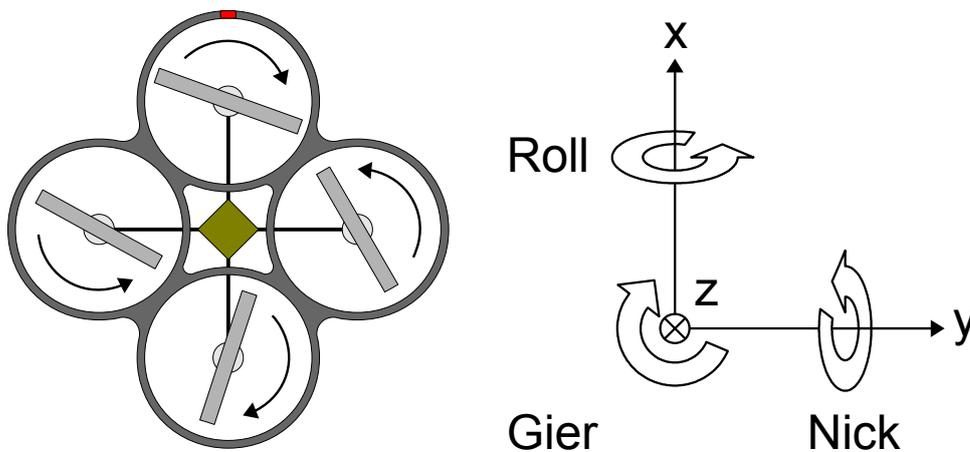


Abbildung 2.1.1.: Darstellung des Quadropter-Koordinatensystems. Der Quadropter ist von oben gezeigt, die Vorderseite ist mit einer roten Markierung festgelegt. Zusätzlich ist die Drehrichtung der Rotoren des Quadropters „X-Ufo“ eingezeichnet.

ve Rollbewegung dreht die linke Seite nach oben und eine positive Gierbewegung dreht den Quadropter von oben betrachtet im Uhrzeigersinn.

### 2.2. Dynamisches Verhalten eines Quadropters

Dieser Abschnitt beschäftigt sich mit der Dynamik von Quadroptern und beschreibt die wichtigsten Kräfte und Drehmomente, die im freien Flug auf einen solchen wirken. Um den Rahmen dieser Arbeit nicht zu übersteigen, wird auf eine eingehende Behandlung der aerodynamischen Aspekte verzichtet<sup>1</sup>. Statt dessen liegt der Schwerpunkt auf der Frage, wie sich verschiedene Kräfte auf die Bewegung des Quadropters auswirken, und es werden Bewegungsgleichungen hergeleitet, die diese Zusammenhänge beschreiben.

Als Grundlage dient ein Modell, welches in [Bou07] vorgestellt wurde und wiederum auf den Ergebnissen des Mesicopter-Projekts basiert ([Fay01]). Aus diesen Quellen wurden grundlegende Annahmen zum Aufbau des Quadropters sowie Informationen zu den Kräften übernommen, die auf die Rotoren wirken. Die darauf aufbauende Herleitung der Bewegungsgleichungen wurde jedoch unabhängig durchgeführt.

Der folgende Unterabschnitt beschreibt zunächst den Aufbau eines Quadropters, der in den folgenden Überlegungen vorausgesetzt wird. Daraufhin werden die äußeren Kräfte und Drehmomente erläutert, die auf Propeller und Rahmen wirken. Für die Drehmomente wird jeweils

<sup>1</sup> Eine detailliertere Einführung in die Aerodynamik von Hubschraubern findet sich z.B. in [Lei06].

eine einheitliche Darstellung mit Bezug auf den Schwerpunkt des Quadropters hergeleitet. Es wird stets das lokale Koordinatensystem des Quadropters verwendet. Schließlich wird aus der Gesamtformel aller äußeren Kräfte und Drehmomente die Beschleunigung und Winkelbeschleunigung des Rahmens berechnet. Das Ergebnis wird mit den in [Bou07] angegebenen Formeln verglichen.

### 2.2.1. Beschreibung des Modells

Der mechanische Aufbau eines Quadropters ist deutlich einfacher als der eines bemannten Helikopters in der verbreiteten Bauform mit Haupt- und Heckrotor, da ein solcher Helikopter zum Erzeugen von Drehmoment um die Roll- und Nickachse den Anstellwinkel der Rotorblätter positionsabhängig ändern muss. Ein Quadropter kann dagegen allein durch Variation der Rotationsgeschwindigkeit der vier starren Rotoren gesteuert werden.

Ein Quadropter kann daher als System aus fünf starren Körpern modelliert werden. Diese sind der Rahmen mit allen darauf befestigten Massen, z.B. Batterien und Elektronik, und die vier Rotoren. Es wird angenommen, dass der Rahmen symmetrisch aufgebaut ist, so dass die lokalen Koordinatenachsen den Hauptträgheitsachsen des Rahmens entsprechen.

Die Rotoren sind so mit dem Rahmen verbunden, dass sie um die lokale z-Achse gedreht werden können und damit genau einen Freiheitsgrad gegen den Rahmen besitzen. Sie werden nach ihrer Position mit den Buchstaben  $V$ ,  $H$ ,  $L$  und  $R$  (vorne, hinten, links und rechts) bezeichnet. Als Schwerpunkt jedes Rotors wird der Punkt angenommen, an dem die Rotorblätter zusammentreffen. Vertikal befinden sich die Rotorschwerpunkte in einer Höhe von  $h$  über dem Gesamtschwerpunkt des Quadropters, horizontal in einer Entfernung  $l$  von diesem. Die relative Position der Rotoren zum Gesamtschwerpunkt ist damit:

$$\vec{r}_V = \begin{pmatrix} l \\ 0 \\ -h \end{pmatrix}, \quad \vec{r}_H = \begin{pmatrix} -l \\ 0 \\ -h \end{pmatrix}, \quad \vec{r}_L = \begin{pmatrix} 0 \\ -l \\ -h \end{pmatrix}, \quad \vec{r}_R = \begin{pmatrix} 0 \\ l \\ -h \end{pmatrix} \quad (2.2.1)$$

Der vordere und der hintere Rotor sind so ausgelegt, dass sie dem linken und rechten Rotor entgegengesetzt rotieren. Die Entscheidung, welches Rotorpaar in welche Richtung rotiert, ist zunächst – wie auch die Festlegung, welcher der Rotoren der vordere ist – vollkommen willkürlich. Der in dieser Arbeit verwendete Quadropter „X-Ufo“ folgt der Konvention, dass der vordere Rotor positiv um die z-Achse gedreht wird<sup>2</sup>, weshalb diese Konvention auch für das Modell verwendet wird. Die gegensätzliche Drehrichtung ist notwendig, damit das später beschriebene Bremsmoment zwischen den Rotoren ausgeglichen werden kann. Abgesehen von der unterschiedlichen Solldrehrichtung sind die Rotoren identisch aufgebaut.

<sup>2</sup>In anderen Worten: Der vordere und der hintere Rotor drehen sich von oben betrachtet im Uhrzeigersinn, der linke und der rechte Rotor gegen den Uhrzeigersinn.

## 2. Grundlagen der Quadroptersteuerung

Um die unterschiedliche Drehrichtung der Rotoren kompakt darstellen zu können, wird  $dir_i$  als das Vorzeichen der bevorzugten Drehrichtung des Rotors  $i \in \{V, H, L, R\}$  um die z-Achse definiert:

$$dir_i = \begin{cases} 1 & \text{falls } i \in \{V, H\} \\ -1 & \text{falls } i \in \{L, R\} \end{cases} \quad (2.2.2)$$

### 2.2.2. Rotorkräfte

Auf jeden der vier Rotoren wirken Kräfte und Drehmomente durch die Wechselwirkung mit der umgebenden Luft. Die folgenden Abschnitte beschreiben eine Aufteilung in zwei Kräfte und zwei Momente für jeden Rotor, bezogen auf dessen Schwerpunkt. Die wichtigsten Ursachen dieser Kräfte werden erläutert. In jedem Abschnitt wird daraufhin ein Kraftvektor und ein Drehmomentvektor mit Bezug auf den Gesamtschwerpunkt des Quadropters hergeleitet, der die besprochene Kraft für alle vier Rotoren beschreibt.

Da interne Kräfte (z.B. durch die Reibung der Motorlager oder die Beschleunigung der Rotoren gegen den Rahmen) nur zwischen verschiedenen Teilkörpern des Quadropters wirken, ändern sie aufgrund der Impuls- bzw. Drehimpulserhaltung nicht den Gesamtimpuls bzw. Gesamtdrehimpuls des Quadropters als System aller Teilkörper. Daher werden diese Kräfte hier noch nicht berücksichtigt, sondern fließen erst in Unterabschnitt 2.2.5 in die Bewegungsgleichungen ein.

Die auf den Rotor wirkenden Kräfte werden in die Schubkraft und die Horizontalkraft aufgeteilt, wobei die Schubkraft parallel zur Rotorachse wirkt, die Horizontalkraft senkrecht zu dieser. Analog dazu erfolgt auch die Aufteilung des auf den Rotor wirkenden Drehmoments in das Bremsmoment, das parallel zur Achse ausgerichtet ist und der Drehung des Rotors entgegenwirkt, und das Rotor-Rollmoment, das um eine Achse senkrecht zur Rotorachse wirkt.

#### Schub ( $T$ )

Die Schubkraft  $T$  (thrust) ist die Summe aller vertikalen Kraftkomponenten, die auf einen Rotor wirken. Bei vorgesehener Drehrichtung wirkt der Schub jedes Rotors nach oben, oder genauer: entgegen der körperfesten z-Achse. Der Schub wird dadurch erzeugt, dass die Rotorblätter Luft nach unten beschleunigen.

In [Bou07] wird als Annahme vorausgesetzt, dass der Schub quadratisch mit der Drehzahl des Rotors steigt. Neben der Drehzahl wird der Schub jedoch auch von weiteren Größen beeinflusst, von denen einige nur vom verwendeten Rotor abhängen, andere dagegen von der Umgebung oder dem aktuellen Bewegungszustand.

Zu den Rotoreigenschaften gehören Radius, Form und Anzahl der Blätter, sowie der Winkel, in dem diese Blätter angebracht sind. Als Umgebungseigenschaft wirkt sich die Dichte der Luft

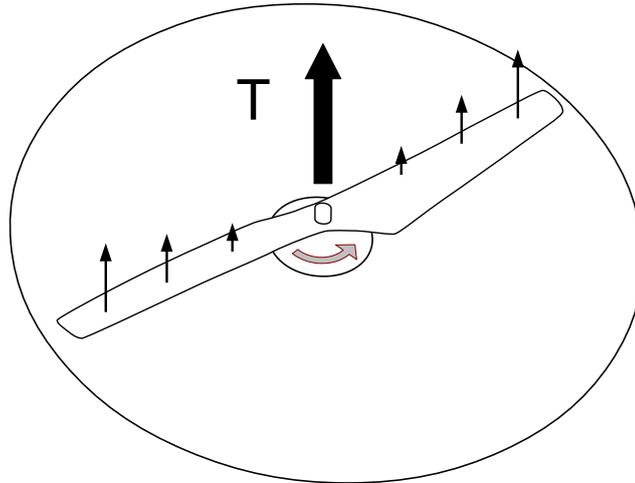


Abbildung 2.2.1.: Entstehung der Schubkraft. Durch die Drehbewegung des Rotors (grauer Pfeil) wird Luft nach unten beschleunigt. Dadurch entstehen am Rotor nach oben gerichtete Kräfte (kleine schwarze Pfeile). Die Summe der vertikalen Teilkräfte ist die Schubkraft  $T$  (großer schwarzer Pfeil).

aus, welche wiederum von Luftdruck, Luftfeuchtigkeit und Temperatur abhängt. Bei höherer Luftdichte beschleunigt ein Rotor eine größere Luftmasse bei gleichem Volumen, so dass ein größerer Schub erzeugt wird.

Der Schub wird auch durch die Bewegung des Rotors relativ zur umgebenden Luft beeinflusst. Wenn sich der Rotor in Richtung seiner Achse bewegt, z.B. bei einem schnell aufsteigenden Helikopter, ändert sich der Winkel, in dem die Luft auf die drehenden Rotorblätter trifft. Bei hohen Geschwindigkeiten sinkt der erzeugte Schub oder wird sogar negativ, wenn der Rotor sich im Verhältnis zur Geschwindigkeit so langsam dreht, dass er wie ein Windrad vom Luftstrom beschleunigt wird.

Material in der Umgebung des Rotors kann sich ebenfalls auf den Schub auswirken. Ein wichtiges Beispiel hierfür ist der Bodeneffekt, welcher den Schub eines Rotors erhöht, wenn dieser sich knapp über dem Boden befindet, was insbesondere bei Start und Landung der Fall ist. Dies kann dadurch erklärt werden, dass der Luftstrom durch den Rotor in Bodennähe abgebremst wird, so dass die Luft in einem steileren Winkel auf die Rotorblätter trifft. Die Stärke des Bodeneffekts sinkt mit zunehmender Flughöhe und trägt bei Quadroptern ab einer Flughöhe von ca. einem Rotordurchmesser nicht mehr deutlich zum Schub bei. Dies kann in einer Messung in [Bou07, Seite 22] beobachtet werden.

Es sollte beachtet werden, dass der Bodeneffekt in geschlossenen Räumen auch in der Nähe der Decke wirkt. Da der Schub in diesem Fall ebenfalls nach oben verstärkt wird, ergibt sich jedoch ein selbstverstärkender Effekt, bei dem der Quadropters an die Decke „gesaugt“ wird. Daher ist es empfehlenswert, einen entsprechenden Abstand zur Decke einzuhalten.

## 2. Grundlagen der Quadroptersteuerung

Die in [Bou07] für den Schub  $T$  angegebene Formel lautet:

$$T = C_T \rho A (\omega r)^2$$

Dabei ist  $C_T$  der Schubbeiwert, ein einheitenloser Faktor, welcher von den oben angesprochenen Rotoreigenschaften sowie der Bewegung des Rotors relativ zur umgebenden Luft abhängt.  $\rho$  ist die Luftdichte,  $A$  die Fläche des gedachten Kreises, der von den Rotorblättern überstrichen wird.  $\omega$  ist die Winkelgeschwindigkeit des Rotors in  $rad/s$  und  $r$  ist der Radius des Rotors. Dies entspricht auch der Definition des Schubbeiwerts in [Lei06, Seite 729]. Der einzige Unterschied besteht in einem konstanten Faktor von 2, der vermutlich auf die Verwendung unterschiedlicher Konventionen zurückzuführen ist.

Da sich die Fläche  $A$  direkt aus  $r$  ergibt, kann die Formel auch zu  $T = C_T \rho \omega^2 \pi r^4$  umgestellt werden. Die gezeigte Form wurde vermutlich deshalb gewählt, weil sie eine bestimmte Interpretation der Variablen nahelegt. So ist  $\omega r$  die lineare Geschwindigkeit der Rotorblattspitzen.  $A$  kann als Referenzfaktor für den Schubbeiwert betrachtet werden, der in der Definition von  $C_T$  benötigt wird, um eine einheitenlose Zahl zu erhalten, die unter Rotoren unterschiedlicher Größe verglichen werden kann.

Die Schubkräfte wirken entgegen der Richtung der lokalen z-Achse des Quadropters. Daher entspricht die Schubkraft jedes Rotors  $i \in \{V, H, L, R\}$  als Vektor  $\vec{T}_i = (0 \ 0 \ -T_i)^T$ . Der Gesamtschub entspricht der Summe dieser Kräfte. Da die Kräfte nicht im Quadropterschwerpunkt ansetzen, verursachen sie zusätzlich ein Drehmoment im Bezug auf diesen, welches von der relativen Position der Rotorschwerpunkte zum Schwerpunkt des Quadropters abhängt:

$$\vec{M}_{T,i} = \vec{r}_i \times \vec{T}_i$$

Setzt man für alle Rotoren die in Gleichung 2.2.1 angegebenen Rotorschwerpunkte für  $\vec{r}_i$  ein, erhält man das gesamte, auf den Quadropterschwerpunkt bezogene Schubdrehmoment  $\vec{M}_T$ :

$$\begin{aligned} \vec{M}_T &= \sum_{i \in \{V, H, L, R\}} \vec{r}_i \times \vec{T}_i \\ &= \begin{pmatrix} l \\ 0 \\ -h \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ -T_V \end{pmatrix} + \begin{pmatrix} -l \\ 0 \\ -h \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ -T_H \end{pmatrix} + \begin{pmatrix} 0 \\ -l \\ -h \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ -T_L \end{pmatrix} + \begin{pmatrix} 0 \\ l \\ -h \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ -T_R \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ l \cdot T_V \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -l \cdot T_H \\ 0 \end{pmatrix} + \begin{pmatrix} l \cdot T_L \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -l \cdot T_R \\ 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} l \cdot (T_L - T_R) \\ l \cdot (T_V - T_H) \\ 0 \end{pmatrix} \end{aligned} \tag{2.2.3}$$

Wenn der Schub des linken Rotors höher ist als der des rechten Rotors, entsteht also ein positives Drehmoment um die  $x$ -Achse. Analog dazu entsteht ein positives Drehmoment um die  $y$ -Achse, wenn der vordere Rotor einen höheren Schub erzeugt als der hintere. Dies deckt sich mit der anschaulichen Vorstellung und ermöglicht eine gezielte Beeinflussung von Nick- und Rollbeschleunigung.

Die Gesamtschubkraft  $\vec{F}_T$  ist die Summe der Schubkräfte aller Rotoren:

$$\vec{F}_T = \begin{pmatrix} 0 \\ 0 \\ -\sum_i(T_i) \end{pmatrix} \quad (2.2.4)$$

### Rollmoment ( $R$ )

Wie der Schub entsteht auch das Rollmoment aus den vertikalen Kräften, die auf den Rotor wirken. Unter idealisierten Bedingungen sind diese Kräfte punktsymmetrisch über den Rotor verteilt, so dass sich im Bezug auf den Rotorschwerpunkt lediglich eine lineare Schubkraft ergibt. Wenn der Rotor sich jedoch seitwärts durch die Luft bewegt, dann ist die Geschwindigkeit der Rotorblätter relativ zur Luft auf einer Seite des Rotors höher als auf der anderen, wodurch auch die vertikalen Kräfte auf einer Seite höher sind (vgl. Abbildung 2.2.2). Durch dieses Ungleichgewicht entsteht im Rotorschwerpunkt ein Drehmoment, das in [Bou07] als Rollmoment bezeichnet wird. Trotz dieser Bezeichnung wirkt das Rotor-Rollmoment nicht nur entlang der Rollachse, sondern ist ein Drehmomentvektor auf der Ebene von Roll- und Nickachse.

Das Rotor-Rollmoment setzt sich nach der oben gegebenen Beschreibung aus Kräften zusammen, die vertikal, also parallel zur Achse des Rotors wirken. Jede dieser Kräfte erzeugt daher ein Drehmoment auf den Rotor, dessen Richtung senkrecht zur Achse steht. Die Summe dieser Momente, und damit das gesamte Rotor-Rollmoment, steht daher ebenfalls senkrecht zur Achse des Rotors. Wenn die Geschwindigkeit des Quadropters relativ zur Luft die einzige Ursache für das Rollmoment ist, kann angenommen werden, dass die Richtung des Drehmomentvektors ungefähr der Projektion des Vektors der Luftgeschwindigkeit auf die  $xy$ -Ebene entspricht. Anders ausgedrückt entsteht bei einem Vorwärtsflug des Quadropters ein Drehmoment entlang der Rollachse, bei einem Seitwärtsflug entlang der Nickachse. Dies muss jedoch nicht exakt zutreffen, da die genaue Verteilung der Kräfte am Rotor von dessen Form abhängt.

Die Richtung des Rollmoments durch Luftgeschwindigkeit hängt offensichtlich von der Drehrichtung des betrachteten Rotors ab, der einzige Unterschied liegt allerdings im Vorzeichen. Um dies in der zusammengefassten Darstellung zum Ausdruck zu bringen, kann das Rollmoment  $\vec{R}_i$  jedes Rotors  $i$  so berechnet werden, als würde er sich in die gleiche Richtung drehen wie der vordere Rotor. Die tatsächliche Drehrichtung wird erst in der zusammengefassten For-

## 2. Grundlagen der Quadroptersteuerung

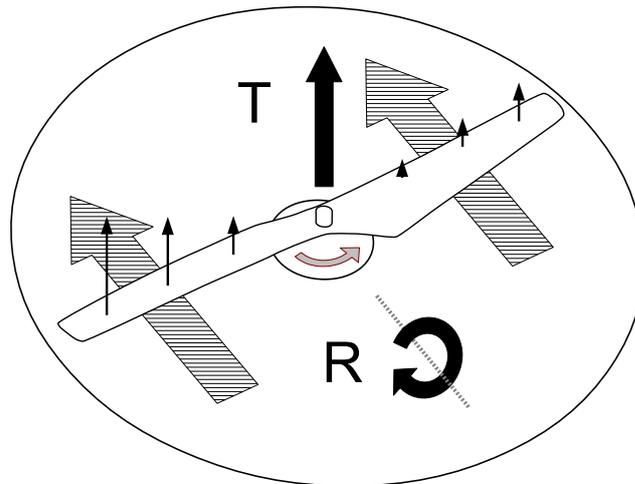


Abbildung 2.2.2.: Zu der Situation in Abbildung 2.2.1 wird ein Seitenwind hinzugefügt (schraffierte Pfeile). Als Folge sind die Teilkräfte nun auf der Seite des Rotors niedriger, auf der sich das Rotorblatt in Windrichtung bewegt, auf der anderen Seite sind die Kräfte größer. Dadurch entsteht das Rollmoment  $R$ , dessen Achse grob der Richtung des Seitenwinds entspricht.

mel berücksichtigt:

$$\vec{M}_R = \sum_i (\vec{R}_i \cdot dir_i) = \begin{pmatrix} \sum_i (R_{i,x} \cdot dir_i) \\ \sum_i (R_{i,y} \cdot dir_i) \\ 0 \end{pmatrix} \quad (2.2.5)$$

### Bremsmoment ( $D$ )

Auf die Rotoren wirken nicht nur vertikale, sondern auch horizontale Kräfte. Für eine genauere Betrachtung werden sie hier in das Bremsmoment und die Horizontalkraft aufgeteilt, die im nächsten Abschnitt beschrieben wird.

Das Bremsmoment  $D$  (drag) entsteht als unvermeidbare Folge des Funktionsprinzips eines Rotors, da diesem die zur Erzeugung des Schubs notwendige Leistung über das Drehmoment des Motors zugeführt wird. Wenn sich der Drehimpuls des Rotors nicht ändert, dann muss der Rotor dieses Drehmoment an die Luft weitergeben, so dass andersherum betrachtet ein der Drehrichtung des Motors entgegengesetztes Drehmoment von der Luft auf den Rotor wirkt. In der Realität ist das Bremsmoment aufgrund aerodynamischer Verluste jedoch höher, als es aus rein energetischen Betrachtungen mindestens sein muss.

Da die Richtung des Bremsmoments von der Drehrichtung des Rotors abhängt, haben die Bremsmomente des vorderen und hinteren Rotors ein anderes Vorzeichen als die des linken und rechten Rotors. Da sich der vordere Rotor im Modell positiv um die z-Achse dreht, wirkt das Bremsmoment dieses Rotors negativ um diese Achse.

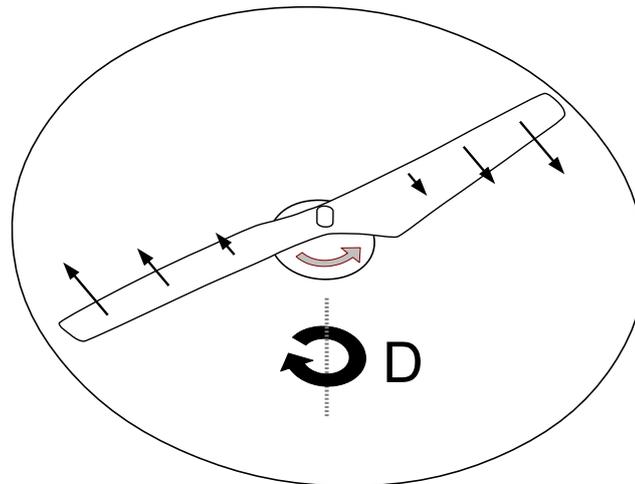


Abbildung 2.2.3.: Entstehung des Bremsmoments. Die Bewegung der Rotorblätter durch die Luft erzeugt der Bewegung entgegengesetzte Kräfte, welche sich im Idealfall zu einem Nullvektor addieren und so lediglich ein Drehmoment hervorrufen, das der Rotation des Rotors entgegenwirkt.

Für alle Rotoren zusammen gilt daher:

$$\vec{M}_D = \begin{pmatrix} 0 \\ 0 \\ -\sum_i (D_i \cdot dir_i) \end{pmatrix} \quad (2.2.6)$$

### Horizontalkraft ( $H$ )

Im Idealfall sind die horizontalen Kraftvektoren an zwei gegenüberliegenden Punkten des Rotors genau entgegengesetzt, so dass sich die Kräfte zu einem Vektor der Länge 0 addieren. Allerdings kann es ähnlich wie bei den vertikalen Kräften vorkommen, dass die Verteilung der horizontalen Kräfte nicht gleichmäßig ist. So entsteht in der Summe eine lineare horizontale Kraft, die hier Horizontalkraft genannt wird.

Eine Ursache für eine ungleichmäßige Verteilung der Kräfte liegt wie auch beim Rollmoment in einer Bewegung des Quadropters relativ zur umgebenden Luft. Da die Geschwindigkeit der Rotorblätter gegenüber der Luft nun auf einer Seite des Rotors höher ist als auf der anderen, sind auf dieser Seite auch die horizontalen Kräfte größer, die durch die Ablenkung der Luft und den Luftwiderstand auf den Rotor wirken. In der Summe ergibt sich dadurch eine Kraft, die ungefähr in Richtung des Seitenwinds wirkt, wobei auch hier die Form des Rotors die tatsächliche Richtung und Stärke der Kraft beeinflusst.

Eine horizontale Kraft am Schwerpunkt eines Rotors wirkt sich im Bezug auf den Quadropterschwerpunkt auch als Drehmoment aus. Die Formel dafür lautet analog zum Drehmoment

## 2. Grundlagen der Quadroptersteuerung

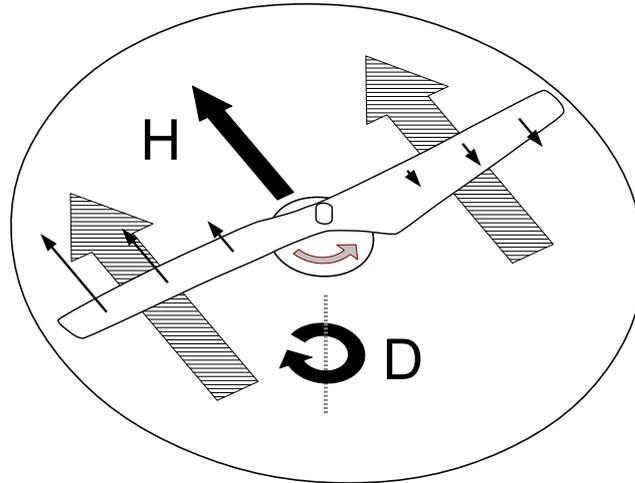


Abbildung 2.2.4.: Entstehung der Horizontalkraft. Durch einen Seitenwind entsteht ein Ungleichgewicht der horizontalen Kräfte, so dass in der Summe eine lineare horizontale Kraft entsteht.

durch die Schubkraft:

$$\vec{M}_{H,i} = \vec{r}_i \times \vec{H}_i$$

Setzt man die Rotorschwerpunkte aus Gleichung 2.2.1 für  $\vec{r}_i$  ein, ergibt sich für alle Rotoren zusammen:

$$\begin{aligned} \vec{M}_H &= \sum_{i \in \{V,H,L,R\}} \vec{r}_i \times \vec{H}_i \\ &= \begin{pmatrix} l \\ 0 \\ -h \end{pmatrix} \times \begin{pmatrix} H_{V,x} \\ H_{V,y} \\ 0 \end{pmatrix} + \begin{pmatrix} -l \\ 0 \\ -h \end{pmatrix} \times \begin{pmatrix} H_{H,x} \\ H_{H,y} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ -l \\ -h \end{pmatrix} \times \begin{pmatrix} H_{L,x} \\ H_{L,y} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ l \\ -h \end{pmatrix} \times \begin{pmatrix} H_{R,x} \\ H_{R,y} \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} h \cdot \sum_i (H_{i,y}) \\ -h \cdot \sum_i (H_{i,x}) \\ l \cdot (H_{V,y} - H_{H,y} + H_{L,x} - H_{R,x}) \end{pmatrix} \end{aligned} \quad (2.2.7)$$

$$\vec{F}_H = \begin{pmatrix} \sum_i (H_{i,x}) \\ \sum_i (H_{i,y}) \\ 0 \end{pmatrix} \quad (2.2.8)$$

Eine gleichmäßige Horizontalkraft in Richtung der x-Achse erzeugt also ein negatives Drehmoment um die y-Achse, wenn sich die Rotoren oberhalb des Gesamtschwerpunkts befinden, während eine gleichmäßige Horizontalkraft in Richtung der y-Achse ein positives Drehmoment um die x-Achse auslöst.

Ein Drehmoment um die  $z$ -Achse entsteht nur, wenn eine Kraftdifferenz zwischen den gegenüberliegenden Rotoren besteht: Wirkt zum Beispiel auf den vorderen Rotor eine Kraft nach rechts und auf den hinteren eine Kraft nach links ( $H_{V,y} > 0, H_{H,y} < 0$ ), dann entsteht (von oben betrachtet) ein Drehmoment im Uhrzeigersinn, d.h. in positiver Richtung um die  $z$ -Achse.

### 2.2.3. Schwerkraft

Die Schwerkraft wirkt gleichmäßig auf alle Massepunkte des Rahmens und der Rotoren und erzeugt daher lediglich eine Beschleunigung in Richtung der  $z$ -Achse des erdfesten Koordinatensystems, ohne die Rotation zu beeinflussen. Da alle Kräfte im körperfesten Koordinatensystem dargestellt werden sollen, muss die Kraft entsprechend transformiert werden.

Wenn  $R_k$  eine Rotationsmatrix ist, die Vektoren vom erdfesten in das körperfeste Koordinatensystem überführt, kann die Schwerkraft wie folgt dargestellt werden:

$$\vec{F}_g = R_k \cdot \begin{pmatrix} 0 \\ 0 \\ m \cdot g \end{pmatrix} = \begin{pmatrix} R_{k,31} \cdot mg \\ R_{k,32} \cdot mg \\ R_{k,33} \cdot mg \end{pmatrix} \quad (2.2.9)$$

$m$  ist die Masse des gesamten Quadropters,  $g$  ist die Erdbeschleunigung,  $R_{k,31}$ ,  $R_{k,32}$  und  $R_{k,33}$  sind die Komponenten der dritten Spalte von  $R_k$ .

### 2.2.4. Luftwiderstand

Die aerodynamischen Eigenschaften der Rotoren sind bereits berücksichtigt, der Rahmen besitzt jedoch auch einen Luftwiderstand, welcher sich im Prinzip auf jede Bewegung des Quadropters durch die Luft auswirkt. In Anlehnung an das Modell in [Bou07] wird der Luftwiderstand des Rahmens hier nur vereinfacht als lineare Kraft betrachtet, welche anhand der Formel für den Strömungswiderstand berechnet wird (vgl. [Dem94, Seite 226]):

$$|F_W| = c_w \frac{\rho}{2} v^2 A$$

Dabei ist  $\rho$  die Luftdichte,  $v$  die Geschwindigkeit des Körpers relativ zur Luft,  $c_w$  der Widerstandsbeiwert, welcher von Form und Anströmrichtung des Körpers abhängt, und  $A$  eine Referenzfläche.

Um statt dem Betrag einen Kraftvektor zu erhalten kann man als grobe Näherung annehmen, dass der Luftwiderstand der Bewegung direkt entgegenwirkt. Dies kann in der Formel ausgedrückt werden, indem  $v^2$  durch  $-\vec{v} \cdot |\vec{v}|$  ersetzt wird. Damit ergibt sich für den Luftwiderstand bei einer Bewegung in beliebige Richtungen:

$$\vec{F}_W \approx -c_w \frac{\rho}{2} \vec{v} |\vec{v}| A \quad (2.2.10)$$

### 2.2.5. Bewegungsgleichungen

Die oben aufgeführten äußeren Drehmomente beschreiben die Änderung des Gesamtdrehimpulses des Quadropters bezogen auf dessen Schwerpunkt. Um die tatsächliche Bewegung des Quadropters zu ermitteln, müssen allerdings die Winkelgeschwindigkeiten und Winkelbeschleunigungen betrachtet werden. Dazu wird im Folgenden zunächst der Zusammenhang zwischen den Winkelgeschwindigkeiten des Quadropters und seinem Gesamtdrehimpuls hergestellt. Durch Ableitung wird daraus der Zusammenhang zwischen Winkelbeschleunigungen und äußerem Drehmoment ermittelt. Schließlich werden die linearen Beschleunigungen hinzugezogen, um Bewegungsgleichungen für Rotation und Translation zu erhalten.

#### Trägheitstensor

Ein Trägheitstensor ist eine  $3 \times 3$ -Matrix, welche die Rotationsträgheit eines starren Körpers beschreibt. Der Tensor nimmt für die Rotationsbewegung eine Rolle ein, die mit der trägen Masse in linearen Bewegungen vergleichbar ist. Während bei linearen Bewegungen der Zusammenhang zwischen Impuls  $\vec{p}$  und Geschwindigkeit  $\vec{v}$  über die Masse  $m$  durch die bekannte Formel  $\vec{p} = m\vec{v}$  gegeben ist, stellt der Trägheitstensor  $I$  für einen starren Körper den Zusammenhang zwischen dem Drehimpuls  $\vec{L}$  und der Winkelgeschwindigkeit  $\vec{\omega}$  her (vgl. [Dem94, Seite 136]):

$$\vec{L} = I\vec{\omega} \quad (2.2.11)$$

Für die Darstellung der Rotationsträgheit reicht eine skalare Größe nicht aus, da die Rotation um verschiedene Achsen eines starren Körpers unterschiedlich träge sein kann. Ein Besenstiel kann sehr einfach um seine Längsachse gedreht werden, die Rotation um die Querachsen ist dagegen deutlich träger.

Die Trägheit der Rotation um eine feste Achse wird als Trägheitsmoment bezeichnet. Die Achsen des größten und des kleinsten Trägheitsmoments eines starren Körpers stehen immer senkrecht zueinander und bilden zusammen mit einer weiteren Achse senkrecht zu diesen beiden die Hauptträgheitsachsen des Körpers.<sup>3</sup>

In einem Koordinatensystem entlang der Hauptträgheitsachsen eines Körpers ist der Trägheitstensor eine diagonale Matrix. Die drei diagonalen Elemente entsprechen dann dem Trägheitsmoment der Rotation um die drei Hauptträgheitsachsen. Aus der oben angegebenen Formel  $\vec{L} = I\vec{\omega}$  ist daher ersichtlich, dass der Drehimpuls bei einer Rotation um eine Hauptträgheitsachse parallel zur Winkelgeschwindigkeit verläuft.

Bei einer Rotation um eine beliebige Achse, die keine Hauptträgheitsachse des Körpers ist, weichen die Richtungen von Drehimpuls und Winkelgeschwindigkeit jedoch voneinander ab.

<sup>3</sup>Im Fall des Besenstiels ist nur die Achse des kleinsten Trägheitsmoments eindeutig. Aus Symmetriegründen ist beim Besenstiel jede Querachse eine Hauptträgheitsachse.

Dadurch ändert sich aus Sicht des körperfesten Koordinatensystems die Richtung des Drehimpulses über die Zeit, was zu einer Änderung der Rotationsachse führt. Der Körper führt daher – ohne äußere Krafteinwirkung – eine komplizierte Taumelbewegung aus.

### Modellierung des Gesamtdrehimpulses

Die in den vorangegangenen Abschnitten besprochenen Drehmomente bestimmen die Änderung des Gesamtdrehimpulses des Quadropters. Der Quadropter besteht jedoch aus fünf Teilkörpern (dem Rahmen und den Rotoren), die unterschiedliche Winkelgeschwindigkeiten besitzen können. Um eine Aussage über die Rotation der einzelnen Körper treffen zu können, muss daher zuerst der Zusammenhang zwischen deren Rotationsbewegungen untersucht werden. Dazu wird eine Gleichung hergeleitet, die den Gesamtdrehimpuls als Summe der Drehimpulse der einzelnen Teilkörper beschreibt und dabei den Zusammenhang zwischen den Winkelgeschwindigkeiten der Teilkörper berücksichtigt.

Für den Rahmen (body) ergibt sich der Drehimpuls  $\vec{L}_B$  über die oben angegebene Gleichung 2.2.11. Dafür wird  $\vec{\omega}_B$  als die Winkelgeschwindigkeit des Rahmens definiert,  $I_B$  als der Trägheitstensor für die Rotation des Rahmens um den Schwerpunkt des Quadropters. Da die Achsen des Quadropter-Koordinatensystems Hauptträgheitsachsen des Rahmens sind und der Gesamtschwerpunkt des Quadropters aufgrund der Symmetrie nur vertikal vom Rahmenschwerpunkt abweicht, ist  $I_B$  eine Diagonalmatrix.

$$\vec{L}_B = I_B \vec{\omega}_B \quad (2.2.12)$$

Für die Bewegung der Rotoren wird  $\Omega_i$  als die skalare Winkelgeschwindigkeit des Rotors  $i$  in seiner Vorzugsrichtung relativ zum Rahmen definiert. Die gesamte Winkelgeschwindigkeit des Rotors  $i$  beträgt damit:

$$\vec{\omega}_i = \vec{\omega}_B + (\Omega_i \vec{z}) \cdot \text{dir}_i \quad (2.2.13)$$

$\vec{z}$  ist dabei der Einheitsvektor in Richtung der z-Achse.

Um den Drehimpuls der Rotoren zu bestimmen, betrachten wir zunächst die Trägheitstensoren für die Rotation jedes Rotors um seinen eigenen Schwerpunkt. Bei Rotoren mit drei und mehr Blättern ist das Trägheitsmoment um jede Achse senkrecht zur Motorachse gleich, der Trägheitstensor ist also unabhängig von der aktuellen Ausrichtung des Rotors um die Motorachse. Für zweiblättrige Rotoren ist dies nicht der Fall, hier ist eine Rotation entlang der Längsachse, die durch die Spitzen der Rotorblätter verläuft, offensichtlich weniger träge als eine Rotation entlang der Querachse. Da die Rotoren sich jedoch in den meisten Fällen schnell um die Motorachse drehen, wird hier vereinfachend angenommen, dass auch zweiblättrige Rotoren für Trägheitsberechnungen ähnlich wie rotierende Scheiben betrachtet werden können, so dass der Trägheitstensor bei einer Rotation um die Motorachse unverändert bleibt.

## 2. Grundlagen der Quadroptersteuerung

Da alle Rotoren (abgesehen von Spiegelung) als baugleich vorausgesetzt werden, ermöglicht es diese Vereinfachung, für die Rotation jedes Rotors um seinen Schwerpunkt den gleichen Trägheitstensor  $I_R$  zu verwenden. Dieser Tensor ist aufgrund der Ausrichtung und Symmetrie der Rotoren ebenfalls eine Diagonalmatrix:

$$I_R = \begin{pmatrix} I_{R,xx} & 0 & 0 \\ 0 & I_{R,yy} & 0 \\ 0 & 0 & I_{R,zz} \end{pmatrix}$$

Der Drehimpuls jedes Rotors  $i$  um seinen eigenen Schwerpunkt wird mit  $\vec{L}_{R_i}$  bezeichnet und ergibt sich aus Gleichung 2.2.11. Durch Einsetzen von Gleichung 2.2.13 erhält man:

$$\begin{aligned} \vec{L}_{R_i} &= I_R \vec{\omega}_i \\ &= I_R \vec{\omega}_B + I_R (\Omega_i \vec{z}) \cdot \text{dir}_i \\ &= I_R \vec{\omega}_B + I_{R,zz} \Omega_i \vec{z} \cdot \text{dir}_i \end{aligned}$$

Um den Drehimpuls des Rotors  $i$  mit Bezug auf den Quadropterschwerpunkt zu bestimmen, muss zusätzlich der Drehimpuls des Rotorschwerpunkts um den Quadropterschwerpunkt berücksichtigt werden, der durch die Rotation des Rahmens entsteht:

$$\begin{aligned} \vec{L}_i &= I_{R_i B} \vec{\omega}_B + \vec{L}_{R_i} \\ &= I_{R_i B} \vec{\omega}_B + I_R \vec{\omega}_B + I_{R,zz} \Omega_i \vec{z} \cdot \text{dir}_i \\ &= (I_{R_i B} + I_R) \vec{\omega}_B + I_{R,zz} \Omega_i \vec{z} \cdot \text{dir}_i \end{aligned}$$

Dabei ist  $I_{R_i B}$  der Trägheitstensor, der die Trägheit der Rotation des Rotorschwerpunkts um den Quadropterschwerpunkt beschreibt. Dieser Tensor ist für jeden Rotor aufgrund der unterschiedlichen Position der Schwerpunkte unterschiedlich. Er errechnet sich aus der Höhe  $h$  und der horizontalen Entfernung  $l$  des Rotorschwerpunkts relativ zum Quadropterschwerpunkt, sowie der Rotormasse  $m_R$  (vgl. [Dem94, Seite 136]):

$$\begin{aligned} I_{R_V B} &= m_R \cdot \begin{pmatrix} h^2 & 0 & hl \\ 0 & h^2 + l^2 & 0 \\ hl & 0 & l^2 \end{pmatrix} & I_{R_H B} &= m_R \cdot \begin{pmatrix} h^2 & 0 & -hl \\ 0 & h^2 + l^2 & 0 \\ -hl & 0 & l^2 \end{pmatrix} \\ I_{R_L B} &= m_R \cdot \begin{pmatrix} h^2 + l^2 & 0 & 0 \\ 0 & h^2 & -hl \\ 0 & -hl & l^2 \end{pmatrix} & I_{R_R B} &= m_R \cdot \begin{pmatrix} h^2 + l^2 & 0 & 0 \\ 0 & h^2 & hl \\ 0 & hl & l^2 \end{pmatrix} \end{aligned}$$

In der Summe dieser Tensoren  $I_{RB}$  heben sich die nichtdiagonalen Elemente gegenseitig auf, so dass eine Diagonalmatrix entsteht. Der Grund dafür liegt in der symmetrischen Verteilung

der vier Rotorschwerpunkte.

$$\begin{aligned} I_{RB} &= \sum_{i \in \{V, H, L, R\}} I_{R_i B} \\ &= m_R \cdot \begin{pmatrix} 4h^2 + 2l^2 & 0 & 0 \\ 0 & 4h^2 + 2l^2 & 0 \\ 0 & 0 & 4l^2 \end{pmatrix} \end{aligned}$$

Der gesamte Drehimpuls aller Rotoren um den Quadropterschwerpunkt kann nach diesen Überlegungen wie folgt dargestellt werden:

$$\begin{aligned} \vec{L}_{rot} &= \sum_i \vec{L}_i \\ &= \sum_i (I_{R_i B} + I_R) \vec{\omega}_B + I_{R,zz} \Omega_i \vec{z} \cdot \text{dir}_i \\ &= (I_{RB} + 4I_R) \vec{\omega}_B + I_{R,zz} \sum_i (\Omega_i \cdot \text{dir}_i) \vec{z} \end{aligned}$$

Zusammen mit dem Rahmendreimpuls ergibt sich der Gesamtdrehimpuls des Quadropters um seinen Schwerpunkt:

$$\begin{aligned} \vec{L}_{ges} &= \vec{L}_B + \vec{L}_{rot} \\ &= I_B \vec{\omega}_B + (I_{RB} + 4I_R) \vec{\omega}_B + I_{R,zz} \sum_i (\Omega_i \cdot \text{dir}_i) \vec{z} \\ &= (I_B + I_{RB} + 4I_R) \vec{\omega}_B + I_{R,zz} \sum_i (\Omega_i \cdot \text{dir}_i) \vec{z} \end{aligned}$$

Zur Vereinfachung dieser Formel definieren wir:

$$I_Q = I_B + I_{RB} + 4I_R \quad (2.2.14)$$

$I_Q$  ist der Trägheitstensor für die Rotation des gesamten Quadropters um seinen Schwerpunkt. Dieser berücksichtigt nicht nur die Trägheit des Rahmens, sondern auch die der Rotoren, welche der Rotation des Rahmens folgen. Da alle drei summierten Tensoren diagonal sind, ist auch  $I_Q$  eine Diagonalmatrix.

Ferner fassen wir die Winkelgeschwindigkeiten der Rotoren gegen den Rahmen zur Restgeschwindigkeit  $\Omega_r$  zusammen:

$$\Omega_r = \sum_i (\Omega_i \cdot \text{dir}_i) \quad (2.2.15)$$

Dies ermöglicht eine vereinfachte Darstellung des Gesamtdrehimpulses:

$$\vec{L}_{ges} = I_Q \vec{\omega}_B + I_{R,zz} \Omega_r \vec{z} \quad (2.2.16)$$

## 2. Grundlagen der Quadroptersteuerung

### Winkelbeschleunigung des Rahmens

Aus Gleichung 2.2.16 lässt sich durch Ableitung der Zusammenhang zwischen äußerem Drehmoment und Winkelbeschleunigung ermitteln. Es jedoch zuerst notwendig, die Konsequenzen der Wahl des Koordinatensystems genauer zu betrachten.

Alle Drehmomente, Winkelgeschwindigkeiten, Drehimpulse und Trägheitstensoren wurden bisher im körperfesten Koordinatensystem angegeben, da sie in diesem System einfach dargestellt werden können. Im Fall der Trägheitstensoren ist ein weiterer Vorteil, dass diese im körperfesten Koordinatensystem konstante Eigenschaften des Quadropters sind. Dieses Koordinatensystem dreht sich jedoch mit dem Rahmen des Quadropters mit, es handelt sich also nicht um ein Inertialsystem.

Solange sich alle Komponenten einer Formel auf den gleichen Zeitpunkt beziehen, wie es in den vorherigen Abschnitten immer implizit der Fall war, hat die zeitliche Veränderlichkeit des Koordinatensystems keine Auswirkung. Bei einer Ableitung über die Zeit muss die Rotation des Koordinatensystems jedoch beachtet werden.

In einem Inertialsystem entspricht die Änderung des Drehimpulses dem äußeren Drehmoment, in rotierenden Systemen entstehen jedoch Scheinmomente, die keine äußere Ursache besitzen. Um das äußere Drehmoment zu erhalten, muss die Ableitung daher in einem raumfesten Inertialsystem  $R$  erfolgen:

$$\vec{M} = \left( \frac{d\vec{L}}{dt} \right)_R$$

In einem solchen System sind aber die Trägheitstensoren nicht konstant. Es ist jedoch möglich, das äußere Drehmoment aus der Ableitung im rotierenden körperfesten Koordinatensystem  $K$  zu bestimmen, indem man die Rotationsgeschwindigkeit  $\vec{\omega}$  des Systems berücksichtigt (vgl. [Dem94, Seite 141]):

$$\vec{M} = \left( \frac{d\vec{L}}{dt} \right)_K + \vec{\omega} \times \vec{L}$$

Alle Vektoren in dieser Formel sind im körperfesten Koordinatensystem ausgedrückt.

Mit Hilfe dieser Formel lässt sich nun das Gesamtdrehmoment durch Ableitung von Gleichung 2.2.16 herleiten:

$$\begin{aligned} \vec{M}_{ges} &= \left( \frac{d\vec{L}_{ges}}{dt} \right)_K + \vec{\omega}_B \times \vec{L}_{ges} \\ &= \left( \frac{d(I_Q \vec{\omega}_B + I_{R,zz} \Omega_r \vec{z})}{dt} \right)_K + \vec{\omega}_B \times (I_Q \vec{\omega}_B + I_{R,zz} \Omega_r \vec{z}) \\ &= I_Q \dot{\vec{\omega}}_B + I_{R,zz} \dot{\Omega}_r \vec{z} + \vec{\omega}_B \times (I_Q \vec{\omega}_B + I_{R,zz} \Omega_r \vec{z}) \end{aligned}$$

Das Ergebnis wird zur Winkelbeschleunigung des Rahmens  $\dot{\vec{\omega}}_B$  umgestellt:

$$\begin{aligned}
 I_Q \dot{\vec{\omega}}_B &= \vec{M}_{ges} - I_{R,zz} \dot{\Omega}_r \vec{z} - \vec{\omega}_B \times \begin{pmatrix} I_{Q,xx} \omega_{B,x} \\ I_{Q,yy} \omega_{B,y} \\ I_{Q,zz} \omega_{B,z} \end{pmatrix} - \vec{\omega}_B \times \begin{pmatrix} 0 \\ 0 \\ I_{R,zz} \Omega_r \end{pmatrix} \\
 &= \vec{M}_{ges} - \begin{pmatrix} 0 \\ 0 \\ I_{R,zz} \dot{\Omega}_r \end{pmatrix} - \begin{pmatrix} \omega_{B,y} \cdot I_{Q,zz} \omega_{B,z} - \omega_{B,z} \cdot I_{Q,yy} \omega_{B,y} \\ \omega_{B,z} \cdot I_{Q,xx} \omega_{B,x} - \omega_{B,x} \cdot I_{Q,zz} \omega_{B,z} \\ \omega_{B,x} \cdot I_{Q,yy} \omega_{B,y} - \omega_{B,y} \cdot I_{Q,xx} \omega_{B,x} \end{pmatrix} - \begin{pmatrix} \omega_{B,y} \cdot I_{R,zz} \Omega_r \\ -\omega_{B,x} \cdot I_{R,zz} \Omega_r \\ 0 \end{pmatrix} \\
 &= \vec{M}_{ges} + \begin{pmatrix} 0 \\ 0 \\ -I_{R,zz} \dot{\Omega}_r \end{pmatrix} + \begin{pmatrix} \omega_{B,y} \omega_{B,z} (I_{Q,yy} - I_{Q,zz}) \\ \omega_{B,z} \omega_{B,x} (I_{Q,zz} - I_{Q,xx}) \\ \omega_{B,x} \omega_{B,y} (I_{Q,xx} - I_{Q,yy}) \end{pmatrix} + \begin{pmatrix} -I_{R,zz} \omega_{B,y} \Omega_r \\ I_{R,zz} \omega_{B,x} \Omega_r \\ 0 \end{pmatrix} \quad (2.2.17)
 \end{aligned}$$

An dieser Formel zeigen sich mehrere Trägheitseffekte, welche die Winkelbeschleunigung unabhängig vom äußeren Drehmoment beeinflussen. Der erste Summand nach dem äußeren Drehmoment  $\vec{M}_{ges}$  erfasst das Drehmoment, welches durch Beschleunigung und Abbremsen der Rotoren zwischen Rahmen und Rotoren wirkt. Der zweite Summand beschreibt die gyroscopischen Effekte, welche durch die Bewegung des Rahmens selbst entstehen, wenn dieser sich nicht entlang einer Hauptträgheitsachse dreht. Der letzte Term beschreibt schließlich die gyroscopischen Effekte der Rotoren durch die Änderung der Rotor-Rotationsachsen.

## Lineare Beschleunigung

Die lineare Beschleunigung des Quadropters kann direkt aus den äußeren Kräften hergeleitet werden. Dabei muss allerdings wie auch bei der Winkelbeschleunigung die Rotation des Koordinatensystems berücksichtigt werden. In einem raumfesten Inertialsystem  $R$  gilt, dass die Änderung des Impulses  $m\vec{v}$  einer äußeren Kraft entspricht:

$$\vec{F} = \left( \frac{d(m\vec{v})}{dt} \right)_R$$

Aus der Sicht eines körperfesten Koordinatensystems  $K$ , das mit der Winkelgeschwindigkeit  $\omega$  rotiert, ändert sich jedoch bei einer geradlinigen Bewegung kontinuierlich die Bewegungsrichtung. Dies kann durch die gleiche Umformung berücksichtigt werden, die auch in Abschnitt 2.2.5 angewendet wurde:

$$\begin{aligned}
 \vec{F}_{ges} &= \left( \frac{d(m\vec{v})}{dt} \right)_K + \vec{\omega}_B \times m\vec{v} \\
 &= m\dot{\vec{v}} + \vec{\omega}_B \times m\vec{v} \\
 m\dot{\vec{v}} &= \vec{F}_{ges} - \vec{\omega}_B \times m\vec{v} \quad (2.2.18)
 \end{aligned}$$

## 2. Grundlagen der Quadroptersteuerung

### Vollständige Bewegungsgleichungen

Die folgenden Gleichungen fassen die Ergebnisse der letzten Abschnitte zusammen. Zur besseren Lesbarkeit sind hier alle verwendeten Symbole noch einmal aufgelistet:

- $\vec{\omega}_B$  Winkelgeschwindigkeit des Rahmens
- $\Omega_r$  Summe der Rotorgeschwindigkeiten gegen den Rahmen (Gleichung 2.2.15)
- $dir_i$  Vorzeichen der Drehrichtung des Rotors  $i$  um die  $z$ -Achse (Gleichung 2.2.2)
- $g$  Erdbeschleunigung
- $h$  Abstand der Rotoren vom Gesamtschwerpunkt in  $z$ -Richtung (Gleichung 2.2.1)
- $l$  Abstand der Rotoren vom Gesamtschwerpunkt in  $xy$ -Richtung (Gleichung 2.2.1)
- $m$  Masse des Quadropters
- $\vec{v}$  Geschwindigkeit des Quadropters
- $D_i$  Bremsmoment an Rotor  $i$  (Abschnitt 2.2.2)
- $\vec{F}_{ges}$  Summe aller Kräfte, die auf den Quadropter wirken
- $\vec{F}_W$  Luftwiderstand des Rahmens (Gleichung 2.2.10)
- $H_i$  Horizontalkraft an Rotor  $i$  (Abschnitt 2.2.2)
- $I_Q$  Trägheitstensor des gesamten Quadropters (Gleichung 2.2.14)
- $I_{R,zz}$  Trägheitsmoment eines Rotors um die  $z$ -Achse
- $\vec{M}_{ges}$  Summe aller äußeren Drehmomente
- $R_i$  Rotor-Rollmoment an Rotor  $i$  (Abschnitt 2.2.2)
- $R_k$  Rotationsmatrix vom erdfesten in das körperfeste Koordinatensystem
- $T_i$  Schub am Rotor  $i$  (Abschnitt 2.2.2)

$$\vec{M}_{ges} = \begin{pmatrix} l(T_L - T_R) + \sum_i (R_{i,x} dir_i) + h \sum_i (H_{i,y}) \\ l(T_V - T_H) + \sum_i (R_{i,y} dir_i) - h \sum_i (H_{i,x}) \\ l(H_{V,y} - H_{H,y} + H_{L,x} - H_{R,x}) - \sum_i (D_i dir_i) \end{pmatrix} \quad (2.2.19)$$

$$I_Q \dot{\vec{\omega}}_B = \vec{M}_{ges} + \begin{pmatrix} 0 \\ 0 \\ -I_{R,zz} \dot{\Omega}_r \end{pmatrix} + \begin{pmatrix} \omega_{B,y} \omega_{B,z} (I_{Q,yy} - I_{Q,zz}) \\ \omega_{B,z} \omega_{B,x} (I_{Q,zz} - I_{Q,xx}) \\ \omega_{B,x} \omega_{B,y} (I_{Q,xx} - I_{Q,yy}) \end{pmatrix} + \begin{pmatrix} -I_{R,zz} \omega_{B,y} \Omega_r \\ I_{R,zz} \omega_{B,x} \Omega_r \\ 0 \end{pmatrix} \quad (2.2.20)$$

$$\begin{aligned} m \dot{\vec{v}} &= \vec{F}_{ges} - \vec{\omega}_B \times m \vec{v} \\ &= \begin{pmatrix} R_{k,31} mg + \sum_i (H_{i,x}) - F_{W,x} \\ R_{k,32} mg + \sum_i (H_{i,y}) - F_{W,y} \\ R_{k,33} mg - \sum_i (T_i) - F_{W,z} \end{pmatrix} - \vec{\omega}_B \times m \vec{v} \end{aligned} \quad (2.2.21)$$

Alle Vektoren und Tensoren sind im Quadropter-Koordinatensystem dargestellt.

Gleichung 2.2.19 enthält alle in den vorangegangenen Abschnitten erläuterten äußeren Drehmomente. Darunter wird die Winkelbeschleunigung des Rahmens abhängig vom äußeren Drehmoment sowie vom Bewegungszustand des Rahmens und der Rotoren beschrieben. Schließlich wird die lineare Beschleunigung des Quadropters gezeigt.

### Vergleich

Die oben gezeigten Bewegungsgleichungen wurden mit dem Ziel entwickelt, das in [Bou07] angegebene dynamische Modell nachzuvollziehen. Dabei wurden die gleichen Größen berücksichtigt, die auch im ursprüngliche Modell behandelt werden, die Herleitung der Bewegungsgleichungen aus diesen Größen erfolgte jedoch unabhängig.

Um einen Vergleich zu vereinfachen, sind die Gleichungen aus [Bou07] hier so umgeschrieben, dass sie den in dieser Arbeit verwendeten Schreibweisen und der Struktur der oben angegebenen Formeln folgen:

$$\vec{M}_{ges} = \begin{pmatrix} l(T_L - T_R) + \sum_i (R_{i,x} dir_i) - h \sum_i (H'_{i,y}) \\ l(T_V - T_H) + \sum_i (R_{i,y} dir_i) + h \sum_i (H'_{i,x}) \\ -l(H'_{V,y} - H'_{H,y} + H'_{L,x} - H'_{R,x}) - \sum_i (D_i dir_i) \end{pmatrix} \quad (2.2.22)$$

$$I_Q \begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \vec{M}_{ges} + \begin{pmatrix} 0 \\ 0 \\ I_{R,zz} \dot{\Omega}'_r \end{pmatrix} + \begin{pmatrix} \dot{\theta} \dot{\psi} (I_{Q,yy} - I_{Q,zz}) \\ \dot{\psi} \dot{\phi} (I_{Q,zz} - I_{Q,xx}) \\ \dot{\phi} \dot{\theta} (I_{Q,xx} - I_{Q,yy}) \end{pmatrix} + \begin{pmatrix} I_{R,zz} \dot{\theta} \Omega'_r \\ -I_{R,zz} \dot{\phi} \Omega'_r \\ 0 \end{pmatrix} \quad (2.2.23)$$

$$m \vec{v}_E = \begin{pmatrix} (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \sum_i (T_i) - \sum_i (H'_{i,x}) - F_{W,x} \\ (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) \sum_i (T_i) - \sum_i (H'_{i,y}) - F_{W,y} \\ mg - \cos \psi \cos \phi \sum_i (T_i) \end{pmatrix} \quad (2.2.24)$$

In diesen Gleichungen tauchen mehrere Symbole auf, die in den in dieser Arbeit entwickelten Gleichungen nicht vorkommen. So sind  $\phi$ ,  $\theta$  und  $\psi$  die absoluten Roll-, Nick- und Gierwinkel des Quadropters.  $H'_i$  und  $\Omega'_r$  entsprechen  $H_i$  und  $\Omega_r$ , unterscheiden sich jedoch im Vorzeichen.  $\vec{v}_E$  entspricht der Geschwindigkeit des Quadropters  $\vec{v}$ , bezieht sich jedoch auf das erdfeste Koordinatensystem.

Zunächst sollen die Bewegungsgleichungen für die Rotation verglichen werden. Diese sind (abgesehen von der Darstellung der Winkelgeschwindigkeiten) bis auf zwei Unterschiede identisch: In [Bou07, Seite 24] sind sowohl die Summe der Rotorgeschwindigkeiten  $\Omega'_r$  als auch die Horizontalkraft  $H'_i$  mit umgekehrtem Vorzeichen definiert. Anschauliche Überlegungen bestätigen das Ergebnis in dieser Arbeit und belegen damit, dass das unterschiedliche Ergebnis nicht auf einen Fehler in der Herleitung zurückzuführen ist, sondern eine unterschiedliche Definition der entsprechenden Größen als Ursache hat.

Aus Gleichung 2.2.20 ergibt sich  $-\frac{I_{R,zz}}{I_{Q,zz}} \dot{\Omega}'_r$  als Winkelbeschleunigung des Rahmens um die z-Achse durch eine Winkelbeschleunigung der Rotoren gegenüber dem Rahmen, die z.B. durch

## 2. Grundlagen der Quadroptersteuerung

die Motoren ausgelöst werden kann. Wenn ein Motor einen Rotor positiv um die z-Achse beschleunigt, wird gleichzeitig der Rahmen negativ um die z-Achse beschleunigt. Die angegebene Formel spiegelt das wieder: Unter der Voraussetzung, dass alle anderen Rotoren ihre Geschwindigkeit gegenüber dem Rahmen beibehalten, ist  $\dot{\Omega}_r$  positiv, was zu einer Beschleunigung des Rahmens in die Gegenrichtung führt.

Für die Horizontalkraft gilt ebenfalls, dass die entsprechenden Terme konsistent ein umgekehrtes Vorzeichen aufweisen. Dies gilt insbesondere auch für die Formeln zur linearen Beschleunigung, in denen die Horizontalkraft ohne weitere Faktoren auftaucht. Eine anschauliche Beschreibung der Formeltermine zur Horizontalkraft, die die Richtigkeit der Vorzeichen nahelegt, ist bereits in Abschnitt 2.2.2 gegeben.

Ein weiterer Unterschied besteht darin, dass Gleichung 2.2.23 die Rotation als Ableitung von Roll-Nick-Gier-Winkeln ausdrückt, während in den in dieser Arbeit hergeleiteten Gleichungen der körperfeste Winkelgeschwindigkeitsvektor  $\vec{\omega}_B$  verwendet wird. Die Ableitung der Roll-Nick-Gier-Winkel ist jedoch nur bei einer horizontalen Ausrichtung mit den lokalen Winkelgeschwindigkeiten identisch. Die Bewegungsgleichungen aus [Bou07, Seite 24] sind also eine Näherung, die lediglich nahe der horizontalen Ausrichtung gilt. Eine Betrachtung der Herleitung in [BS05a] und [BS05b] zeigt jedoch, dass das Modell exakt hergeleitet wurde und lediglich für die Darstellung in Roll-Nick-Gier-Winkeln eine Näherung benutzt wurde.

Der Vergleich der Gleichungen für die lineare Beschleunigung zeigt dagegen größere Unterschiede. Neben dem bereits besprochenen umgekehrten Vorzeichen der Horizontalkraft und der unterschiedlichen Darstellung der Ausrichtung (Roll-, Nick- und Gierwinkel in Gleichung 2.2.24, Rotationsmatrix in Gleichung 2.2.21) wird beim Betrachten der Formeln deutlich, dass sich Gleichung 2.2.24 auf ein erdfestes Koordinatensystem bezieht: Die Schwerkraft wirkt in dieser Gleichung exakt in die negative z-Richtung, während der Schub  $T$  abhängig von der Ausrichtung des Quadropters eine Beschleunigung auf allen Achsen auslösen kann. Da  $\vec{v}_E$  in Gleichung 2.2.24 eine Ableitung in einem Inertialsystem ist, entfällt dort der in Gleichung 2.2.21 verwendete Term  $\vec{\omega}_B \times m\vec{v}$ , der den Effekt der Rotation des Koordinatensystems ausdrückt.

Des Weiteren fällt auf, dass die Horizontalkraft  $H'$  in Gleichung 2.2.24 nicht abhängig von der Ausrichtung des Quadropters transformiert wird, sondern direkt in die Beschleunigung auf den beiden horizontalen Achsen eingeht. Dies kann nicht dadurch erklärt werden, dass  $H'$  im erdfesten statt im körperfesten Koordinatensystem dargestellt sei, da dann bei der vorherigen Verwendung von  $H'$  in Gleichung 2.2.23 eine Transformation nötig gewesen wäre. Das Gleichungssystem scheint daher nur dann korrekt zu sein, wenn der Quadropter exakt so ausgerichtet ist, dass das körperfeste und das erdfeste Koordinatensystem übereinstimmen. In diesem Fall wäre jedoch die Transformation von  $T$  ebenfalls nicht notwendig. Es scheint sich bei den Gleichungen aus [Bou07] daher um eine Näherung zu handeln, die nur im Bereich nahe dieser Grundausrichtung des Quadropters gilt.

Der verbleibende Unterschied besteht darin, dass der Luftwiderstand  $\vec{F}_W$  in Gleichung 2.2.24 für die z-Achse nicht beachtet wird. Der Rahmen-Luftwiderstand ist auf dieser Achse jedoch

vermutlich deutlich schwächer als die von den Rotoren verursachten Schubkräfte, so dass dieser Unterschied ebenfalls als Näherung erklärt werden kann.

### 2.3. Sensoren zur Lagebestimmung

Eine Regelung benötigt Sensoren, die Hinweise auf den aktuellen Zustand des zu regelnden Systems geben. Eine Quadrocoptersteuerung muss daher über Sensoren verfügen, die Rückschlüsse auf die aktuelle Ausrichtung des Fluggeräts erlauben.

Eine Option ist die Verwendung eines Lagekreisels, wie er in [BGK<sup>+</sup>98, Seite 76] beschrieben ist, der durch das Prinzip der Drehimpulserhaltung eine auf mittlere Zeit stabile Referenzausrichtung beibehält. Der bereits angesprochene Quadrocopter *X-Ufo* benutzt einen solchen Sensor zur Lagebestimmung.

Eine kleinere und leichtere Alternative zum Nachverfolgen der Rotation wird durch Gyrosensor-ICs geboten, die ihre eigene Rotationsgeschwindigkeit messen, indem sie die auf schwingende Massen wirkenden Trägheitskräfte ermitteln. Da sie sehr kleine bewegliche Elemente enthalten, werden solche ICs als MEMS (Micro-electromechanical systems) bezeichnet. Miniaturisierte Lagekreisel mit vergleichbarer Größe werden laut [Ber03] nicht kommerziell hergestellt. Daher bieten MEMS-Gyrosensoren einen deutlichen Vorteil in Größe und Gewicht. Wenn die Rotationsgeschwindigkeit um alle drei Achsen zu jedem Zeitpunkt bekannt ist, kann die gesamte Rotationsbewegung nachvollzogen werden, so dass die Ausrichtung relativ zu einer Startausrichtung rekonstruiert werden kann.

Weder Lagekreisel noch Gyrosensoren können langfristig stabile Ausrichtung bereitstellen. Ein Lagekreisel wird durch mechanische Ungenauigkeiten langsam aus seiner ursprünglichen Ausrichtung ausgelenkt, während bei einem MEMS-Gyrosensor die Messungenauigkeiten durch Rauschen, Drift und zeitliche Auflösung bei der Rekonstruktion der Ausrichtung ebenfalls zu einer immer größeren Abweichung führen. Selbst eine exakte Nachverfolgung der raumfesten Ausrichtung wäre jedoch alleine nicht zur langfristigen Bestimmung der horizontalen Lage geeignet, da die horizontale Lage an einem Punkt auf der Erdoberfläche aufgrund der Erdrotation nicht raumfest ist.

Man benötigt daher Möglichkeiten, die Ausrichtung relativ zur Erde zu bestimmen. Dazu kann die Schwerkraftrichtung mit Hilfe eines Beschleunigungssensors gemessen werden. Beschleunigungssensoren sind ebenfalls als MEMS-ICs verfügbar, sind jedoch bei der Verwendung als Lagesensor anfällig für Störungen durch Vibrationen und tatsächliche Beschleunigungen (vgl. [Fis10, Seite 1]). In Abschnitt 2.4 werden jedoch Techniken vorgestellt, mit denen die Messwerte von Gyro- und Beschleunigungssensoren kombiniert werden können, um eine bessere Abschätzung der Ausrichtung zu ermitteln.

Im Folgenden wird kurz die Funktionsweise jeder der angesprochenen Sensorarten beschrieben.

## 2. Grundlagen der Quadroko­ptersteuerung

### 2.3.1. Lagekreisel

Diese Art von Sensor enthält eine schnell rotierende Scheibe, welche so aufgehängt ist, dass sie nahezu widerstandsfrei um jede Achse rotieren kann. Aufgrund der Drehimpulserhaltung kann sich die Rotationsachse der Scheibe nur durch ein äußeres Drehmoment ändern. Wenn die Scheibe frei drehbar und zentral aufgehängt ist, werden jedoch kaum äußere Drehmomente übertragen. Je nach Qualität der Aufhängung und dem Betrag des gespeicherten Drehimpulses ändert sich daher die Rotationsachse der rotierenden Scheibe im raumfesten Koordinatensystem nur sehr langsam. Wenn die Scheibe so angedreht wird, dass ihre Rotationsachse parallel zur Schwerkraftachse verläuft (z.B. vor dem Start des Quadroko­pters), kann aus der Auslenkung ihrer Rotationsachse gegen den Quadroko­pter die Auslenkung des Quadroko­pters aus der horizontalen Lage abgelesen werden.

Der im X-Ufo eingebaute Lagekreisel funktioniert allerdings konstruktionsbedingt nur bis zu einer begrenzten Schräglage des Quadroko­pters von ca. 20°. Wenn diese überschritten wird, stößt die Kreisscheibe an das Gehäuse, was zu einem Verlust der Ausrichtungsinformation und in Folge zu einem Absturz des X-Ufos führen kann, der auch „Flip of Death“ genannt wird (siehe [Bor11]).

### 2.3.2. MEMS-Beschleunigungssensoren

Das Funktionsprinzip von MEMS-Beschleunigungssensoren besteht im Allgemeinen darin, die Kräfte zwischen einer beweglich aufgehängten Testmasse und dem fest mit dem Rahmen verbundenen Rest des Sensorchips zu messen. Idealerweise werden diese Kräfte nur durch die Trägheit der Testmasse bei Beschleunigung des Sensors sowie durch die Gravitation ausgelöst. Die Gravitation lenkt die Testmasse in Richtung des Erdmittelpunkts aus, eine Beschleunigung des Sensors erzeugt dagegen eine Auslenkung der Testmasse entgegen der Beschleunigungsrichtung. Ein Beschleunigungssensor misst daher die Beschleunigung relativ zum freien Fall.

In der Methode der Kraftmessung unterscheiden sich verschiedene MEMS-Sensoren jedoch deutlich. Eine verbreitete Technik ist die kapazitive Messung, bei der Elektroden mit der Testmasse verbunden sind, deren Abstand zu unbeweglichen Elektroden durch die Auslenkung der Masse verändert wird. Dies führt zu einer Änderung der Kapazität zwischen den Elektroden, die mit Hilfe einer entsprechenden Schaltung gemessen werden kann (vgl. [Kes99]). Ein Beispiel für kapazitive Beschleunigungssensoren ist der *MMA7260QT* von Freescale Semiconductor. Bei diesem Sensor bewegt sich jede mit der Testmasse verbundene Elektrode B zwischen zwei festen Elektroden A und C. Eine Auslenkung der Testmasse, die die Kapazität zwischen A und B vergrößert, verkleinert dabei die Kapazität zwischen B und C ([Fre08, Seite 4]).

Eine weitere Technik besteht darin, ein piezoelektrisches Material der auftretenden Kraft auszusetzen. Durch eine Verformung des Materials findet eine Ladungstrennung statt. Die Menge

der freigesetzten Ladung kann mit einem Ladungsverstärker oder einem Elektrometerverstärker bestimmt werden ([Buc09]). Aus dieser Messung kann auf die Stärke der Kraft zurückgeschlossen werden. Piezoelektrische Sensoren sind laut [Kis] äußerst robust und langlebig und können einen großen Dynamikbereich von Beschleunigungen messen. Im Gegensatz zu kapazitiven und piezoresistiven Sensoren ist diese Technik nach [Kes99] allerdings nicht dazu geeignet, eine konstante Kraft zu messen. Dies ist laut [Kis] dadurch zu erklären, dass mit einem Ladungsverstärker keine drifffreie Messung möglich ist, da das Abfließen von Ladungen nicht vollständig verhindert werden kann. Messungen von Kräften, die über Stunden konstant wirken, sind daher nicht fehlerfrei möglich. [HTLMB06] beschreibt einen solchen MEMS-Beschleunigungssensor mit piezoelektrischer Kraftmessung.

Andere Sensoren verwenden wiederum ein piezoresistives Material in der Aufhängung der Testmasse, das durch die wirkenden Kräfte verformt wird und dadurch seinen Widerstand ändert. Laut [LPL<sup>+</sup>03] sind piezoresistive Sensoren im Vergleich zu kapazitiven Sensoren besser für die Messung von hochfrequenten Beschleunigungen geeignet, sie sind jedoch anfälliger für Temperaturschwankungen und Drift. Im Gegensatz zu piezoelektrischen Sensoren sind sie auch für die Messung statischer Beschleunigungen geeignet.

### 2.3.3. MEMS-Gyrosensoren

Laut [Ber03] nutzen alle MEMS-Gyrosensoren den Coriolis-Effekt zur Messung der Winkelgeschwindigkeit. Dieser Effekt beschreibt die scheinbare Ablenkung einer bewegten Masse  $m$  in einem rotierenden Bezugssystem durch die Coriolis-Kraft  $\vec{F}_c = 2m(\vec{v}' \times \vec{\omega})$ . Dabei ist  $\vec{\omega}$  die Rotationsgeschwindigkeit des rotierenden Systems und  $\vec{v}'$  die scheinbare Geschwindigkeit der Masse im rotierenden System ([Dem94, Seite 83]). Die Coriolis-Kraft wirkt also senkrecht zur scheinbaren Bewegungsrichtung der Masse sowie zur Rotationsachse, ist besonders groß, wenn die Bewegung senkrecht zur Rotationsachse erfolgt, und verschwindet für Bewegungen parallel zur Rotationsachse des Systems.

Wenn eine Testmasse auf dem Sensor-IC in Richtung der x-Achse bewegt wird, ergibt sich eine Kraft in der yz-Ebene, wenn der Sensor-IC rotiert:

$$\vec{F}_c = 2m \begin{pmatrix} v \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} 0 \\ -2mv\omega_z \\ 2mv\omega_y \end{pmatrix}$$

Durch Messen dieser Kraft kann die Rotationsgeschwindigkeit um y- und z-Achse bestimmt werden.

Es ist jedoch offensichtlich nicht möglich, eine Testmasse auf einem Sensor-IC konstant linear zu bewegen. Daher werden die Testmassen in MEMS-Gyrosensoren zu Vibrationen angeregt. Bei einer linearen Vibration schwingt die Geschwindigkeit der Masse entlang einer Achse zwischen  $-v$  und  $v$ . Dadurch wirkt auf die Masse unter Rotation auch eine Coriolis-Kraft mit

## 2. Grundlagen der Quadrokoptersteuerung

wechselndem Vorzeichen, welche die Masse zu einer Schwingung entlang einer anderen Achse anregt, die senkrecht zur bestehenden Vibrationsachse steht. Die Stärke dieser Vibration gibt Auskunft über die Rotationsgeschwindigkeit.

Verschiedene Bauformen realer Sensorchips nutzen dieses Prinzip auf unterschiedliche Art und Weise. Als Beispiele werden hier Stimmgabel-Gyroskope und Gyroskope mit vibrierenden Scheiben genauer beschrieben.

Ein Stimmgabel-Gyroskop enthält zwei Massen, die an den Zinken einer stimmgabelförmigen Aufhängung befestigt sind. Die beiden Massen werden zur Schwingung auf der Gabelebene<sup>4</sup> angeregt, wobei die beiden Zinken entgegengesetzt schwingen. Dadurch heben sich die Schwingungen am Gabelpunkt gegenseitig auf, so dass sie sich idealerweise nicht auf den Rest des ICs ausbreiten. Eine Rotation des Sensorchips um die Achse in der Gabelebene, die senkrecht zur Vibrationsrichtung der Zinken ist, regt die Zinken über den oben beschriebenen Mechanismus zu entgegengesetzten Schwingungen auf der Achse senkrecht zur Gabelebene an.

Gyroskope mit vibrierender Scheibe ähneln Lagekreiseln. Die Scheibe ist jedoch nicht frei drehbar, sondern lediglich gefedert aufgehängt und wird nicht konstant gedreht, sondern zu einer schwingenden Rotationsbewegung angeregt, bei der sich die Scheibe nur um einen kleinen Winkel in beide Richtungen dreht. Die Coriolis-Kraft führt aus Sicht des Sensorchips zu einer Neigung der Scheibe aus ihrer Ebene, die z.B. mit Hilfe von Kapazitätsmessungen zwischen der Scheibe und mehreren darunter angebrachten Kondensatorplatten festgestellt werden kann. Aus der Stärke und Richtung der Auslenkung kann die Rotationsgeschwindigkeit des Sensors um die Achsen senkrecht zur Achse der Scheibe bestimmt werden.

### 2.4. Zustandsschätzung

Mit Hilfe der besprochenen Sensoren können Rückschlüsse auf den aktuellen Zustand des Quadrokopters gezogen werden. Für eine Regelung zur Stabilisierung sind insbesondere die Ausrichtung im erdfesten Koordinatensystem und deren Änderung (die Winkelgeschwindigkeit) von Interesse. Letztere kann mit Hilfe von Gyrosensoren direkt gemessen werden; für eine gute Abschätzung der Ausrichtung müssen jedoch die Messwerte mehrerer verschiedenartiger Sensoren kombiniert werden, da auf jede Art von Sensor eine eigene Art von Störeinflüssen wirkt.

Gyrosensoren können schnelle Änderungen in der Ausrichtung mit hoher Genauigkeit erfassen, ermöglichen aber nicht das Messen der absoluten Ausrichtung. Der Messwert des Sensors unterliegt langsamen Veränderungen durch Schwankungen von Temperatur und anderen Umgebungsvariablen. Solche sehr niederfrequenten Messabweichungen werden als Drift bezeichnet. Bei anderen Sensorarten existiert ebenfalls eine Drift, bei Gyrosensoren besitzt sie aber

---

<sup>4</sup>Mit Gabelebene ist die Ebene gemeint, die von den beiden Zinken aufgespannt wird.

eine besonders hohe Bedeutung, da Gyrosensoren nur die Änderung der Ausrichtung messen. Durch geringe, aber langfristige Messabweichungen sammelt sich daher eine immer größere Abweichung der geschätzten Ausrichtung an.

Die wichtigsten Störeinflüsse bei der Messung der Schwerkraftrichtung durch Beschleunigungssensoren sind echte Beschleunigungen, wobei Vibrationen (vergleichsweise hochfrequente periodische Beschleunigungen) als Sonderfall betrachtet werden können. Magnetfeldsensoren ermöglichen die Lagebestimmung anhand des Erdmagnetfelds, werden aber von Magnetfeldern gestört, die auf dem Quadropter selbst entstehen, sowie von Magnetfeldern in der Umgebung, in der sich das Gerät bewegt.

Die von den Sensoren gemessenen Daten können kombiniert werden, um eine genauere Schätzung der Lage zu erhalten, als es mit jeder einzelnen Sensorart möglich wäre. Das Ermitteln eines wahrscheinlichen Zustands aus indirekten oder fehlerbehafteten Messungen wird auch als Zustandsschätzung bezeichnet.

Das Schätzen der Ausrichtung aus den genannten Sensorarten wurde bereits vielfach untersucht, und es wurden unterschiedliche Verfahren dazu entwickelt. Im Folgenden werden drei häufig eingesetzte Verfahren kurz beschrieben und untersucht, um eine geeignete Auswahl für die zu entwickelnde Steuerungssoftware zu treffen. Zunächst werden das Kalman-Filter und das erweiterte Kalman-Filter besprochen, für die [WB06] eine genauere Einführung gibt. Daraufhin wird kurz die Familie der Partikelfilter umrissen, in die z.B. [DJ08] detaillierter einführt. Schließlich wird das von Mahony et al. in [MHP08] entwickelte komplementäre Filter vorgestellt.

### 2.4.1. Kalman-Filter

Das Kalman-Filter, benannt nach Rudolf Emil Kálmán der das Filter 1960 in [Ká60] vorstellte, schätzt den aktuellen Zustand eines Systems in diskreten Schritten. In jedem Schritt wird zuerst aus dem letzten geschätzten Zustand ein neuer Zustand anhand eines vorgegebenen Modells berechnet, ohne Sensoreingaben zu berücksichtigen. Danach wird die neue Schätzung über die Sensoreingaben korrigiert. Das Kalman-Filter kombiniert also Vorwissen über das Verhalten des Systems mit den tatsächlichen Messwerten. Dabei werden die Ungenauigkeit des Systemmodells, der aktuellen Schätzung und der Sensordaten berücksichtigt, um zuverlässigere Informationen höher zu gewichten.

Der aktuell geschätzte Zustand wird als mehrdimensionale Normalverteilung  $\mathcal{N}(\hat{x}, P)$  dargestellt, die als Zustandsverteilung bezeichnet wird. Der Erwartungsvektor  $\hat{x}$  der Verteilung entspricht dem wahrscheinlichsten Zustand. Die Ungenauigkeit der Schätzung wird durch die Kovarianzmatrix  $P$  beschrieben. Die diagonalen Elemente von  $P$  geben die Varianz der einzelnen Elemente des Zustandsvektors an, während die nichtdiagonalen Elemente Informationen über den Zusammenhang zwischen den unterschiedlichen Komponenten von  $\hat{x}$  enthalten.

## 2. Grundlagen der Quadroptersteuerung

In jedem Schritt  $k$  findet zunächst eine Vorhersage statt, die ausgehend von der letzten Zustandsverteilung  $\mathcal{N}(\hat{x}_{k-1}, P_{k-1})$ , einem Vektor mit bekannten Steuergrößen  $u_{k-1}$  (z.B. Steuersignale für Motoren) und einem linearen Modell des Systems in Form der quadratischen Matrix  $A$  eine neue Zustandsverteilung  $\mathcal{N}(\hat{x}_k^-, P_k^-)$  ergibt:

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q\end{aligned}$$

Die Matrix  $B$  transformiert die Steuergrößen des Vektors  $u_{k-1}$  in deren Auswirkungen auf den Zustandsvektor in der Zeit zwischen zwei Schritten.  $Q$  ist eine Kovarianzmatrix, welche die Ungenauigkeit des Systemverhaltens selbst sowie der Modellierung des Systems beschreibt.

Als Beispiel könnte der Zustandsvektor die Position und Geschwindigkeit eines Fahrzeugs enthalten. Die Matrix  $A$  kann dann so gewählt werden, dass die Geschwindigkeit in der Vorhersage erhalten bleibt, und die Position abhängig von der Geschwindigkeit geändert wird.

Da das Verhalten eines Systems nie mit absoluter Sicherheit bestimmt ist, wächst in der Vorhersage die Ungenauigkeit der Zustandsschätzung. Durch die Auswertung der Sensordaten wird die Genauigkeit wieder verbessert. Da die Sensoren oft nicht direkt die Zustandsvariablen messen, wird beim Kalman-Filter angenommen, dass die Messwerte der Sensoren  $z$  über eine weitere Matrix  $H$  linear vom Zustand abhängen und mit einem normalverteilten Rauschen überlagert sind, das durch die Kovarianzmatrix  $R$  bestimmt ist.

Bei der Korrektur der Vorhersage durch die Sensordaten wird die Differenz zwischen den tatsächlichen Messwerten  $z_k$  und den der Vorhersage nach erwarteten Messwerten  $H\hat{x}_k^-$  gebildet. Da die Ungenauigkeit  $P_k^-$  der Vorhersage und die Ungenauigkeit  $R_k$  der Sensordaten bekannt sind, kann eine Gewichtung bestimmt werden, mit der die Differenz auf die Vorhersage aufgerechnet wird. Dies kann an einem Beispiel mit skalaren Werten verdeutlicht werden: Wenn ein Spannungssensor 3 Volt misst, der Vorhersage nach aber eine Spannung von 2 Volt erwartet würde, kann der wahrscheinlichste tatsächliche Wert über die Formel  $2V + \alpha \cdot (3V - 2V)$  berechnet werden.  $\alpha$  liegt dabei zwischen 0 und 1 und ist ein Maß für das Verhältnis der Genauigkeit zwischen Messung und Vorhersage. Wenn die Messung mit einer deutlich größeren Ungenauigkeit behaftet ist als die Vorhersage, ist  $\alpha$  klein, d.h. die Differenz wird geringer gewichtet. Die Formel ergibt dann eine verbesserte Schätzung nahe 2 Volt. Falls die Vorhersage jedoch deutlich ungenauer ist als die Messung, dann liegt  $\alpha$  nahe 1, so dass ein Wert nahe 3 Volt für die Spannung geschätzt wird.

Analog dazu berechnet das Kalman-Filter die sogenannte Kalman-Matrix  $K_k$  aus den Kovarianzmatrizen  $P_k^-$  und  $R$  sowie der Matrix  $H$ . Die Kalman-Matrix transformiert die Differenz zwischen den vorhergesagten und den tatsächlichen Messwerten gewichtet in den Zustandsraum. Das Ergebnis wird als Korrektur auf den vorhergesagten Zustandsvektor addiert. Dadurch entsteht die verbesserte Schätzung  $\hat{x}_k$ . Die entsprechende Kovarianzmatrix  $P_k$  wird

ebenfalls über die Kalman-Matrix  $K_k$  bestimmt:

$$\begin{aligned} K_k &= P_k^- H^\top (H P_k^- H^\top + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \\ P_k &= (I - K_k H) P_k^- \end{aligned}$$

Die neue Zustandsverteilung  $\mathcal{N}(\hat{x}_k, P_k)$  wird dann als Ausgangspunkt für den nächsten Schritt benutzt. Auf eine Erklärung der Berechnung der Kalman-Matrix wird hier verzichtet.

Das Kalman-Filter ist ein optimales Schätzverfahren, es liefert also die beste mögliche Schätzung zu den Eingaben, sofern das Modell dem tatsächlichen Systemverhalten entspricht. Andererseits können mit dem Kalman-Filter nur lineare Systeme mit normalverteilten Ungenauigkeiten modelliert werden. Insbesondere die erste Einschränkung verhindert eine direkte Verwendung des Kalman-Filters für die Schätzung von Ausrichtungen aus Winkelgeschwindigkeiten. Da die Rotation im Gegensatz zur linearen Bewegung nicht kommutativ ist, muss die Reihenfolge der Rotationen beachtet werden, um die korrekte Ausrichtung zu erhalten. Dies ist im linearen Modell des einfachen Kalman-Filters jedoch nicht möglich.

Um auch nichtlineare Modelle für die Vorhersage zu erlauben, wurde das erweiterte Kalman-Filter entwickelt. In dieser Variation des Kalman-Filters kann eine beliebige Funktion  $f$  für die Vorhersage des Folgezustands verwendet werden:

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1})$$

Der Zusammenhang zwischen Zustand und Messwerten kann im erweiterten Kalman-Filter durch eine beliebige Funktion  $h$  ausgedrückt werden:

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-))$$

Durch die Verwendung beliebiger Funktionen ist die korrekte Zustandsverteilung jedoch im Allgemeinen keine Normalverteilung mehr. Man kann sich als Beispiel vorstellen, dass die Position eines Autos modelliert wird, das auf eine Gabelung zufährt. Ein nichtlineares Modell kann aussagen, dass das Auto mit gleichbleibender Geschwindigkeit weiterfährt und abhängig von seiner Position entweder der linken oder der rechten Straße folgt. Auch wenn die Position des Autos auf der Straße zunächst normalverteilt ist, befindet sich das Auto nach diesem Modell hinter der Gabelung entweder auf der einen oder auf der anderen Straße, aber nie dazwischen. Dies entspricht eindeutig nicht mehr einer Normalverteilung, die immer ein zentrales Maximum besitzt.

Das erweiterte Kalman-Filter nähert die Zustandsverteilung dennoch als Normalverteilung an, um eine effiziente Berechnung des neuen geschätzten Zustands nach dem Muster des einfachen Kalman-Filters zu ermöglichen. Das erweiterte Kalman-Filter ist daher allerdings kein optima-

## 2. Grundlagen der Quadroptersteuerung

les Filter, und erfordert auch einen höheren Rechenaufwand als das einfache Kalman-Filter, da zur Berechnung der Kovarianzmatrizen für die Zustandsverteilung in jedem Schritt vier Matrizen mit partiellen Ableitungen von  $f$  und  $h$  um den aktuellen Zustand aufgestellt und mit den Kovarianzmatrizen verrechnet werden müssen.

In der Praxis wird das erweiterte Kalman-Filter durchaus für das Schätzen von Ausrichtungen eingesetzt (vgl. z.B. [KG04]). Die große Zahl an Fließkommaberechnungen, die in jedem Schritt für das Multiplizieren und Invertieren von Matrizen erforderlich sind, macht das Filter jedoch unattraktiv für den Einsatz auf rechenschwachen CPUs ohne Fließkommaeinheit.

### 2.4.2. Partikelfilter

Partikelfilter sind eine Familie von Algorithmen, deren grundsätzliche Idee darin besteht, die geschätzte Zustandsverteilung als Menge gewichteter konkreter Systemzustände (Partikel) anzunähern. Wenn beispielsweise versucht wird, die Position eines Objekts abzuschätzen, dann wird die Zustandsverteilung als Punktwolke dargestellt, wobei Aufenthaltsorte mit einer höheren Wahrscheinlichkeitsdichte stärker gewichtet werden. Diese Darstellung ermöglicht es, beliebige Wahrscheinlichkeitsverteilungen anzunähern, während das Kalman-Filter nur Normalverteilungen zulässt.

Auf der Grundlage der beschriebenen Darstellung können die Schritte der Vorhersage und Korrektur durchgeführt werden. Dabei wird zunächst mit Hilfe eines Systemmodells aus der aktuellen Partikelmenge eine neue Partikelmenge erzeugt, welche die Wahrscheinlichkeitsverteilung im nächsten Schritt ohne Berücksichtigung neuer Messwerte annähert. Jedes Partikel des Vorzustands kann dazu separat betrachtet und anhand des Systemmodells in eine beliebige Zahl neuer Partikel überführt werden, die möglichen Folgezuständen dieses Ausgangszustands entsprechen. Daraufhin wird die Gewichtung aller Partikel im Folgezustand unter Zuhilfenahme der aktuellen Messwerte korrigiert. Um die Anzahl der verwendeten Partikel zu begrenzen, können Partikel auch verworfen werden. In den Details der Erzeugung neuer Partikelmengen und dem Verwerfen bestehender Partikel bestehen große Unterschiede zwischen verschiedenen Partikelfiltern.

Aus dieser Beschreibung sollte deutlich werden, dass Partikelfilter eine hohe Flexibilität erlauben. Im Gegensatz zum erweiterten Kalman-Filter können z.B. auch Zustandsverteilungen dargestellt werden, die mehrere wahrscheinliche Theorien zum aktuellen Zustand abbilden (sog. multimodale Verteilungen). Diese Flexibilität wird jedoch mit einem hohen Einsatz von Rechenleistung und Arbeitsspeicher erkauft, da in jedem Schritt Berechnungen für alle vorhandenen Partikel durchgeführt werden müssen. Weiterhin ermitteln Partikelfilter lediglich eine stochastische Näherungslösung. Die Genauigkeit dieser Lösung hängt entscheidend von der Anzahl der verwendeten Partikel ab, so dass eine Abwägung zwischen Genauigkeit und Rechenaufwand getroffen werden muss. Auf einem Mikrocontroller mit stark begrenzten Res-

sources könnten nur wenige Partikel simuliert werden, so dass die Schätzung vermutlich mit einer hohen Ungenauigkeit behaftet wäre.

### 2.4.3. Explizites komplementäres Filter

Neben diesen allgemeinen Verfahren zur Zustandsschätzung werden auch spezialisierte Verfahren zur Schätzung der Ausrichtung eingesetzt. Ein solches Verfahren ist das explizite komplementäre Filter, das in [MHP08] beschrieben ist. Dieses Filter wurde speziell für die Ermittlung der Ausrichtung aus Messwerten von zwei unterschiedlichen Sensorarten entwickelt. Zum einen ist eine Messung der Winkelgeschwindigkeit notwendig, die von einem dreiaxigen Gyrosensor stammen kann. Zum anderen kann eine beliebige Zahl von Sensoren ausgewertet werden, welche Richtungen messen, die im erdfesten Koordinatensystem bekannt sind, z.B. die Schwerkraftrichtung und die Richtung des Erdmagnetfelds.

Die Idee des Filters besteht darin, die Ausrichtung primär nach den Messwerten des Gyrosensors abzuschätzen und anhand der gemessenen bekannten Richtungen allmählich zu korrigieren. Wenn die gemessenen Richtungen nicht den Richtungen entsprechen, die in der geschätzten Ausrichtung erwartet werden, wird die festgestellte Abweichung im Folgeschritt teilweise korrigiert, indem ein Korrekturwert auf die gemessenen Winkelgeschwindigkeiten addiert wird. Die Stärke der Korrektur ist ein Parameter des Filters, der abhängig von der Zuverlässigkeit der jeweiligen Sensoren gewählt werden sollte<sup>5</sup>. Wenn der Gyrosensor zu einer starken Drift neigt und die Richtungssensoren zuverlässige Messwerte liefern, sollten die Richtungsmessungen vergleichsweise stark bewertet werden; wenn der Gyrosensor dagegen sehr stabil arbeitet und z.B. der Beschleunigungssensor nur sehr verrauschte Messwerte liefert, sollte in jedem Schritt eine deutlich geringere Korrektur erfolgen.

Der Effekt dieses Vorgehens ist, dass schnelle und zufällige Änderungen in den Richtungsmessungen, wie sie durch Vibrationen entstehen können, nur abgeschwächt in die Schätzung der Ausrichtung eingehen. Das Filter wirkt also als Tiefpassfilter für diese Messungen. Dadurch hat der Gyrosensor bei hochfrequenten Änderungen einen deutlich höheren Einfluss auf die geschätzte Ausrichtung als die anderen Sensoren, während diese bei niedrigen Frequenzen ein höheres Gewicht besitzen. Dieses komplementäre Verhalten sorgt dafür, dass die Sensoren in den Frequenzbereichen geringer gewichtet werden, in denen sie wie am Anfang des Abschnitts besprochen hohe Messungenauigkeiten aufweisen. Dadurch entsteht in der Kombination eine Abschätzung der Ausrichtung, die sowohl schnellen Änderungen folgt als auch resistent gegen Drift ist.

Die Gyrodraft kann auf diese Weise allerdings nicht vollständig eliminiert werden, da diese in jedem Schritt vollständig in die Schätzung der Ausrichtung eingeht, aber nur anteilig korri-

<sup>5</sup>Die Gewichtung kann auch dynamisch an verschiedene Situationen angepasst werden. Während aggressiver Flugmanöver mit hohen, schnell wechselnden Beschleunigungen ist der Messwert eines Beschleunigungssensors vermutlich weniger gut als Abschätzung der Schwerkraftrichtung geeignet als im Schwebeflug. Die Messungen eines Magnetfeldsensors könnten während solcher Manöver relativ höher gewichtet werden.

## 2. Grundlagen der Quadroptersteuerung

giert wird. Daher wird in [MHP08] als Erweiterung des Filters vorgeschlagen, die ermittelten Korrekturwerte zu integrieren und mit einem Faktor ebenfalls als Korrektur anzuwenden, um ähnlich wie bei einem PI-Regler die Restabweichung zu beseitigen.

Das explizite komplementäre Filter eignet sich gut für eine Implementierung auf beschränkter Hardware, da nur wenig Arbeitsspeicher und moderate Rechenleistung benötigt werden. Der Zustand des Filters zwischen den Schritten beschränkt sich auf eine Darstellung der Ausrichtung (z.B. als Quaternion), den aktuellen Korrekturwert, der im nächsten Schritt angewendet werden soll (ein dreidimensionaler Vektor) sowie das Integral der Korrekturwerte (ein weiterer dreidimensionaler Vektor), falls diese Erweiterung des Filters verwendet wird. Insbesondere aufgrund dieser Eigenschaften, und aufgrund der vergleichsweise einfachen Verständlichkeit des Filters, wurde dieses auf der Steuerplatine eingesetzt, die für diese Arbeit entwickelt wurde. Eine detailliertere Beschreibung des Filters befindet sich daher in Abschnitt 3.7.

### 2.5. Quaternionen

Um Berechnungen mit Rotationen und Ausrichtungen durchführen zu können, muss eine mathematische Repräsentation für diese ausgewählt werden. Neben den verbreiteten Rotationsmatrizen werden dazu auch oft Quaternionen eingesetzt, da diese eine kompaktere Darstellung und eine effizientere Verkettung von Rotationen ermöglichen. Daher wird in diesem Abschnitt kurz in die Eigenschaften von Quaternionen und ihre Verwendung zur Darstellung von Rotationen eingeführt. Die Informationen zu den Grundlagen stammen größtenteils aus [Ham44], während [Wik12e] sowie [Wik12c] und [Wik12d] hilfreiche Informationen über den Zusammenhang zwischen Quaternionen und Rotationen enthalten.

#### 2.5.1. Grundlagen

Quaternionen sind verwandt mit den komplexen Zahlen, enthalten allerdings eine reelle und drei imaginäre Komponenten. Die imaginären Komponenten sind auf den drei imaginären Zahlen  $i$ ,  $j$  und  $k$  aufgebaut. Für diese gilt die folgende Gleichung:

$$i^2 = j^2 = k^2 = i \cdot j \cdot k = -1$$

Die Multiplikation zwischen  $i$ ,  $j$  und  $k$  ist antikommutativ, d.h.  $ij = -ji$ . Aus diesen Eigenschaften können alle Produkte zwischen  $i$ ,  $j$  und  $k$  berechnet werden:

$$\begin{aligned} i \cdot j \cdot k = -1 &\implies i \cdot j \cdot k \cdot k = -1 \cdot k &\implies -i \cdot j = -k &\implies i \cdot j = k \\ i \cdot j \cdot k = -1 &\implies i \cdot i \cdot j \cdot k = i \cdot -1 &\implies -j \cdot k = -i &\implies j \cdot k = i \\ i \cdot j \cdot k = -1 &\implies i \cdot i \cdot j \cdot k \cdot k = i \cdot -1 \cdot k &\implies j = -i \cdot k &\implies k \cdot i = j \end{aligned}$$

Die Menge der Quaternionen wurde nach Sir William Rowan Hamilton, der die Quaternionen in [Ham44] definierte, mit  $\mathbb{H}$  benannt. Man kann Quaternionen in der Form  $a + bi + cj + dk$  mit  $a, b, c, d \in \mathbb{R}$  aufschreiben, die Schreibweise als 4-Tupel  $(a, b, c, d)$  ist jedoch auch üblich. Eine weitere Darstellung teilt eine Quaternion  $q$  in ihren Realteil und den dreidimensionalen Imaginärteil als Vektor auf:  $q = (a, \vec{v}) = \left(a, \begin{pmatrix} b & c & d \end{pmatrix}^\top\right)$ .

Die Addition von Quaternionen erfolgt komponentenweise, d.h.:

$$\begin{aligned} q_1 + q_2 &= (a_1 + b_1i + c_1j + d_1k) + (a_2 + b_2i + c_2j + d_2k) \\ &= (a_1 + a_2) + (b_1 + b_2)i + (c_1 + c_2)j + (d_1 + d_2)k \end{aligned}$$

Die Formel für die Multiplikation kann aus den oben angegebenen Multiplikationsregeln für die Zahlen  $i, j$  und  $k$  sowie dem Distributivgesetz hergeleitet werden:

$$\begin{aligned} q_1 \cdot q_2 &= (a_1 + b_1i + c_1j + d_1k) \cdot (a_2 + b_2i + c_2j + d_2k) \\ &= a_1a_2 + a_1b_2i + a_1c_2j + a_1d_2k + b_1a_2i + b_1b_2i^2 + b_1c_2ij + b_1d_2ik \\ &\quad + c_1a_2j + c_1b_2ji + c_1c_2j^2 + c_1d_2jk + d_1a_2k + d_1b_2ki + d_1c_2kj + d_1d_2k^2 \\ &= a_1a_2 + a_1b_2i + a_1c_2j + a_1d_2k + b_1a_2i - b_1b_2 + b_1c_2k - b_1d_2j \\ &\quad + c_1a_2j - c_1b_2k - c_1c_2 + c_1d_2i + d_1a_2k + d_1b_2j - d_1c_2i - d_1d_2 \\ &= a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2 \\ &\quad + (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)i \\ &\quad + (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)j \\ &\quad + (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)k \end{aligned}$$

### 2.5.2. Darstellung von Rotationen durch Quaternionen

Rotationen im dreidimensionalen Raum können als Einheitsquaternionen, d.h. als Quaternionen der Länge 1 dargestellt werden, wobei die Länge einer Quaternion laut [Wik12c] genau wie die eines Vektors in  $\mathbb{R}^4$  als  $|(a, b, c, d)| = \sqrt{a^2 + b^2 + c^2 + d^2}$  definiert ist. Um eine Funktion  $\text{rotate}(\vec{v}, q)$  zu definieren, welche die von  $q$  dargestellte Rotation auf den Vektor  $\vec{v}$  anwendet, müssen zunächst noch zwei weitere Operationen eingeführt werden.

Ähnlich wie bei den komplexen Zahlen ist für jede Quaternion  $q = (a, \vec{v})$  eine konjugierte Quaternion  $\bar{q} = (a, -\vec{v})$  definiert. Bei Einheitsquaternionen entspricht die konjugierte Quaternion dem inversen Element, d.h.  $\forall q \in \mathbb{H} \wedge |q| = 1 : \bar{q} = q^{-1}$ . Für die Multiplikation inverser Quaternionen gilt  $q_1^{-1} \cdot q_2^{-1} = (q_2 \cdot q_1)^{-1}$ , daher gilt für Einheitsquaternionen auch  $\bar{q}_1 \cdot \bar{q}_2 = \overline{(q_2 \cdot q_1)}$

Weiterhin wird eine Funktion  $\mathbf{p}(\vec{v}) = (0, \vec{v})$  benötigt, die eine Quaternion aus einem dreidimensionalen Vektor erzeugt, indem die Komponenten des Vektors als imaginäre Komponenten der Quaternion übernommen werden. Eine solche Quaternion, deren Realteil 0 ist, wird auch als rein imaginäre Quaternion bezeichnet.

## 2. Grundlagen der Quadroptersteuerung

$\text{rotate}(\vec{v}, q)$  kann nun wie folgt definiert werden:

$$\text{rotate}(\vec{v}, q) = \mathbf{p}^{-1}(q \cdot \mathbf{p}(\vec{v}) \cdot \bar{q}) \quad (2.5.1)$$

Das Produkt  $q_1 \cdot q_2$  aus zwei Einheitsquaternionen ist ebenfalls eine Einheitsquaternion und entspricht einer verketteten Rotation, bei der zuerst  $q_2$  und dann  $q_1$  angewendet wird. Dies kann aus der Formel für die Rotation eines Vektors hergeleitet werden:

$$\begin{aligned} \text{rotate}(\vec{v}, q_1 \cdot q_2) &= \mathbf{p}^{-1}(q_1 \cdot q_2 \cdot \mathbf{p}(\vec{v}) \cdot \overline{(q_1 \cdot q_2)}) \\ &= \mathbf{p}^{-1}(q_1 \cdot q_2 \cdot \mathbf{p}(\vec{v}) \cdot \bar{q}_2 \cdot \bar{q}_1) \\ &= \mathbf{p}^{-1}(q_1 \cdot \mathbf{p}(\mathbf{p}^{-1}(q_2 \cdot \mathbf{p}(\vec{v}) \cdot \bar{q}_2)) \cdot \bar{q}_1) \\ &= \text{rotate}(\text{rotate}(\vec{v}, q_2), q_1) \end{aligned}$$

Hier sollte darauf hingewiesen werden, dass die Multiplikation von Quaternionen nicht kommutativ ist. Die Verkettung von Rotationen teilt diese Eigenschaft. Wenn man zwei Rotationen in vertauschter Reihenfolge ausführt, erhält man im Allgemeinen ein anderes Ergebnis. Dreht man ein horizontal und nach Norden ausgerichtetes Flugzeug zuerst um  $90^\circ$  um die Ost-Achse und dann um  $90^\circ$  um die Nord-Achse, dann zeigt die Front des Flugzeugs nach Osten, und die rechte Tragfläche nach unten. Dreht man zuerst um die Nord- und dann um die Ostachse, dann zeigt die Front nach oben, und die rechte Tragfläche nach Norden.

Nun wurde beschrieben, wie Rotationen in Quaternionendarstellung verkettet werden können. Um diese Operation zu nutzen, müssen jedoch zuerst Rotationen in dieser Darstellung vorliegen. In [Wik12e] ist beschrieben, wie eine Quaternion zu einer Rotation berechnet werden kann, deren Rotationsachse als Einheitsvektor  $\vec{u}$  und deren Winkel als  $\alpha$  bekannt ist:

$$\text{quaternionAusAchseWinkel}(\vec{u}, \alpha) = \left( \cos \frac{\alpha}{2}, \vec{u} \cdot \sin \frac{\alpha}{2} \right) \quad (2.5.2)$$

Später in dieser Arbeit liegen Rotationen auch in einer Darstellung als nicht-normierter Rotationsvektor  $\vec{\omega}$  vor, dessen Richtung die Rotationsachse und dessen Länge den Winkel der Rotation angibt. Durch Einsetzen von  $\vec{u} = \frac{\vec{\omega}}{|\vec{\omega}|}$  und  $\alpha = |\vec{\omega}|$  in die oben angegebene Formel und spezielle Behandlung der Definitionslücke bei  $|\vec{\omega}| = 0$  kann die folgende Funktion hergeleitet werden:

$$\text{quaternionAusRotationsvektor}(\vec{\omega}) = \begin{cases} \left( \cos \frac{|\vec{\omega}|}{2}, \frac{\vec{\omega}}{|\vec{\omega}|} \cdot \sin \frac{|\vec{\omega}|}{2} \right) & \text{falls } |\vec{\omega}| > 0 \\ \left( 1, (0 \ 0 \ 0)^\top \right) & \text{falls } |\vec{\omega}| = 0 \end{cases} \quad (2.5.3)$$

Bei genauer Betrachtung dieser Funktionen fällt auf, dass alle Rotationen mit Winkeln bis  $360^\circ$  in jede Richtung eindeutig abgebildet werden. Die Rotation mit Winkel  $\alpha$  um die Achse  $\vec{u}$  ergibt daher eine andere Quaternion als die Rotation mit Winkel  $2\pi - \alpha$  um die Achse  $-\vec{u}$ , obwohl beide Rotationen insofern identisch sind, dass sie im Bezug auf die Rotation von Vektoren der

gleichen Transformation entsprechen. Tatsächlich existieren zu jeder Rotation zwei Quaternionen  $q$  und  $-q$ , die bei Anwendung der `rotate`-Funktion das gleiche Ergebnis erzeugen.

Bei manchen Operationen ist es daher sinnvoll, die Quaternionen zuerst in eine einheitliche Darstellung zu transformieren. Wenn man z.B. beim Ausrechnen eines Rotationsvektors aus einer Quaternion  $q = (a, b, c, d)$  den Rotationswinkel aus dem Realteil  $a$  berechnet, erhält man Winkel über  $180^\circ$ , falls  $a < 0$  ist. Um statt dessen die kürzere Rotation in die Gegenrichtung zu erhalten, kann man in diesem Fall die Berechnung mit der Gegenquaternion  $-q = (-a, -b, -c, -d)$  durchführen.

### 2.5.3. Rechenbeispiel

Die Verwendung von Quaternionen zur Darstellung von Rotationen kann anhand des oben beschriebenen anschaulichen Beispiels verdeutlicht werden, bei dem ein nach Norden fliegender Flugzeug zunächst um  $90^\circ$  um die Ost-Achse und dann um  $90^\circ$  um die Nord-Achse gedreht wird. Die Quaternionen, die diesen Rotationen entsprechen, können über die in Gleichung 2.5.2 definierte Funktion `quaternionAusAchseWinkel`( $\vec{u}, \alpha$ ) berechnet werden:

$$\begin{aligned} q_{Ost} &= \text{quaternionAusAchseWinkel} \left( \vec{y}, \frac{\pi}{2} \right) \\ &= \left( \cos \frac{\pi}{4}, 0, \sin \frac{\pi}{4}, 0 \right) \\ &= \left( \sqrt{\frac{1}{2}}, 0, \sqrt{\frac{1}{2}}, 0 \right) \end{aligned}$$

Analog dazu ergibt sich für die Rotation um die Nord-Achse:

$$q_{Nord} = \left( \sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}, 0, 0 \right)$$

Um die Verkettung dieser Rotationen in der Reihenfolge Ost-Nord zu ermitteln, werden die Quaternionen multipliziert:

$$\begin{aligned} q_g &= q_{Nord} \cdot q_{Ost} \\ &= \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2}} - \sqrt{\frac{1}{2}} \cdot 0 - 0 \cdot \sqrt{\frac{1}{2}} - 0 \cdot 0 \\ &\quad + \left( \sqrt{\frac{1}{2}} \cdot 0 + \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2}} + 0 \cdot 0 - 0 \cdot \sqrt{\frac{1}{2}} \right) i \\ &\quad + \left( \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2}} - \sqrt{\frac{1}{2}} \cdot 0 + 0 \cdot \sqrt{\frac{1}{2}} + 0 \cdot 0 \right) j \\ &\quad + \left( \sqrt{\frac{1}{2}} \cdot 0 + \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2}} - 0 \cdot 0 + 0 \cdot \sqrt{\frac{1}{2}} \right) k \\ &= \left( \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right) \end{aligned}$$

## 2. Grundlagen der Quadroptersteuerung

Um nun zu ermitteln, in welche Richtung die Nase des Flugzeugs nach dieser Transformation zeigt, wird die Rotation mit der in Gleichung 2.5.1 definierten Funktion  $\text{rotate}(\vec{v}, q)$  auf den Vektor  $\vec{x}$  angewendet:

$$\begin{aligned}\vec{x}' &= \text{rotate}(\vec{x}, q_g) \\ &= \mathbf{p}^{-1}(q_g \cdot \mathbf{p}(\vec{x}) \cdot \bar{q}_g) \\ &= \mathbf{p}^{-1}\left(\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right) \cdot (0, 1, 0, 0) \cdot \left(\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right)\right) \\ &= \mathbf{p}^{-1}\left(\left(-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, -\frac{1}{2}\right) \cdot \left(\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right)\right) \\ &= \mathbf{p}^{-1}(0, 0, 1, 0) \\ &= \vec{y}\end{aligned}$$

Die Nase des Flugzeugs zeigt also nach Osten, was dem bereits besprochenen anschaulichen Ergebnis entspricht. Wendet man die gleiche Operation auf den Vektor  $\vec{y}$  an, erhält man  $\vec{z}$ , d.h. die rechte Tragfläche zeigt wie erwartet nach unten.

### 2.5.4. Vergleich mit Rotationsmatrizen

Im Vergleich zu anderen Darstellungsarten ist die Verkettung von Rotationen in der Quaternionendarstellung sehr effizient. Wie aus der Multiplikationsformel oben ersichtlich ist, müssen nur 16 Multiplikationen und 12 Additionen bzw. Subtraktionen durchgeführt werden. Bei einer Darstellung durch Rotationsmatrizen werden für die Verkettung laut [Wik12e] 27 Multiplikationen und 18 Additionen bzw. Subtraktionen benötigt. Das Rotieren von Vektoren ist dagegen mit Rotationsmatrizen effizienter möglich. Hier werden nur 9 Multiplikationen und 6 Additionen benötigt, während eine Vektorrotation mit Quaternionen 15 Multiplikationen und 15 Additionen bzw. Subtraktionen erfordert. Es hängt daher von der Anwendung ab, welche Darstellung effizienter ist.

Rotationsmatrizen besitzen neun Komponenten gegenüber den vier Komponenten einer Quaternion. Dadurch belegt eine Rotationsmatrix nicht nur mehr Speicher, sondern enthält auch mehr Redundanzen bzw. Invarianten. Eine Rotationsquaternion muss lediglich die Länge 1 besitzen, während alle Spaltenvektoren einer Rotationsmatrix die Länge 1 besitzen sowie senkrecht zueinander stehen müssen. Aufgrund numerischer Ungenauigkeiten können nach einigen Operationen deutliche Abweichungen von diesen Invarianten entstehen, so dass eine Korrektur der Quaternion oder Matrix erforderlich wird. Bei Quaternionen ist dies analog zum Normalisieren eines Vektors auf  $\mathbb{R}^4$ , bei Matrizen ist dagegen ein größerer Aufwand erforderlich. Ebenso ist die Konvertierung eines Rotationsvektors in eine Quaternion laut [Wik12e] weniger aufwändig als die Konvertierung in eine Rotationsmatrix.

## 2.6. PID-Regler

Das Ziel eines Reglers ist es, eine *Regelgröße* so zu beeinflussen, dass sie einem vorgegebenen *Sollwert* möglichst nahe kommt. Dazu kann der Regler eine *Stellgröße* vorgeben, welche die Regelgröße beeinflusst. Für die Stabilisierung eines Quadropters ist in der Hauptsache eine Regelung der Ausrichtung erforderlich, die über eine Steuerung der Motorleistung erfolgt.

Der PID-Regler gehört zu den bekanntesten Reglern und kann für ein breites Spektrum von Regelproblemen eingesetzt werden. Das Prinzip dieses Reglers kann zudem ohne spezielles Vorwissen verstanden werden, und er ist mit geringem Rechenaufwand realisierbar. Es sollte jedoch beachtet werden, dass auch andere Regler existieren, die u.U. deutlich besser für die Lösung eines gegebenen Regelproblems geeignet sind. Zusätzlich ist der PID-Regler für die Regelung einer skalaren Größe ausgelegt, so dass eine Regelung der Ausrichtung mit einem einzelnen PID-Regler nicht möglich ist. Daher befasst sich Abschnitt 3.8 mit der Frage, wie die Ausrichtung durch die Kombination mehrerer PID-Regler stabilisiert werden kann.

Die Funktionsweise eines PID-Reglers soll am Beispiel eines Temperaturreglers erläutert werden, der mit Hilfe eines Thermometers die aktuelle Temperatur in einem Raum bestimmen kann. Als Stellgröße kann der Regler den Stromfluss durch einen elektrischen Heizkörper steuern. Die Regelgröße dieses Reglers ist die Raumtemperatur, deren gemessener *Istwert* möglichst konstant einen vorgegebenen Sollwert einhalten soll. Dazu kann die Differenz zwischen Sollwert und Istwert betrachtet werden, die als *Regelabweichung*  $e$  bezeichnet wird:

$$e(t) = \text{soll}(t) - \text{ist}(t)$$

Ein einfacher Regler könnte die Leistung des Heizkörpers aus dem Beispiel *proportional* zur Temperaturdifferenz einstellen, so dass mit abnehmender Temperatur eine immer größere Leistung verwendet wird. Das ist das Prinzip eines *P-Reglers*:

$$\text{stell}(t) = k_P \cdot e(t)$$

$k_P$  ist dabei eine Konstante, die auf das zu regelnde System angepasst werden muss. Die Einstellung von  $k_P$  und weiterer Regelparameter für ein gegebenes System wird weiter unten besprochen.

Ein Nachteil des P-Reglers besteht darin, dass die Abweichung zwischen Soll- und Istwert oft nicht vollständig ausgeglichen wird. Im Beispiel des Temperaturreglers entweicht ständig Wärme aus dem Raum, wodurch ein dauerhaftes Heizen erforderlich ist, um die Solltemperatur zu halten. Ein P-Regler sieht jedoch keine Regelabweichung mehr, sobald die Solltemperatur erreicht ist, und erzeugt daher auch einen Stellwert von 0. Die Raumtemperatur wird sich daher auf einen Wert unterhalb der Solltemperatur einstellen. Die nötige Heizleistung, die zum Halten der Solltemperatur benötigt wird, kann auch nicht direkt vorhergesagt werden, da sie von unbekanntem Größen abhängt, z.B. von der Außentemperatur.

## 2. Grundlagen der Quadroptersteuerung

Dieses Problem kann durch einen *I-Regler* gelöst werden. Ein I-Regler bestimmt den Stellwert nicht direkt aus der Regelabweichung, sondern aus deren Integral. Dieses wird mit einem Parameter  $k_I$  gewichtet. Eine konstante Regelabweichung führt damit zu einer immer stärkeren Reaktion des I-Reglers:

$$stell(t) = k_I \int_0^t e(t) dt$$

Der I-Regler kann im Beispiel ein Gleichgewicht bei der Solltemperatur erreichen, da eine Regelabweichung von 0 bei dieser Art von Regler dazu führt, dass der Stellwert unverändert bleibt. I-Regler reagieren jedoch nur allmählich auf Regelabweichungen, während P-Regler einer Abweichung sofort entgegenwirken.

Ein *PI-Regler* kombiniert P-Regler und I-Regler, um die Vorteile beider Arten von Regler zu erhalten:

$$stell(t) = k_P \cdot e(t) + k_I \int_0^t e(t) dt$$

Ein PI-Regler bildet also lediglich die Summe aus dem Stellwert eines P- und eines I-Reglers. Durch eine geeignete Wahl von  $k_I$  und  $k_P$  kann man einen Regler erhalten, der durch den P-Anteil schnell auf Regelabweichungen reagiert und durch den I-Anteil den erforderlichen Stellwert erzeugt, wenn der Sollwert erreicht ist.

Für viele Anwendungen ist es hilfreich, nicht erst einer bestehenden Regelabweichung entgegenzuwirken, sondern bereits auf eine Änderung der Regelabweichung zu reagieren. Das ist insbesondere dann sinnvoll, wenn die Änderung der Regelgröße eine Trägheit besitzt, z.B. wenn die Position eines Objekts geregelt wird, wobei die Stellgröße einer Kraft entspricht. Wenn sich das Objekt bereits an der gewünschten Position befindet, durch einen äußeren Störeinfluss aber eine Geschwindigkeit erhält, sollte der Regler dieser Geschwindigkeit bereits entgegenwirken, bevor eine deutliche Abweichung der Position entsteht.

Dies kann realisiert werden, indem ein zusätzlicher D-Anteil in den Regler eingebaut wird, der auf die Ableitung (derivative) der Regelabweichung reagiert. Dadurch erhält man den *PID-Regler*:

$$stell(t) = k_P \cdot e(t) + k_I \int_0^t e(t) dt + k_D \cdot \dot{e}(t)$$

Der PID-Regler kann zur Lösung vieler unterschiedlicher Regelprobleme eingesetzt werden, er muss dazu jedoch jeweils passend über die Konstanten  $k_P$ ,  $k_I$  und  $k_D$  parametrisiert werden. Es existieren verschiedene Verfahren, mit denen diese Parameter aus Eigenschaften des zu regelnden Systems berechnet werden können, es ist jedoch auch üblich die Parameter experimentell zu ermitteln, indem das Verhalten des Reglers mit verschiedenen Einstellungen getestet wird. Dabei kann es hilfreich sein, einen Parameter testweise sehr hoch oder sehr nied-

rig einzustellen, um das dadurch entstehende unerwünschte Verhalten zu beobachten, z.B. ein Aufschaukeln durch zu starke Korrektur. Auf diese Weise kann das entsprechende Verhalten während eines Experiments einfacher erkannt und eingeordnet werden, um den entsprechenden Regelparameter anzupassen.

Weitere Informationen über PID-Regler befinden sich u.A. in [RW12] und [Wik12b], die auch als Quellen für diesen Abschnitt verwendet wurden. Dort sind auch formalisierte Verfahren beschrieben, um geeignete Regelparameter zu ermitteln.



### 3. Entwicklung einer Steuerplatine



Abbildung 3.0.1.: Der Quadrocopter X-Ufo mit der entwickelten Steuerplatine zum Abschluss der Arbeit. Unter der Platine (im Bild nicht sichtbar) sind ein Fernsteuerungsempfänger und eine Wii MotionPlus befestigt.

Im Folgenden wird eine im Rahmen dieser Arbeit entwickelte Steuerplatine vorgestellt. Dabei wurde insbesondere darauf geachtet, dass die Platine leicht und kostengünstig nachgebaut werden kann, und dass die verwendeten Techniken einfach verständlich sind.

Als Grundlage für die Umsetzung stand ein X-Ufo der Firma Silverlit zur Verfügung. Es handelt sich dabei um einen Spielzeug-Quadrocopter mit einem Durchmesser von 58cm und einem Rotordurchmesser von 20cm. Ein Rahmen aus aufgeschäumtem Kunststoff schützt die Rotoren vor Kontakt mit Hindernissen und ist nach unten zu Standfüßen ausgeprägt. Die Original-Steuerplatine war beschädigt, so dass die Regelung der Ausrichtung nicht mehr funktionierte.

### 3. Entwicklung einer Steuerplatine

Die neu entwickelte Steuerplatine wurde als Ersatz für die Originalplatine in das X-Ufo eingebaut. Die Platine ist daher zwar auf die Eigenschaften dieses Geräts optimiert, kann aber dennoch als Ausgangspunkt für die Entwicklung einer Steuerung für andere Quadroptermodele dienen. Der dazu nötige Aufwand hängt stark von den Eigenschaften des entsprechenden Modells ab. Sofern die Spannung für die Motoren zwischen 9 und 16 Volt liegt und der Quadroptermotor mit Bürstenmotoren angetrieben wird, die über Pulsweitenmodulation gesteuert werden können (siehe Unterabschnitt 3.6.3), kann die Hardware der Steuerplatine unverändert übernommen werden. Es müssen dann lediglich die Parameter des Ausrichtungsreglers (siehe Abschnitt 3.8) und die Linearisierung der Motorsteuerung (siehe Unterabschnitt 3.6.4) angepasst werden.

In Abschnitt 3.1 wird zunächst die gesamte Steuerplatine vorgestellt und ein kurzer Einblick in den Ablauf ihrer Entwicklung gegeben. Im Anschluss wird in Abschnitt 3.2 der eingesetzte Mikrocontroller *ATmega168* von Atmel beschrieben. Die Steuerplatine verwendet einen Gyro- und einen Beschleunigungssensor um Ausrichtung und Rotation zu erfassen. Als Gyrosensor wird die *MotionPlus*-Erweiterung der Spielkonsole *Nintendo Wii* eingesetzt. Die *Wii MotionPlus* und deren Verwendung als Gyrosensor sind in Abschnitt 3.3 beschrieben. Der Beschleunigungssensor ist der von Freescale hergestellte *MMA7260QT*, auf den Abschnitt 3.4 genauer eingeht. Auf den Einsatz weiterer Sensoren wurde verzichtet. Wie in Abschnitt 3.5 beschrieben ist, kann jedoch auch ein Fernsteuerungsempfänger angeschlossen werden.

Abschnitt 3.6 befasst sich mit der Ansteuerung der Motoren durch den Mikrocontroller, die mit Hilfe von vier Leistungstransistoren vom Typ *IRLU3717* erfolgt. Die verbleibenden Abschnitte beschreiben das verwendete Verfahren zur Schätzung der aktuellen Ausrichtung (Abschnitt 3.7) und den Regler zur Stabilisierung der Ausrichtung (Abschnitt 3.8).

#### 3.1. Aufbau der Platine

Es wurden drei Versionen der Steuerplatine entwickelt. Die erste Platine diente in der Hauptsache zum Testen der bis dahin entwickelten Schaltung, um Probleme zu identifizieren und in der zweiten Version auszubessern. Diese erste Version verfügte noch nicht über einen Beschleunigungssensor, so dass nur kurze Testflüge möglich waren bis die Gyrodreift zu einer zu starken Schräglage führte. Dieses Problem war bereits beim Entwurf bekannt, wurde aber in Kauf genommen, um zur weiteren Planung Erfahrung mit den Eigenschaften der eingesetzten Hardware zu sammeln.

Abbildung 3.1.1 zeigt diese erste Platine. An die Molexbuchse unten links wird die 12 Volt Spannungsversorgung angeschlossen. Die vier Molexbuchsen rechts daneben sind die Leistungsausgänge für die Motoren. Darüber sind die vier Leistungs-MOSFETs sichtbar, welche den Motorstrom schalten, sowie Pulldown-Widerstände, die einen definierten Gate-Pegel an den MOSFETs garantieren (siehe Unterabschnitt 3.6.2). Weiter oberhalb befindet sich die zehnpolige Debug-Buchse, die zur Ausgabe von Debug-Nachrichten und zur Programmierung des

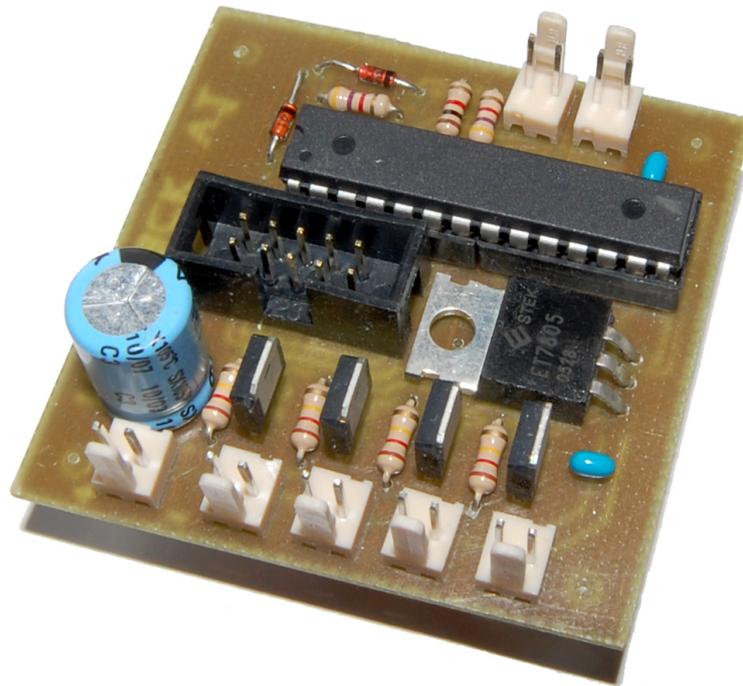


Abbildung 3.1.1.: Die erste Version der Steuerplatine.

Mikrocontrollers verwendet werden kann. Rechts daneben befindet sich ein 5 Volt Spannungsregler 7805 im TO-220-Gehäuse. Darüber befindet sich wiederum der gesockelte Mikrocontroller *ATmega168*, und ganz oben sind die zwei Molexbuchsen zu sehen, an denen die Wii MotionPlus angeschlossen wird. Die Widerstände und Dioden links davon senken die Versorgungsspannung für die MotionPlus auf 3,6 Volt. Zwei der Widerstände sind als Pullup-Widerstände mit den I<sup>2</sup>C-Busleitungen SCL und SDA verbunden (siehe Unterabschnitt 3.3.3).

Für die zweite Version, die in Abbildung 3.1.2 gezeigt ist, wurde das Platinenlayout optimiert, um in der Mitte der Platine Platz für den neu eingeplanten Beschleunigungssensor zu schaffen. Dieser wird in die dort sichtbaren Pinheader-Buchsen eingesteckt. Die drei rechts oberhalb davon eingesetzten Widerstände bilden mit den drei Kondensatoren rechts einen Tiefpassfilter für jeden der drei analogen Ausgänge des Beschleunigungssensors. Die Details dazu sind in Abschnitt 3.4 beschrieben. Links vom Beschleunigungssensor ist eine Gruppe aus zwei Dioden, einem Widerstand und einem Kondensator zu sehen, welche die Spannung für den Sensor auf 3,6 Volt absenkt und im Aufbau der Schaltung gleicht, die auf der ersten Platine für die Wii MotionPlus eingesetzt wurde.

Die MotionPlus wird nun ohne Verwendung weiterer Bauteile mit dem Mikrocontroller verbunden, da festgestellt wurde, dass sie über einen internen Spannungsregler verfügt und daher problemlos mit 5 Volt betrieben werden kann. Als Pullup-Widerstände für den I<sup>2</sup>C-Bus werden nun die internen Pullup-Widerstände des Mikrocontrollers verwendet. Der Anschluss erfolgt weiterhin über zwei Molexbuchsen, die sich nun rechts auf der Platine befinden.

### 3. Entwicklung einer Steuerplatine

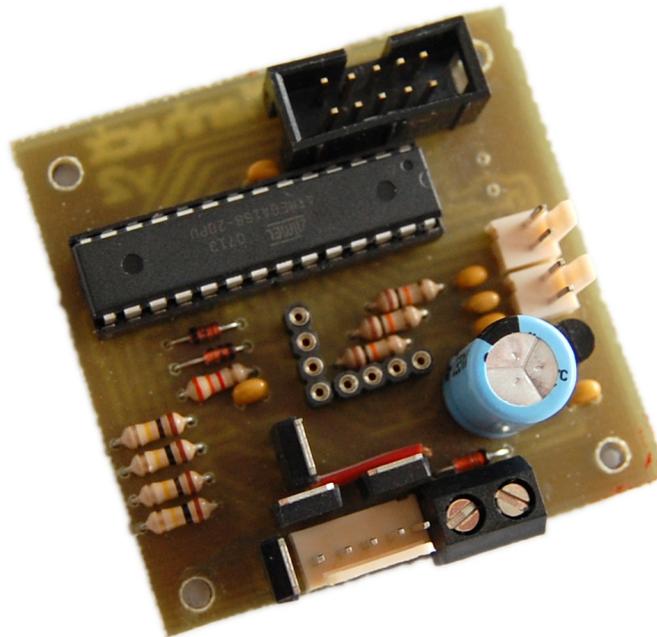


Abbildung 3.1.2.: Die Steuerplatine in der zweiten Version, die bis zum Abschluss der Arbeit verwendet wurde.

Zwischen der 12 Volt Spannungsversorgung und dem 5 Volt Spannungsregler wurde eine Diode eingesetzt, die die Auswirkungen von Spannungseinbrüchen im 12 Volt-Bereich auf den 5 Volt-Bereich der Schaltung vermindern soll. Solche Spannungseinbrüche wurden bei kurzzeitiger hoher Motorlast beobachtet. Abgesehen davon wird nun der kleinere und leichtere 78L05 im TO-92-Gehäuse als Spannungsregler verwendet, da diese Bauform für die benötigten Ströme ausreicht. Der Spannungsregler ist rechts unten zu sehen, teilweise verdeckt durch den Kondensator zur Glättung der Eingangsspannung.

Die Ausgangsbuchsen für die Motoren wurden zu einer einzigen fünfpoligen Buchse mit einem kombinierten +12 Volt-Ausgang für alle Motoren zusammengefasst, um Platz zu sparen und Verwechslungen in der Reihenfolge der Anschlüsse zu verhindern. Neben der Buchse, die im Bild unten zu sehen ist, befindet sich nun eine Schraubklemme für den Anschluss der Stromversorgung, die gegenüber der vorher verwendeten Molexverbindung einen robusteren Kontakt mit geringerem Widerstand ermöglicht.

Schließlich wurde die Position der Transistoren geändert, um die Leiterbahnen für den Motorstrom möglichst kurz zu gestalten und so Verluste zu verringern. Die Pulldown-Widerstände wurden dazu an den linken Rand der Platine verlegt.

Nach dem Bau dieser zweiten Version waren weitere Korrekturen nötig, die jedoch direkt an der Platine durchgeführt wurden. Dabei handelt es sich um Änderungen am Anschluss des Beschleunigungssensors, die in Anhang A genauer beschrieben sind. Um die Korrekturen zu

berücksichtigen, wurde eine dritte Version der Steuerplatine entworfen, die zusätzlich kleinere Optimierungen des Platinenlayouts enthält. Diese wurde jedoch nicht gebaut, da sie in der Funktion identisch zur geänderten zweiten Version ist. Schaltplan und Platinenlayout der dritten Version sind im Anhang abgebildet und befinden sich für alle drei Versionen auf der CD, die dieser Arbeit beiliegt.

## 3.2. Mikrocontroller: ATmega168

Für die Regelung und Steuerung des Quadropters wurde der Mikrocontroller *ATmega168* der Firma Atmel ausgewählt, dessen Eigenschaften in [Atm09] beschrieben sind. Die Mikrocontroller der ATmega-Reihe enthalten eine 8 Bit CPU, die je nach Taktquelle mit bis zu 20 MHz betrieben werden kann. Das Modell ATmega168 enthält 1 KB RAM zur Ablage von zur Laufzeit veränderlichen Daten sowie 16 KB Flash-Speicher zur Ablage von Programmcode.

Neben CPU und Speicher enthält der Mikrocontroller diverse Hardwarekomponenten, die für Aufgaben wie Steuerung, Kommunikation und Zeiterfassung nötig sind. 23 der 28 Pins des ATmega168 können als generische digitale Ein- und Ausgänge verwendet werden. Zusätzlich können alle I/O-Pins für vom Pin abhängige alternative Funktionen verwendet werden, z.B. als analoge Eingangsleitung des im ATmega integrierten Analog-Digital-Wandlers oder als Sendeleitung für die serielle Schnittstelle.

Der ATmega168 verfügt über ausreichende Rechenleistung und Hardware für den Einsatz auf der Steuerplatine. Durch die Verwendung dieser Familie von Mikrocontrollern in verschiedenen Veranstaltungen der Universität Koblenz ist sowohl die notwendige Technik als auch Erfahrung für die Programmierung dieses Mikrocontrollers vorhanden. Im Gegenzug eignet sich der Quadropter durch den Einsatz eines Chips aus dieser Familie auch gut als anschauliches Demonstrationsobjekt für den Einsatz in den erwähnten Lehrveranstaltungen. Mit einem Preis von aktuell ca. 3,40 Euro passt der Mikrocontroller in das Konzept, eine möglichst preisgünstige Steuerung zu entwerfen.

### 3.2.1. Verwendete Hardwaremodule

Im folgenden werden die verwendeten Hardwaremodule des Mikrocontrollers zusammen mit ihrer Verwendung auf der Steuerplatine beschrieben. Eine detailliertere Beschreibung der Hardware befindet sich in [Atm09]. Für ein genaueres Verständnis der Verwendung kann der kommentierte Quellcode des Projekts betrachtet werden, der auf der beiliegenden CD zur Verfügung steht.

Auf der Steuerplatine wird der ATmega ohne externen Taktgeber betrieben, da auf diese Weise weniger Bauteile nötig sind. Statt dessen wird der *interne RC-Oszillator* mit der höchsten einstellbaren Frequenz von 8 MHz benutzt. Im Vergleich zu einem Betrieb mit externem 20

### 3. Entwicklung einer Steuerplatine

MHz-Quarz ist die Rechengeschwindigkeit damit deutlich geringer. Während der Entwicklung zeigte sich jedoch, dass die Rechenleistung bei 8 MHz für die Regelung des Quadropters mit den verwendeten Techniken ausreicht.

Ein weiterer beachtenswerter Nachteil des internen RC-Oszillators ist die geringe Genauigkeit, die im Datenblatt mit  $\pm 10\%$  angegeben ist. Abweichende Taktraten machen sich insbesondere beim Erfassen von Zeitabständen bemerkbar, z.B. beim Messen der Pulslänge des Signals vom Fernsteuerungsempfänger. Die Berechnung der aktuellen Ausrichtung aus den Messwerten des Gyrosensors und die Reaktionen des PID-Reglers sind davon ebenfalls betroffen. Zur Verringerung dieser Abweichungen kann der RC-Oszillator kalibriert werden. Aus dem Datenblatt geht hervor, dass die Genauigkeit des Oszillators durch eine solche Kalibrierung auf  $\pm 1\%$  verbessert werden kann.

Der ATmega168 verfügt über einen *Analog-Digital-Wandler* mit einer Auflösung von 10 Bit, dessen Eingang zwischen sechs Pins umgeschaltet werden kann. Das ermöglicht es, die Spannung auf verschiedenen analogen Leitungen nacheinander zu messen. Am AREF-Pin kann eine externe Referenzspannung angelegt werden, welche das obere Ende der Skala für die Messungen festlegt. Auf der Steuerplatine wird der A/D-Wandler verwendet, um die Spannung der drei analogen Ausgänge des Beschleunigungssensors zu erfassen.

Die drei *Timer* des Mikrocontrollers ermöglichen eine zuverlässige Erfassung von Zeitabständen, können aber auch für die Erzeugung von PWM-Signalen verwendet werden. Timer 0 und 2 verfügen über einen Wertebereich von 8 Bit, Timer 1 kann bis zu 16 Bit verwenden. Timer 0 und 1 werden ausschließlich zur Erzeugung der PWM-Signale benutzt, mit welchen die Motoren des Quadropters gesteuert werden. Genauere Ausführungen zu dieser Anwendung befinden sich in Unterabschnitt 3.6.3.

Timer 2 wird dagegen für die Messung von Zeitabständen eingesetzt. Zu diesem Zweck ist der Timer so eingestellt, dass er alle 64 CPU-Zyklen inkrementiert wird, was einer Zeitauflösung von 8 Mikrosekunden entspricht. Bei einem Überlauf des 8 Bit Wertebereichs wird ein Interrupt ausgelöst, in welchem eine 16 Bit Zählvariable inkrementiert wird. Dadurch können Zeitspannen bis  $2^{24}$  Timerzyklen erfasst werden, was ca. 134 Sekunden entspricht.

Die *serielle Schnittstelle (UART)* kann für die Kommunikation des Mikrocontrollers mit einem PC verwendet werden. Über diese Schnittstelle können Debug-Nachrichten an den PC gesendet werden. So wurde z.B. in einer Testversion der Steuersoftware regelmäßig der Messwert des Beschleunigungssensors übertragen, um sicherzustellen, dass die erwarteten Werte vom Sensor empfangen werden. Die I/O-Pins, die für die serielle Schnittstelle vorgesehen sind, werden in der finalen Version der Steuerplatine allerdings als generische I/O-Pins für den Anschluss des Empfängers der Fernsteuerung eingesetzt, so dass im normalen Betrieb kein Debuggen mehr möglich ist. Die Verwendung der Leitungen in diesem Zusammenhang ist in Abschnitt 3.5 erläutert.

Über die *Serial Downloading-Schnittstelle*, die zusammen mit der seriellen Schnittstelle an der zehnpoligen Debug-Buchse ausgeführt ist, kann der Flash-Programmspeicher des ATmega direkt auf der Platine neu programmiert werden.

Zur Kommunikation mit der Wii MotionPlus kommt das *TWI* (Two-Wire Interface) zum Einsatz, welches den Anschluss des Mikrocontrollers an einen *I<sup>2</sup>C-Bus* ermöglicht. Eine kurze Übersicht dieses Bussystems befindet sich in Unterabschnitt 3.3.3.

## 3.3. Gyrosensor: Wii MotionPlus

Die Wii MotionPlus ist ein Zusatzgerät für die Spielkonsole Wii von Nintendo. Sie erweitert die bewegungsempfindliche Wii Remote um Gyrosensoren, die die Winkelgeschwindigkeit um alle drei Achsen erfassen können. Dadurch können Spiele, die diese Erweiterung benutzen, die Bewegung und räumliche Lage der Wii Remote genauer bestimmen als solche, die sich nur auf den direkt in der Wii Remote vorhandenen dreiachsigen Beschleunigungssensor verlassen.

Insbesondere ist es durch die Kombination beider Sensoren möglich, über kurze Zeiträume sowohl die räumliche Position als auch die Ausrichtung der erweiterten Wii Remote zu verfolgen. Da hierzu die Sensordaten über die Zeit integriert werden müssen — bei der Berechnung der Position aus der Beschleunigung sogar doppelt — sammeln sich jedoch nach kurzer Zeit Messabweichungen an, so dass eine regelmäßige Kalibrierung notwendig ist.[IOI<sup>+</sup>]

Da für die MotionPlus keine detaillierte Dokumentation öffentlich zur Verfügung gestellt wird, wurde sie wie auch andere Peripheriegeräte für die Wii von Bastlern untersucht. Viele Ergebnisse wurden auf Blogs und in Internetforen veröffentlicht. Eine Stelle, an der so gewonnene Erkenntnisse zusammengetragen werden, ist das WiiBrew Wiki[wii]. Auf der Seite „Wiimote/Extension Controllers/Wii Motion Plus“[Wii11b] sind dort viele Details zum internen Aufbau und den Schnittstellen der MotionPlus beschrieben.

Die MotionPlus wurde bereits in mehreren Projekten zum Erfassen von Winkelgeschwindigkeiten eingesetzt. Als Beispiel sei hier ein Headtracking-System (vgl. [HHS<sup>+</sup>10]) genannt. Sie wird auch als Sensor auf Quadrokoptern eingesetzt[mul].

Dank dem vergleichsweise geringen Preis von aktuell ca. 18 Euro sowie der einfachen Verfügbarkeit in vielen Elektronikgeschäften ist die MotionPlus für den Einsatz in Hobbyprojekten interessant. Durch das geringe Gewicht eignet sie sich auch als Sensor für Modellfluggeräte. Zum Anschluss an einen Mikrocontroller sind keine zusätzlichen Bauteile erforderlich, und der Austausch von Daten erfolgt über das von vielen Mikrocontrollern unterstützte *I<sup>2</sup>C*-Protokoll. Die Nachteile sollten allerdings ebenfalls bedacht werden. So ist es möglich, mit anderen Sensoren eine leichtere und insbesondere deutlich genauere Lösung zu konstruieren. Aufgrund der oben genannten Vorteile und vorhandener Erfahrung mit dem Einsatz der MotionPlus wurde diese jedoch für das Quadrokopterprojekt ausgewählt.

### 3. Entwicklung einer Steuerplatine

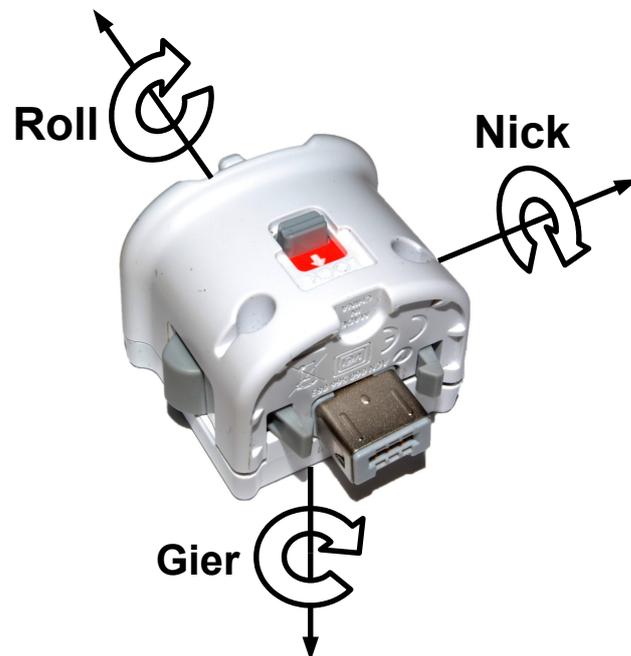


Abbildung 3.3.1.: Ansicht auf das Gehäuse der MotionPlus. Die gezeigten Rotationsachsen folgen den Bezeichnungen aus [Wii11b], die auch in Unterabschnitt 3.3.6 verwendet werden. [Moo09] folgt einer anderen Konvention, in der Nick- und Rollachse vertauscht sind.

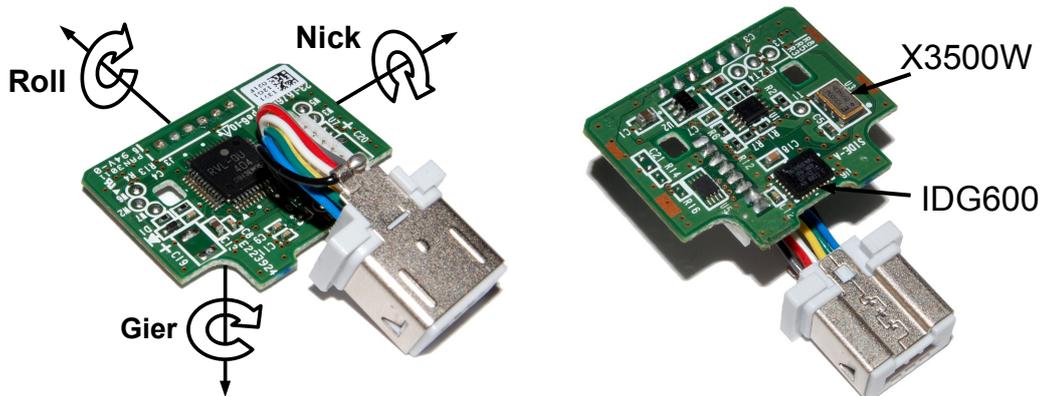


Abbildung 3.3.2.: Ober- und Unterseite der Platine der MotionPlus. Die Buchse zum Anschluss weiterer Erweiterungen wurde hier bereits entfernt. Sie war ursprünglich mit einem Flachbandkabel an den sechs Lötstellen oben links im Bild befestigt.

### 3.3.1. Aufbau und Funktionsweise

Im Gehäuse der Wii MotionPlus ist eine beidseitig mit SMD-Bauteilen bestückte Platine untergebracht, die an die beiden von außen sichtbaren Steckverbindungen angeschlossen ist. Auf der Oberseite der Platine befindet sich ein IC (U4) mit der Beschriftung *RVL-GU 404*. Dabei handelt es sich laut [Wii11b] um einen Mikrocontroller vom Typ *NEC 78K0/KC2*. Eine Untersuchung der Verbindungen dieses Bauteils bestätigt diese Angabe, so sind z.B. die analogen Ausgänge der Gyrosensoren mit Pins verbunden, die im Datenblatt [NEC07] des *NEC 78K0/KC2* als analoge Eingänge ausgewiesen sind, und einer der beiden nach außen geführten I<sup>2</sup>C-Ports verwendet die SCL- und SDA-Pins des Mikrocontrollers.

Die Unterseite der Platine enthält die beiden Gyrosensoren. Der mit U6 bezeichnete IC ist ein zweiachsiger Gyrosensor vom Typ *Invensense IDG-600*. Dieser misst die Winkelgeschwindigkeit um die Nick- und Rollachse. Er besitzt für jede Achse zwei analoge Ausgänge mit unterschiedlicher Empfindlichkeit (vgl. Informationen zum verwandten IDG-650 in [Inv]). Messungen auf der Platine zeigen, dass diese Ausgänge direkt mit analogen Eingängen zum A/D-Wandler des Mikrocontrollers verbunden sind. Der Grund für die Verwendung von zwei Ausgängen pro Achse liegt darin, dass der A/D-Wandler nur eine begrenzte Auflösung besitzt. Würde nur der unempfindliche Ausgang ausgewertet, könnten kleine Winkelgeschwindigkeiten nur ungenau gemessen werden. Der empfindliche Ausgang erlaubt genauere Messungen, kann aber keine großen Winkelgeschwindigkeiten darstellen. Durch die Auswahl eines Ausganges je nach Geschwindigkeitsbereich können die Vorteile beider Möglichkeiten kombiniert werden [IOI<sup>+</sup>, Seite 2].

### 3. Entwicklung einer Steuerplatine

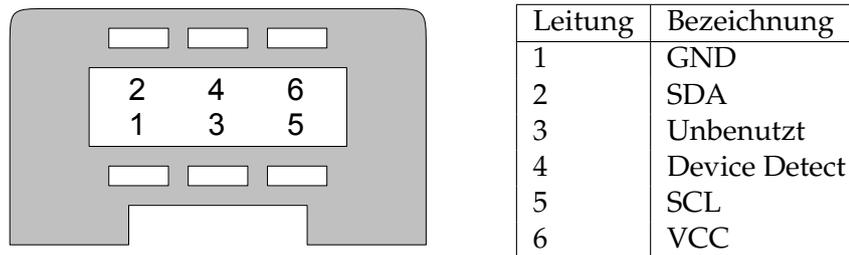


Abbildung 3.3.3.: Belegung der Leitungen des Steckers, mit dem die MotionPlus an die Wii Remote angeschlossen wird (Sicht von vorne auf den Stecker). Die Nummerierung entspricht der Anordnung der Kontakte auf der Platine der MotionPlus. Informationen zur Belegung stammen von [Wii11a] sowie [Moo09].

U3 trägt die Bezeichnung *X3500W* und wird von Epson Toyocom hergestellt. Es handelt sich dabei um einen einachsigen Gyrosensor, der die Winkelgeschwindigkeit um die Gierachse misst, die senkrecht zur Platinebene steht. Da für diesen Chip kein öffentliches Datenblatt existiert, wird in [HHS<sup>+</sup>10, Seite 11] auf das Datenblatt des ähnlich benannten XV-3500CB des gleichen Herstellers verwiesen, das auf der Webseite des Herstellers zu finden ist ([Eps]). Es sollte jedoch beachtet werden, dass die MotionPlus auf der Achse, die von diesem Chip erfasst wird, Winkelgeschwindigkeiten messen kann, die deutlich über den im Datenblatt des XV-3500CB spezifizierten 100° pro Sekunde liegen.

Der *X3500W* besitzt nur einen Analogausgang für die gemessene Achse, dieser wird jedoch durch eine externe Beschaltung verstärkt und ebenfalls in einen empfindlichen und einen unempfindlichen Kanal aufgeteilt.

#### 3.3.2. Externe Schnittstellen

Die beiden nach außen sichtbaren Verbindungen in Form eines Steckers und einer Buchse dienen der Kommunikation sowie der Stromversorgung. Dabei ist der Stecker für die Verbindung zur Wii Remote vorgesehen. Die gegenüberliegende Buchse besitzt die gleiche Pinbelegung wie der Stecker und kann zum Anschließen einer zusätzlichen Erweiterung (z.B. des Nunchuk-Controllers) benutzt werden.

Die Leitungen GND und VCC sowie die in der MotionPlus nicht verwendete Leitung 3 besitzen eine direkte elektrische Verbindung zwischen Stecker und Buchse. Alle anderen Leitungen sind an jeweils unterschiedliche I/O-Pins des internen Mikrocontrollers angeschlossen. Diese Angaben wurden mit Hilfe eines Durchgangsprüfers ermittelt.

Die Leitungen GND und VCC werden für die Stromversorgung benutzt. Die Wii Remote liefert zwischen diesen Leitungen eine Spannung von 3,3V. Die „Device Detect“-Leitung dient der Erkennung, ob eine Erweiterung angeschlossen ist. Leitung 3 ist abgesehen von der Durchverbindung zwischen Stecker und Buchse nicht innerhalb der MotionPlus verbunden.

Der Datenaustausch erfolgt über einen I<sup>2</sup>C-Bus, der über die beiden Leitungen SCL und SDA ausgeführt ist. Da dieses Busprotokoll von vielen Mikrocontrollern direkt unterstützt wird<sup>1</sup>, ist die Kommunikation mit Erweiterungen wie der MotionPlus mit nur geringem Aufwand möglich. Die SCL- und SDA-Leitungen des Steckers sind in der MotionPlus mit der I<sup>2</sup>C-Schnittstelle des Mikrocontrollers verbunden.

#### 3.3.3. Der I<sup>2</sup>C-Bus

Ein I<sup>2</sup>C-Bus verbindet mehrere Geräte über eine Clock-Leitung (SCL) und eine Datenleitung (SDA). Die angeschlossenen Geräte verwenden dabei Open-Collector- bzw. Open-Drain-Ausgänge, die die Busleitungen nur auf einen Low-Pegel ziehen können, nicht aber auf einen High-Pegel. Damit ein High-Pegel möglich ist, müssen beide Leitungen über Pullup-Widerstände mit der gewünschten Spannung verbunden werden. Diese Verschaltung entspricht einer Und-Verknüpfung aller Ausgänge, da eine Busleitung nur dann auf High liegt, wenn kein Busteilnehmer die Leitung auf Low zieht.

Die Kommunikation wird immer von einem „Master“ initiiert, der den gewünschten Busteilnehmer über eine Geräteadresse identifiziert und diesem mitteilt, ob er Daten senden oder empfangen will. Daraufhin werden Datenbytes in die gewünschte Richtung übertragen. Der Empfänger bestätigt den Empfang, sobald er das letzte erwartete Byte erhält. Bei der Übertragung wird das Clock-Signal immer vom Master generiert, allerdings kann der Slave die Übertragung bremsen, indem er die Low-Phase des Clock-Signals verlängert.[Atm09, Abschnitt 21][Wik12a][HHS<sup>+</sup>10, Seite 16]

Der I<sup>2</sup>C-Standard legt verschiedene Standard-Datenraten fest. Allerdings ist unklar, ob die MotionPlus für eine Übertragung mit 400kHz oder nur mit 100kHz ausgelegt ist. In unterschiedlichen Quellen finden sich dazu unterschiedliche Angaben (siehe [Moo09]). Eine Verbindung mit niedrigerer Geschwindigkeit kann helfen, Übertragungsprobleme zu vermeiden.

#### 3.3.4. Elektrische und mechanische Verbindung

Für eine Benutzung der MotionPlus als Teil einer Quadrocoptersteuerung ist es empfehlenswert, das Plastikgehäuse zu entfernen und nur die enthaltene Platine zu benutzen, da das Gehäuse einen Großteil des Gewichts ausmacht.

Die Befestigung sollte so erfolgen, dass sich die Ausrichtung der MotionPlus gegen den Rest des Quadrocopters möglichst wenig verändert. Auf diese Weise werden Messfehler vermieden. Eine Ausrichtung entlang der Achsen des Quadrocopters ist dabei vorteilhaft, weil so der Rechenaufwand für die Transformation der gemessenen Winkelgeschwindigkeiten in das Koordinatensystem des Quadrocopters gering bleibt.

<sup>1</sup>In Mikrocontroller-Datenblättern wird der I<sup>2</sup>C-Bus oft auch als TWI (Two-Wire Interface) bezeichnet.

### 3. Entwicklung einer Steuerplatine

Um die Messdaten der MotionPlus abzufragen, kann der eigentlich zur Verbindung mit der Wii Remote vorgesehene Stecker benutzt werden. Da die gegenüberliegende Buchse zur Durchleitung der Verbindung an andere Erweiterungen gedacht ist und daher bei der Verwendung als einzelnes Sensormodul nicht benötigt wird, kann man diese Buchse ausbauen und als Adapter verwenden, der auf den Stecker aufgesteckt werden kann<sup>2</sup>. Diese Art der Verbindung ermöglicht ein einfaches Austauschen der MotionPlus, so dass diese auch für andere Versuche benutzt werden kann. Eine direkte Lötverbindung spart allerdings zusätzliches Gewicht ein und stellt einen stabileren elektrischen Kontakt her.

Von den am Stecker ausgeführten Leitungen werden nur GND, VCC, SDA und SCL benötigt. SDA und SCL können direkt mit den entsprechenden Pins des ATmega verbunden werden. Ein I<sup>2</sup>C-Bus benötigt Pullup-Widerstände an beiden Leitungen, die nicht in der MotionPlus enthalten sind und daher extern zur Verfügung gestellt werden müssen. Die I/O-Pins des ATmega168 verfügen allerdings über eingebaute Pullup-Widerstände, so dass eine Verbindung ohne zusätzliche Bauteile möglich ist.

Für die Stromversorgung sollte idealerweise genau wie bei der Versorgung durch die Wii Remote eine Spannung von 3,3V benutzt werden. Allerdings scheint auch ein Betrieb mit 5V keine Probleme zu verursachen.<sup>3</sup> Der Grund dafür ist, dass die Versorgungsspannung für die beiden Gyrosensoren in der MotionPlus durch einen Spannungsregler (auf der Platine mit U2 bezeichnet) auf 3 Volt geregelt wird. Der eingebaute Mikrocontroller NEC 78K0/KC2 erhält zwar die volle Versorgungsspannung, kann laut Datenblatt ([NEC07, Seite 582]) aber mit bis zu 6,5 Volt betrieben werden.

Diese Beobachtungen sind zwar kein vollständiger Beleg dafür, dass die MotionPlus auch mit 5V fehlerfrei funktioniert, aber im Rahmen dieser Arbeit sind in der Praxis keine Probleme damit aufgetreten, und auch andere Projekte benutzen diese Versorgungsspannung (vgl. [HHS<sup>+</sup>10, Seite 11] sowie [Moo09]). Eine Verschlechterung des Messverhaltens wurde ebenfalls nicht beobachtet.

Da die Leistungsaufnahme der MotionPlus nur wenige Milliampere beträgt, kann die Versorgungsspannung über einen generischen I/O-Pin des ATmega zur Verfügung gestellt werden. Dadurch erhält man die Möglichkeit, die MotionPlus softwaregesteuert aus- und einzuschalten. Wie oben erläutert ist es nicht notwendig, die Spannung auf 3,3 Volt abzusenken. Die MotionPlus kann also mit allen benötigten Leitungen direkt und ohne weitere Bauteile an den ATmega angeschlossen werden.

---

<sup>2</sup>Dadurch verliert man allerdings die Möglichkeit, die MotionPlus auf einfache Weise mit anderen Erweiterungen wie z.B. einem Nunchuk-Controller zu kombinieren. Diese Kombination erlaubt es, durch Verwendung eines speziellen Abfragemodus sowohl die Gyrodaten der MotionPlus als auch die Beschleunigungsmessungen des Nunchuk-Controllers über den I<sup>2</sup>C-Port der MotionPlus abzufragen. Unterabschnitt 3.3.5 enthält weitere Details dazu.

<sup>3</sup>Das gilt zumindest für das in dieser Arbeit verwendete Modell. Der Aufbau von anderen Modellen oder Nachbageräten von Drittherstellern könnte abweichen.

### 3.3.5. Initialisierung und Aktivierung

Manche Spiele für die Wii unterstützen die MotionPlus nicht, dafür aber eine andere Erweiterung. Um zu gewährleisten, dass solche Spiele ihnen bekannte Erweiterungen auch dann nutzen können, wenn diese im Erweiterungsport der MotionPlus anstatt direkt an der Wii Remote angeschlossen sind, arbeitet die MotionPlus im initialen deaktivierten Zustand zunächst als Repeater. Alle Anfragen an die normalerweise von Erweiterungen benutzte I<sup>2</sup>C-Geräteadresse 82 (0x52) werden an den Erweiterungsport weitergeleitet. Dabei agiert der Mikrocontroller der MotionPlus als Vermittler zwischen den beiden Schnittstellen, da diese nicht direkt miteinander verbunden sind.

Damit es trotzdem möglich ist, Nachrichten direkt an die MotionPlus zu richten, belegt diese im deaktivierten Zustand die Geräteadresse 83 (0x53) und wechselt erst nach einer Aktivierung auf die Adresse 0x52 ([Wii11b],[Moo09]). Die Aktivierung wird durch das Senden der Bytefolge 0xfe, 0x04 an die Geräteadresse 0x53 durchgeführt. Die MotionPlus wechselt daraufhin die Geräteadresse auf 0x52 und initialisiert sich.[Moo09].

In [Wii11b] wird vor dieser Aktivierung noch eine explizite Initialisierung durchgeführt, indem die Bytefolge 0xf0, 0x55 an die Geräteadresse 0x53 gesendet wird. Dieser zusätzliche Schritt scheint aber kein unterschiedliches Verhalten auszulösen und wird auch in [HHS<sup>+</sup>10] nicht durchgeführt. Es ist jedoch denkbar, dass andere Modelle oder Nachbaugeräte bei der Aktivierung keine automatische Initialisierung durchführen, so dass dieser zusätzliche Schritt notwendig wäre.

Da die MotionPlus im aktivierten Zustand die gleiche Geräteadresse benutzt wie eine normale Erweiterung, ist es nicht direkt möglich, die MotionPlus und eine zusätzlich angeschlossene Erweiterung gleichzeitig zu benutzen. Aus diesem Grund unterstützt die MotionPlus einen Durchleitungsmodus, bei dem die Abfrage des aktuellen Zustands (siehe unten) abwechselnd ein Statuspaket der MotionPlus und der angeschlossenen Erweiterung zurückgibt.

Die durchgeleiteten Pakete werden dabei von der MotionPlus modifiziert, um zusätzliche Informationen darin unterzubringen und um zu kennzeichnen, dass das Paket von der Erweiterung stammt und nicht von der MotionPlus. Um diese zusätzlichen Informationen in das Paket zu integrieren, ohne dessen Länge zu verändern, müssen vorhandene Bits überschrieben werden. Damit dabei keine wichtigen Informationen verloren gehen, muss der MotionPlus beim Aktivieren des Durchleitungsmodus die Art der angeschlossenen Erweiterung mitgeteilt werden. Die MotionPlus ändert das Format der durchgeleiteten Pakete abhängig davon so ab, dass nur weniger wichtige Informationen verloren gehen, z.B. die niederwertigsten Bits von analogen Messwerten.

Der Durchleitungsmodus wird ausgewählt, indem bei der Aktivierung der MotionPlus als zweites Byte statt 0x04 ein anderer Wert (je nach Art der angeschlossenen Erweiterung) gesendet wird (vgl. [Dog09]). Bisher sind zwei mögliche Werte bekannt: 0x05 zum Durchleiten

### 3. Entwicklung einer Steuerplatine

	Bit							
Byte	7	6	5	4	3	2	1	0
0	Gierachse (LSB)							
1	Rollachse (LSB)							
2	Nickachse (LSB)							
3	Gierachse (MSB)				Gier Slow		Nick Slow	
4	Rollachse (MSB)				Roll Slow		Erw. vorh.	
5	Nickachse (MSB)				1		0	

Tabelle 3.1.: Aufbau eines Datenpakets mit Messwerten von der MotionPlus. Die Messdaten jeder Achse sind jeweils in die oberen sechs Bits (MSB) sowie die unteren acht Bits (LSB) aufgeteilt. Die „Slow“-Bits geben an, ob der langsame oder schnelle Ausgang des entsprechenden Gyrosensors gemessen wurde. Das „Erweiterung Vorhanden“-Bit ist gesetzt, wenn eine zusätzliche Erweiterung angeschlossen ist. Die Informationen stammen aus [Wii11b].

von Daten des Nunchuk-Controllers, und 0x07 für den Classic Controller. Genauere Informationen, insbesondere über das Format der durchgeleiteten Pakete, sind in [Wii11b] zu finden.

#### 3.3.6. Abfrage der Messwerte

Die aktuellen Messdaten können abgefragt werden, indem 0x00 an die aktivierte MotionPlus an Geräteadresse 0x52 gesendet wird. Dann kann von der gleichen Geräteadresse ein Datenblock von sechs Bytes gelesen werden.

Die Struktur des empfangenen Datenblocks ist in Tabelle 3.1 gezeigt. Dabei ist zu beachten, dass die MotionPlus für jede Achse anhand der Winkelgeschwindigkeit<sup>4</sup> entscheidet, ob der Messwert des empfindlichen oder des unempfindlichen Kanals für diese Achse übertragen wird. Diese Entscheidung wird im „Slow“-Bit der jeweiligen Achse mitgeteilt, das gesetzt ist, falls der Messwert des empfindlichen Kanals übertragen wurde.

Die gelesenen Rohwerte sind (innerhalb der beiden Geschwindigkeitsbereiche) proportional zur Ausgabespannung der Gyrosensoren, die sich wiederum linear zur Winkelgeschwindigkeit verhält. Damit hängen die Rohwerte linear von der Winkelgeschwindigkeit ab und können über eine lineare Gleichung in die gewünschte Einheit (z.B. rad/s) umgerechnet werden:

$$\omega = k \cdot (\text{rohwert} - \text{offset})$$

Sowohl  $k$  als auch  $\text{offset}$  sind dabei für die beiden Geschwindigkeitsbereiche unterschiedlich.  $k$  unterscheidet sich nur wenig zwischen den Achsen und zwischen verschiedenen individuellen Geräten, so dass hier ein fester Wert für jeden Geschwindigkeitsbereich ausreichend genaue Ergebnisse für viele Anwendungszwecke liefert. Tabelle 3.2 zeigt gemessene Werte für  $k$ .

<sup>4</sup>Eine Untersuchung der Maximal- und Minimalwerte in jedem Modus legt nahe, dass die Entscheidung anhand des Messwerts am empfindlicheren Ausgang getroffen wird.

Modus	Ergebnis aus [Xev]	Eigene Messung
Fast	$0,003967 \text{ rad/s} \pm 5,7\%$	
Slow	$0,000881 \text{ rad/s} \pm 2,5\%$	$0,000881 \text{ rad/s} \pm 0,2\%$

Tabelle 3.2.: Umrechnungsfaktor  $k$  für den langsamen und schnellen Modus. Für die eigene Messung wurde die MotionPlus auf einer Tischplatte achtmal in abwechselnder Richtung um  $360^\circ$  um die Gierachse gedreht. Dabei wurden die offsetbereinigten Messwerte integriert. Die absolute Differenz zwischen dem integrierten Messwert vor und nach der Rotation entspricht  $360^\circ$ . Alle so berechneten Messwerte lagen innerhalb der angegebenen Unsicherheit. Die größere Unsicherheit der Messung aus [Xev] ergibt sich daher, dass dort mehrere MotionPlus-Module und alle Achsen untersucht wurden.

Eine allgemeine Vorgabe der Offset-Werte ist dagegen nicht möglich, da diese sich stark von Gerät zu Gerät sowie auch zwischen den Achsen unterscheiden. Der Offset ist auch von externen Faktoren wie der Umgebungstemperatur abhängig und sollte daher vor jedem Einsatz neu kalibriert werden. Das ist für den Slow-Modus auf einfache Weise möglich, da der Offset den gemessenen Rohwerten im Stillstand entspricht. Um ein möglichst genaues Ergebnis zu erhalten, ist es empfehlenswert, für die Kalibrierung den Mittelwert aus mehreren Messungen zu verwenden. Wenn man zusätzlich die Standardabweichung aus den Messungen errechnet, kann man Störungen erkennen und die Kalibrierung ggf. erneut durchführen. Da sich die Offsetwerte kurz nach dem Aktivieren der MotionPlus noch relativ stark ändern, ist es empfehlenswert, einige Sekunden vor dem Durchführen der Kalibrierung zu warten. Auf der Steuerplatine wird die Kalibrierung nach acht Sekunden durchgeführt, in [Xev] wird allerdings berichtet, dass die Stärke der Offsetschwankungen erst nach ein bis fünf Minuten nicht mehr deutlich abnimmt.

Der Fast-Offset lässt sich sowohl laut [Xev] als auch nach eigenen Untersuchungen nicht zuverlässig aus dem Slow-Offset bestimmen. Da die Umschaltung zwischen Slow- und Fast-Modus jedoch erst bei ca.  $360^\circ$  pro Sekunde erfolgt, wird der Fast-Modus beim Einsatz im Quadropter nur in Ausnahmefällen eine Rolle spielen. Für den Fast-Offset reicht daher bei diesem Einsatz eine grobe Schätzung aus. Dazu kann man z.B. den Slow-Offset auch als Fast-Offset verwenden.

Mit  $k$  und  $offset$  für den entsprechenden Geschwindigkeitsbereich kann man nun für alle Achsen die Winkelgeschwindigkeiten berechnen. Jeder ermittelte Wert entspricht einer Projektion der tatsächlichen Rotationsachse auf die Achse der Messung. Man kann daher die drei Winkelgeschwindigkeiten als Vektor interpretieren, dessen Richtung der tatsächlichen Rotationsachse entspricht, und dessen Länge die Winkelgeschwindigkeit um diese Achse angibt. Da die Berechnungen in der Steuersoftware das in Abschnitt 2.1 definierte Quadropter-Koordinatensystem voraussetzen, muss der gemessene Vektor in dieses Koordinatensystem transformiert werden. Da die MotionPlus auf dem X-Ufo so montiert wurde, dass ihre Ach-

### 3. Entwicklung einer Steuerplatine

sen entlang denen des Quadrokopter-Koordinatensystems liegen, ist dies jedoch durch Vertauschung und Gegenzahlbildung möglich.

## 3.4. Beschleunigungssensor: MMA7260QT

Als Beschleunigungssensor wurde der *Freescale MMA7260QT* ausgewählt. Dieser MEMS-Sensorchip misst Beschleunigungen auf allen drei Achsen und gibt die Ergebnisse als analoges Spannungssignal auf drei Ausgangsleitungen aus.

Alternativ wurde die Nunchuk-Erweiterung für die Nintendo Wii in Erwägung gezogen, die einen Beschleunigungssensor enthält und über den I<sup>2</sup>C-Bus abgefragt werden kann. Der MMA7260QT war allerdings zum Zeitpunkt der Planung der Steuerplatine günstiger erhältlich.

Die in diesem Abschnitt besprochenen Details sind zwar auf den MMA7260QT ausgerichtet, viele Informationen lassen sich aber auch auf andere Beschleunigungssensoren anwenden. Dazu gehört insbesondere die Beschreibung der Abfrage und Kalibrierung der Beschleunigungsdaten in Unterabschnitt 3.4.2.

### 3.4.1. Elektrische und mechanische Verbindung

Der MMA7260QT ist ein SMD-Baustein, der nicht ohne größeren Aufwand und Hilfsmittel auf eine Platine gelötet werden kann. Für die Quadrokoptersteuerung wurde daher eine fertige Platine verwendet, die den Chip enthält und dessen Pins zu Kontaktflächen im üblichen 2,54mm-Raster verbindet. An diese Kontaktflächen wurden Stiftleisten gelötet, um eine einfache Steckverbindung zur Steuerplatine zu ermöglichen.

Abgesehen von drei analogen Ausgängen  $X_{OUT}$ ,  $Y_{OUT}$  und  $Z_{OUT}$ , von denen jeder die aktuelle Beschleunigung auf einer Achse anzeigt, verfügt der MMA7260QT noch über zwei Pins  $V_{SS}$  (GND) und  $V_{DD}$  (+2,2V bis +3,6V) zur Spannungsversorgung, zwei digitale Eingänge g-Select1 und g-Select2 zur Auswahl der Messempfindlichkeit, sowie einen digitalen Eingang „Sleep Mode“, mit dem der Chip in einen Energiesparmodus versetzt werden kann.

Da der Sensor für die Regelung der Ausrichtung des Quadrokopters dauerhaft benötigt wird und der Verbrauch mit 0,5mA[Fre08] im Vergleich zur Leistungsaufnahme der Motoren vernachlässigbar gering ist, ist der Energiesparmodus für diese Anwendung nicht von Interesse. Der „Sleep Mode“-Pin wurde daher auf der Steuerplatine fest mit  $V_{DD}$  verbunden, um den Energiesparmodus dauerhaft zu deaktivieren.

Über die g-Select-Pins kann die Empfindlichkeit so eingestellt werden, dass maximale Beschleunigungswerte von 1,5g, 2g, 4g oder 6g erfasst werden können. Ein größerer erfassbarer

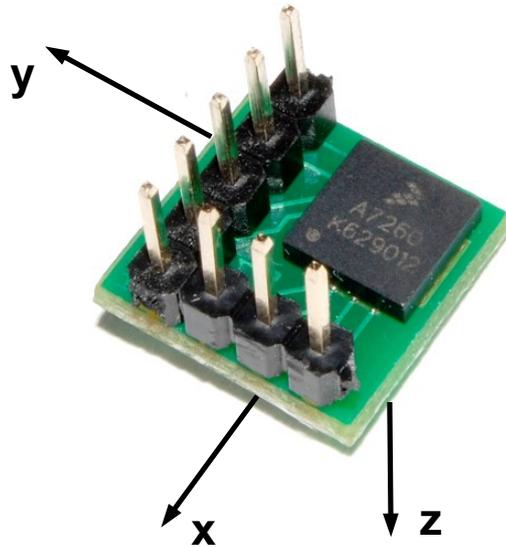


Abbildung 3.4.1.: Der Beschleunigungssensor auf der steckbaren Hilfsplatine. Die eingezeichneten Achsen entsprechen der Konvention aus dem Datenblatt des Sensors.

Wertebereich ermöglicht auch bei anspruchsvollen Flugmanövern sowie bei starken Vibrationen noch eine korrekte Erfassung der Beschleunigung, dieser Vorteil wird aber mit einer größeren Auflösung erkauft. Aufgrund der weiter unten beschriebenen Vibrationen, die auf dem Quadrocopter auftreten, musste die geringste Empfindlichkeit gewählt werden, um ein Übersteuern der Ausgänge zu verhindern. Dazu wurden die g-Select-Pins ebenfalls fest mit  $V_{DD}$  verbunden.

Die Stromversorgung sollte laut [Fre08] mit 2,2V bis 3,6V erfolgen. Die Spannung muss also zunächst von den für die restliche Elektronik verwendeten 5V abgesenkt werden. Als einfache Lösung dieses Problems wurde eine Reihenschaltung von Dioden eingesetzt, an denen ab einer gewissen Grundlast eine näherungsweise konstante Spannung abfällt. Die Grundlast wird dabei durch einen Widerstand erzeugt, weil die Spannung am Sensor sonst u.U. zu groß werden könnte, falls dieser zeitweise nur sehr wenig Strom benötigt. Ein zusätzlicher Kondensator zwischen  $V_{DD}$  und  $V_{SS}$  dient der Unterdrückung von Störeinflüssen in der Versorgungsspannung, die die Messung der analogen Spannungspegel an den Ausgängen des Sensors verfälschen oder die Funktion des Sensors beeinträchtigen könnten.

In [Fre08, Seite 5, Hinweis 1] wird vorgegeben, dass für den Einschaltvorgang ein schnelles Ansteigen der Spannung an  $V_{DD}$  erforderlich ist, so dass innerhalb von 0,1ms oder weniger eine Spannung von 2,2V erreicht wird. Um dies sicherzustellen wird der Beschleunigungssensor (wie auch die Wii MotionPlus) auf der Steuerplatine über einen I/O-Pin des ATmega versorgt. Durch die integrierte Einschaltverzögerung des ATmega ist die Versorgungsspannung

### 3. Entwicklung einer Steuerplatine

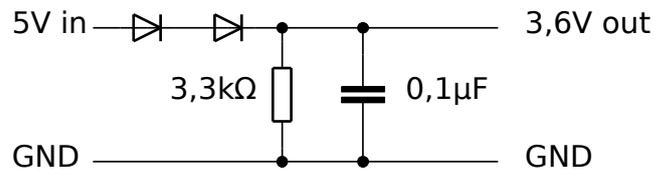


Abbildung 3.4.2.: Dies ist die auf der Steuerplatine verwendete Schaltung zum Absenken der Spannung. Die tatsächliche Ausgangsspannung ist von den verwendeten Dioden abhängig und sollte mit einem Messgerät überprüft werden. Da der MMA7260QT für einen relativ großen Spannungsbereich ausgelegt ist, sollte es mit beliebigen Dioden möglich sein, eine akzeptable Spannung zu erreichen, indem die Anzahl der in Reihe geschalteten Dioden angepasst wird.

bis zum Start der Programmausführung auf 5V stabilisiert, so dass ein ausreichend schneller Spannungsanstieg an  $V_{DD}$  erzeugt werden kann.

Die analogen Ausgänge  $X_{OUT}$ ,  $Y_{OUT}$  und  $Z_{OUT}$  liefern eine Spannung, die sich linear zur gemessenen Beschleunigung verhält. Diese Spannung wird allerdings von einem hochfrequenten Störsignal überlagert, das durch die interne Funktion des Sensorchips erzeugt wird und vor der A/D-Wandlung gefiltert werden sollte. Das Datenblatt des MMA7260QT empfiehlt dazu einen einfachen RC-Tiefpassfilter aus einem  $1k\Omega$ -Widerstand und einem  $0,1\mu F$ -Kondensator [Fre08, Seite 5, Hinweis 5]. Nach der Formel  $f_c = \frac{1}{2\pi R \cdot C}$  besitzt dieser Filter eine Grenzfrequenz  $f_c$  von ca. 1,6kHz. Die gefilterten Ausgangsleitungen werden an drei analoge Eingänge des ATmega angeschlossen.

Beim praktischen Einsatz des Sensors auf dem Quadrocopter wurde die Messung durch Vibrationen gestört, die eine Schwingung mit einer Frequenz von ca. 1kHz im Ausgabesignal des Sensors erzeugten. Auf der am stärksten betroffenen Achse betrug die Amplitude des Störsignals bis zur vierfachen Erdbeschleunigung (4g). Um diese Schwingungen stärker zu dämpfen, wurde in den oben beschriebenen RC-Filtern der empfohlene  $1k\Omega$ -Widerstand durch einen  $10k\Omega$ -Widerstand ersetzt, wodurch die Grenzfrequenz des Filters auf ca. 160Hz gesenkt wird. Zusätzlich wurden digitale Filter in die Software auf dem Mikrocontroller eingebaut, die weiter unten genauer beschrieben werden.

Die Spannung der Messausgänge  $X_{OUT}$ ,  $Y_{OUT}$  und  $Z_{OUT}$  verhält sich ratiometrisch zur Versorgungsspannung. Das bedeutet, dass an den Messausgängen bei einer größeren oder kleineren Versorgungsspannung ebenfalls eine größere oder kleinere Spannung bei gleicher Beschleunigung anliegt (siehe [Fre08, Seite 4]). Ausschlaggebend für das Messen der Beschleunigung ist daher das Verhältnis zwischen der Spannung an den Messausgängen und der Versorgungsspannung. Um eine möglichst genaue und störungsarme Messung zu erreichen, wurde daher auf der Steuerplatine die Versorgungsspannung an  $V_{DD}$  als Referenzspannung für den A/D-

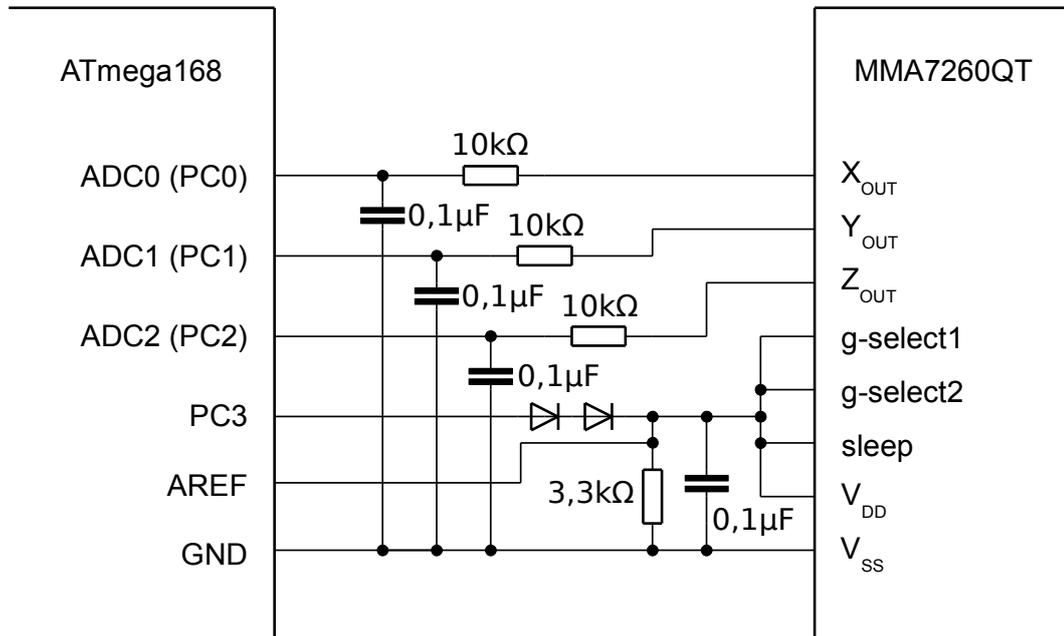


Abbildung 3.4.3.: Dieser Schaltplan zeigt die vollständige Beschaltung des Beschleunigungssensors, wie sie auf der Steuerplatine umgesetzt ist. Auf der ATmega-Seite dienen ADC0 bis ADC2 als Analogeingänge für die Signale auf den drei analogen Ausgängen  $X_{OUT}$  bis  $Z_{OUT}$  des Sensorchips. Die Widerstände und Dioden auf den verbindenden Leitungen bilden für jede Achse den in Unterabschnitt 3.4.1 beschriebenen Tiefpassfilter. Der generische I/O-Pin PC3 des ATmega wird zur Spannungsversorgung des Sensorchips eingesetzt. Unten rechts ist die bereits in Abbildung 3.4.2 abgebildete Schaltung zum Absenken und glätten der Versorgungsspannung zu erkennen. Die Ausgangsspannung dieses Aufbaus wird als Referenzspannung für den A/D-Wandler an den AREF-Pin des ATmega zurückgeführt.

Wandler verwendet. Dazu wurde  $V_{DD}$  mit dem Pin AREF am ATmega verbunden. Schwankungen in der Versorgungsspannung wirken sich im gleichen Verhältnis auf die Messausgänge aus, so dass bei dieser Verschaltung das Ergebnis der A/D-Wandlung nicht verfälscht wird, solange die Frequenz der Schwankungen deutlich unter der Grenzfrequenz des verwendeten Tiefpassfilters liegt.

### 3.4.2. Abfrage der Beschleunigungswerte

Zum Bestimmen der Beschleunigung auf jeder Achse muss nun die Spannung an den verwendeten Analogeingängen des ATmega gemessen werden. Eine Möglichkeit dazu ist, erst dann die Messungen durchzuführen, wenn die aktuelle Beschleunigung im Programm abge-

### 3. Entwicklung einer Steuerplatine

fragt wird. Alternativ dazu ist es auch möglich, im Hintergrund regelmäßig die aktuellen Werte zu erfassen, indem man interruptgesteuert neue Messungen anstößt. Bei der zweiten Variante können die Eingänge mit deutlich höherer Frequenz abgefragt und die Ergebnisse zwischen den Abfragen gefiltert werden, um genauere Messungen zu erhalten und Alias-Effekte zu vermeiden.

Die Messungen erfolgen in der Firmware der Steuerplatine im Hintergrund. Der A/D-Wandler wird dazu in einem Modus betrieben, bei dem nach Abschluss einer Messung sofort die nächste beginnt (Free Running Mode). Nach jeder Konvertierung wird ein Interrupt ausgelöst, der den neuen Messwert verarbeitet und den Eingangskanal des A/D-Wandlers reihum auf die nächste Achse des Beschleunigungssensors einstellt.

Um die nötige Rechenleistung gering zu halten und eine hohe Genauigkeit zu erzielen, wurde mit einem Prescaler von 128 die geringste einstellbare Geschwindigkeit für den A/D-Wandler im Free Running Mode gewählt. Bei dieser Einstellung entspricht ein A/D-Taktzyklus 128 CPU-Taktzyklen. Eine A/D-Wandlung im Free Running Mode benötigt 13 A/D-Takte. Damit ergibt sich die Samplefrequenz  $f_s$  bei der verwendeten CPU-Taktrate von 8 MHz wie folgt:

$$f_s = \frac{8000000 \frac{\text{cycles}}{s}}{128 \cdot 13 \frac{\text{cycles}}{\text{sample}}} \approx 4800 \frac{\text{samples}}{s}$$

Da die drei Achsen des Beschleunigungssensors reihum abgefragt werden, beträgt die Sample-rate auf jedem Kanal ein Drittel dieser Frequenz, und damit ca. 1,6 kHz.

Die gemessenen Werte werden noch in der Interruptroutine exponentiell geglättet, indem der neue Messwert in jedem Schritt gewichtet mit dem gespeicherten Wert addiert wird, um den neuen gespeicherten Wert zu erhalten. Diese Glättung entspricht in der Wirkung einem Tiefpassfilter erster Ordnung und ist daher geeignet, hochfrequente Störungen wie die in Unterabschnitt 3.4.1 angesprochenen Vibrationen weiter zu schwächen.

Wie auch bei der Abfrage der MotionPlus sollten die ermittelten Rohdaten nach der Messung mit Hilfe eines Offsets und eines Faktors mit Hilfe der Formel  $a = k \cdot (\text{rohwert} - \text{offset})$  in eine einheitliche Einheit überführt werden. In diesem Fall wird dazu die Erdbeschleunigung 1g verwendet.

Die Kalibrierung im Betrieb ist schwieriger als beim Gyrosensor. Wenn der Beschleunigungssensor exakt horizontal ausgerichtet ist, sollten an den beiden horizontalen Achsen 0g und an der vertikalen Achse 1g angezeigt werden, womit bei bekanntem  $k$  der Offset kalibriert werden könnte. Es kann allerdings nicht vorausgesetzt werden, dass sich der Sensor beim Start des Quadropters in einer perfekt horizontalen Lage befindet.

Eine einmalige manuelle Kalibrierung für sowohl  $\text{offset}$  als auch  $k$  ist jedoch einfach durchführbar. Dazu richtet man den Beschleunigungssensor mit jeder Achse einmal in Richtung des Erdmittelpunkts und einmal in die entgegengesetzte Richtung aus. Die Ausrichtung kann in diesem Bereich leicht variiert werden, um den maximalen bzw. minimalen Rohwert zu finden.

Kurze Beschleunigungsspitzen, die durch Wackeln entstehen, müssen dabei ignoriert oder mit einem Tiefpassfilter entfernt werden. Die so ermittelten Rohwerte  $min$  und  $max$  für jede Achse entsprechen  $1g$  und  $-1g$ . Der Mittelwert  $\frac{min+max}{2}$  entspricht also  $0g$  und damit dem gesuchten *offset* für die entsprechende Achse. Der Faktor  $k$  wird ermittelt als  $k = \frac{2g}{max-min}$ .

Wenn zwischen der Messachse und der Schwerkraftachse ein Winkel von  $\alpha$  liegt, beträgt die gemessene Beschleunigung  $a = 1g \cdot \cos(\alpha)$ . Da die Kosinusfunktion im Bereich um  $0^\circ$  nahezu konstant ist, sind die mit der beschriebenen Methode ermittelten Werte auch dann noch relativ genau, wenn die Achsen nicht präzise in Richtung der Schwerkraft ausgerichtet werden. Bei einer Abweichung von  $5^\circ$  liegt der gemessene Wert weniger als  $0,4\%$  unter dem tatsächlichen Wert.

Beim Interpretieren der Messwerte muss beachtet werden, dass die Ausgabe des Sensors nicht der tatsächlichen Beschleunigung des Moduls entspricht, sondern der scheinbaren Kraft, die im Bezugssystem des Moduls auf eine im Chip beweglich aufgehängte Masse wirkt. Wenn sich die Masse in einer zentralen Position befindet, liegt an allen Ausgängen eine neutrale Spannung an, die ungefähr der halben Versorgungsspannung entspricht. Wenn die Masse in Richtung einer Achse ausgelenkt wird, steigt die Spannung am Ausgabepin für diese Achse, bei einer Auslenkung entgegen der Richtung der Achse fällt die Spannung. Die Ausrichtung der Achsen im Fall des MMA7260 ist in Abbildung 3.4.1 gezeigt.

Daraus folgt, dass die Erdbeschleunigung „positiv“ gemessen wird, also in Richtung einer Achse eine positive Spannung erzeugt. Echte Beschleunigungen werden dagegen „negativ“ gemessen, da die aufgehängte Masse aufgrund ihrer Trägheit in die der Beschleunigung entgegengesetzte Richtung ausgelenkt wird. Im freien Fall entspricht die tatsächliche Beschleunigung genau der Erdbeschleunigung und der Sensor zeigt an allen Ausgängen eine neutrale Spannung. Wenn der Sensor nicht beschleunigt wird, zeigt der gemessene Vektor in Richtung des Erdmittelpunkts.

Wie schon im Fall des Gyrosensors entsprechen die im Datenblatt definierten Achsen des Beschleunigungssensors nicht dem Koordinatensystem des Quadropters, das in Abschnitt 2.1 festgelegt ist. In diesem Fall ist der Bezug zwischen den Koordinatensystemen jedoch komplizierter. In erster Näherung entspricht die z-Achse des Beschleunigungssensors der negativen z-Richtung im Quadroptersystem, während die x- und y-Achse um  $45^\circ$  diagonal zu den entsprechenden Achsen des Quadroptersystems stehen. Bedingt durch Ungenauigkeiten bei der Konstruktion der Platine ist der Beschleunigungssensor jedoch um mehrere Grad gegen die horizontale Ebene des Quadropters verdreht.

Um diese Schräglage auszugleichen, wurde aus Messungen, bei denen der Quadropter horizontal ausgerichtet wurde, eine Rotationsmatrix berechnet, die die Transformation aus dem Sensor-Koordinatensystem in das Quadropter-Koordinatensystem beschreibt. Diese Matrix wurde als Konstante in das Beschleunigungssensor-Softwaremodul eingebaut. Bei der Abfrage der Beschleunigung werden die ermittelten Werte im Sensor-Koordinatensystem mit dieser

### 3. Entwicklung einer Steuerplatine

Matrix multipliziert, um die Beschleunigungen im Quadrokopter-Koordinatensystem zu erhalten.

## 3.5. Fernsteuerung: robbe Futaba F-14

Um neben dem Erreichen eines Schwebeflugs auch die Umsetzung von Steuervorgaben zu testen, wurde eine Möglichkeit zur Steuerung des Quadrokopters mit einer Vierkanal-Funkfernsteuerung eingebaut. Bei der eingesetzten Fernsteuerung handelt es sich um das Modell *Futaba F-14* der Firma robbe, das auf [robb] beschrieben ist. Sie verfügt über zwei Steuerhebel, von denen jeder auf zwei Achsen beweglich ist.

Die Steuerung wurde so gewählt, dass der linke Steuerhebel in vertikaler Richtung den Gesamtschub bestimmt. In horizontaler Richtung legt dieser Hebel die Winkelgeschwindigkeit um die Gierachse fest. Mit dem rechten Steuerhebel kann die Neigung des Quadrokopters bestimmt werden. Wird dieser Hebel nach oben bewegt, neigt sich der Quadrokopter nach vorne und beschleunigt dadurch in diese Richtung. Eine Bewegung des Hebels nach rechts neigt auch den Quadrokopter nach rechts.

Die Achsen des rechten Steuerhebels werden allerdings nicht als Roll- und Nickwinkel interpretiert, sondern als Komponenten eines horizontalen Rotationsvektors. Die Richtung der Auslenkung des Steuerhebels entspricht damit der Richtung, in die der Quadrokopter geneigt wird, was wiederum der Richtung entspricht, in die der Quadrokopter horizontal beschleunigt.

Im Unterschied zu einer Berechnung der Ausrichtung aus Nick- und Rollwinkel, bei der üblicherweise zuerst die Rotation um die Nickachse und dann um die körperfeste Rollachse durchgeführt wird, werden die Achsen bei dieser Interpretation „gleichberechtigt“ behandelt. Dies erscheint bei einem Quadrokopter intuitiv sinnvoll, da kein konstruktionsbedingter Unterschied zwischen den beiden horizontalen Achsen existiert.

Die weitere Verarbeitung der Eingaben aus der Fernsteuerung erfolgt durch den Regler und ist daher in Abschnitt 3.8 beschrieben.

### 3.5.1. Anschluss und Auswertung des Empfängers

Als Empfangsmodul wird das *Futaba FP-R118F* eingesetzt, das bei robbe im Lieferumfang der Fernsteuerung enthalten ist. Dieser Empfänger verfügt über acht Kanäle, von denen nur vier für den Quadrokopter benötigt werden. Mit einem Gewicht von 35g (siehe [roba]) ist der Empfänger relativ schwer und erhöht daher signifikant den Stromverbrauch der Motoren, da für jedes Gramm zusätzliches Gewicht ca. 0,13 Watt zusätzlich aufgewendet werden müssen, um den Schub für einen Schwebeflug zu erzeugen (vgl. Unterabschnitt 3.6.3). Der Empfänger sollte daher nur als Testausrüstung und nicht als fester Teil der Steuerplatine betrachtet werden.

Eine weiterführende Arbeit könnte die Entwicklung einer leichteren und günstigeren Funksteuerung behandeln.

Der Empfänger verfügt für jeden Kanal über eine Buchse mit drei Kontakten. Zwei dieser Kontakte dienen der Stromversorgung und sind zwischen allen Buchsen verbunden. Die Versorgung des Empfängers erfolgt mit 5 Volt, die über einen I/O-Pin des ATmega zur Verfügung gestellt werden. Der dritte Kontakt dient als Steuerausgang für einen Servomotor. Eine Messung mit einem Oszilloskop zeigt, dass dieser Ausgang die meiste Zeit auf einem Low-Pegel liegt, der bei vorhandenem Funksignal in regelmäßigen Zeitabständen von ca. 21ms durch einen High-Puls unterbrochen wird, dessen Dauer den Stellwert für den jeweiligen Kanal angibt. Die Pulsdauer liegt in der Regel zwischen 1ms und 2ms, allerdings sind je nach Einstellung der Fernsteuerung auch Pulsweiten möglich, die sich knapp außerhalb dieses Bereichs befinden. Eine Pulsdauer von 1,5ms ist als Mittelstellung zu interpretieren. Dies entspricht laut [Wik12g] dem normalen Signal für Modellbauservos.

Die Ausgangsleitungen der unteren vier Kanäle des Empfängers sind direkt mit generischen I/O-Pins des ATmega verbunden. Zum Anschluss an die Steuerplatine wird der vorhandene Debug- und Programmieranschluss (siehe Abschnitt 3.2) verwendet, da dieser alle notwendigen Verbindungen zur Verfügung stellt. Dadurch ist es nicht möglich, die Fernsteuerung und die für die Ausgabe von Debugnachrichten eingesetzte serielle Schnittstelle des ATmega gleichzeitig zu verwenden. Da die Fernsteuerung als letztes angeschlossenes Gerät zur Steuerplatine hinzugefügt wurde, war die Software jedoch schon ausreichend weit entwickelt, dass keine Probleme mehr auftraten, für deren Untersuchung Debugnachrichten ausgegeben werden mussten.

Zur Auswertung der Pulse, die vom Empfänger erzeugt werden, wird das Pin Change Interrupt-Feature des ATmega verwendet. Dieses erlaubt es, einen Interrupt auszulösen, sobald an einem von mehreren I/O-Pins eine Flanke erkannt wird. Dabei kann über eine Bitmaske festgelegt werden, welche Pins einen Interrupt auslösen sollen. Die Interruptroutine hält bei der Erkennung einer steigenden Flanke für den entsprechenden Kanal einen Zeitstempel fest, um später bei der Erkennung einer fallenden Flanke die Differenz zum aktuellen Zeitstempel (und damit die Pulsdauer) berechnen zu können. Der Zeitstempel besitzt wie in Abschnitt 3.2 beschrieben eine Auflösung von 8 Mikrosekunden und kann damit die normalen Pulsweiten zwischen 1ms und 2ms in 125 Abstufungen unterscheiden. Der auf diese Weise bestimmte Zustand jeder Achse wird der restlichen Steuersoftware als Fließkommawert zur Verfügung gestellt, wobei der Pulslängenbereich zwischen 1ms und 2ms linear auf die Fließkommawerte zwischen -1 und 1 abgebildet wird.

Ein weiterer, gröber aufgelöster Zeitstempel wird dazu verwendet, einen Signalverlust der Fernsteuerung zu erkennen. Sobald für länger als 250ms kein Puls mit plausibler Länge empfangen wurde, wird für alle Kanäle die Mittelstellung an die Steuersoftware zurückgegeben, so dass ein einfacher stabilisierter Schwebeflug ausgeführt wird. Dadurch soll verhindert werden, dass sich der Quadrokopter bei einem Signalverlust unkontrolliert verhält, da Störsignale

### 3. Entwicklung einer Steuerplatine

zufällige Steuerpulse auslösen können, wenn kein Signal von der Fernsteuerung empfangen wird.

## 3.6. Antrieb

Der Antrieb ist dafür zuständig, die Ausgabe des Reglers in entsprechende mechanische Reaktionen umzusetzen. Er besteht daher nicht nur aus den Motoren und Rotoren, sondern auch aus den elektrischen Komponenten, welche diese ansteuern, sowie der Software, die die nötigen Steuersignale erzeugt.

Die Motoren und Rotoren bilden den offensichtlichsten Teil des Antriebs und werden in Unterabschnitt 3.6.1 behandelt. Dabei werden vor allem die vier Antriebsmodule des X-Ufos beschrieben und deren elektrischer Aufbau erklärt.

Für die Versorgung der Motoren wird eine deutlich höhere Leistung benötigt, als der ATmega direkt zur Verfügung stellen kann. Zum Schalten dieser Leistung sind daher externe Bauteile erforderlich, die gesteuert von den Ausgängen des ATmega den Strom für jeden einzelnen Motor ein- und ausschalten können. Diese Aufgabe wird auf der Steuerplatine von LeistungsmosFETs erfüllt. Diese Transistoren können große Ströme schalten und sind in leichten und kompakten Ausführungen erhältlich. In Unterabschnitt 3.6.2 werden MosFETs genauer beschrieben, und es werden Auswahlkriterien besprochen.

Um mit Hilfe der MosFETs die Leistung der Motoren zu steuern, wird Pulsweitenmodulation (PWM) eingesetzt. Bei dieser Technik wird die Stromversorgung der Motoren mit hoher Frequenz ein- und ausgeschaltet. Die Leistung wird gesteuert, indem das Tastverhältnis (das Verhältnis der Dauer von An- und Aus-Phase) variiert wird. Der Einsatz dieses Verfahrens ist naheliegend, da der Schub bereits auf der Originalplatine des X-Ufos auf diese Weise gesteuert wird. Die Pulsweitenmodulation wird in Unterabschnitt 3.6.3 behandelt.

Schließlich erläutert Unterabschnitt 3.6.4 die Softwarekomponente des Antriebs, die aus der Ausgabe des Reglers die PWM-Signale erzeugt.

### 3.6.1. Antriebsmodule des X-Ufos

Das X-Ufo verfügt über vier sehr ähnlich aufgebaute Antriebsmodule, die je aus einem Motor, einem Rotor und mehreren elektronischen Bauteilen bestehen. Motor und Rotor sind über Zahnräder mit jeweils acht (am Motor) und 69 Zähnen (am Rotor) verbunden, wodurch ein größeres Drehmoment am Rotor erreicht werden kann.

Zwei gegenüberliegende Rotoren werden im Uhrzeigersinn gedreht, die anderen beiden gegen den Uhrzeigersinn. Die unterschiedliche Drehrichtung wird durch eine unterschiedliche

Beschaltung der Motoren auf den Modulen erreicht, so dass alle Module von außen gleich angesteuert werden können. Damit in beiden Fällen ein Aufwärtsschub erzeugt wird, werden auf den Antriebsmodulen zwei unterschiedliche Ausführungen der Rotoren eingesetzt.

Die auf den Antriebsmodulen vorhandene Beschaltung der Bürstenmotoren ist in Abbildung 3.6.1 gezeigt. Die beiden parallel zum Motor geschalteten Kondensatoren wirken zusammen mit den beiden in Reihe geschalteten Spulen als Tiefpassfilter und sollen verhindern, dass vom Motor verursachte Spannungsspitzen zu Funkstörungen führen oder Bauteile beschädigen. Außerhalb dieses Aufbaus befindet sich eine Freilaufdiode, deren Funktion in der folgenden Beschreibung erläutert wird. Links in der Abbildung befindet sich der LeistungsmOSFET, der zum Schalten der Motorspannung verwendet wird.

Wenn der MOSFET durchgeschaltet wird, liegt die volle Versorgungsspannung des Quadropters am Antriebsmodul an. Durch die Spulen und Kondensatoren in der Schaltung stellt sich erst nach einer kurzen Einschwingphase ein elektrisch stabiler Zustand ein (siehe (b) in Abbildung 3.6.1). Die Freilaufdiode sperrt in diesem Zustand, da sie entgegen der Versorgungsspannung ausgerichtet ist. Die Spannung liegt nahezu in voller Höhe am Motor an, und es fließt Strom durch den Motor, der ein zur Stromstärke proportionales Drehmoment erzeugt.

Wenn dann durch Sperren des MOSFETs die Versorgungsspannung getrennt wird, kann der Stromfluss durch den Motor nicht sofort abbrechen, da die im Motor enthaltenen Spulen sowie die Entstörspulen einer plötzlichen Stromflussänderung entgegenwirken. Der Strom fließt daher nach der Abschaltung durch die Freilaufdiode weiter und entlädt die Kondensatoren (Zustand (c) in der Abbildung). Aufgrund der Schwellspannung der Diode ist dabei die Spannung auf der unteren Leitung in der Abbildung etwas höher als auf der oberen, die noch mit +12V verbunden ist. Ohne die Freilaufdiode würden die Spulen bei der Abschaltung eine sehr hohe Spannung aufbauen, die Bauteile auf der Steuerplatine beschädigen könnte.

Der Strom durch den Motor sinkt, da die Energie der Magnetfelder in den Spulen abgebaut wird, und nach kurzer Zeit stellt sich der elektrisch stabile Zustand (d) ein, in dem kein Strom mehr fließt. In diesem Zustand liegt auf der unteren Leitung eine Spannung zwischen 0V und 12V an. Diese entsteht dadurch, dass im Motor durch dessen Rotation eine Spannung induziert wird, deren Höhe zur Drehzahl des Motors proportional ist. Durch Messen dieser Spannung kann die aktuelle Drehzahl des Motors bestimmt werden. Diese Möglichkeit wurde jedoch auf der entwickelten Steuerplatine nicht genutzt, da hierzu weitere Bauteile erforderlich gewesen wären. Die induzierte Gegenspannung ist so gepolt, dass sie keinen Stromfluss durch die Freilaufdiode auslöst. Dieser Effekt träte nur dann auf, wenn der Motor entgegen der Antriebsrichtung gedreht würde, was aber beim normalen Betrieb des Quadropters nicht vorkommt.

Die induzierte Spannung wird auch dann im Motor erzeugt, wenn der MOSFET durchgeschaltet ist. Wenn der Motor sich in Antriebsrichtung dreht, ist diese Spannung der Versorgungsspannung entgegen gerichtet, so dass ein geringerer Strom durch den Motor fließt. Das bedeutet wiederum, dass der Motor bei gleichbleibender Versorgungsspannung ein geringeres Drehmoment erzeugt, je größer die Drehzahl ist. Wenn der Motor nicht belastet wird, stellt

### 3. Entwicklung einer Steuerplatine

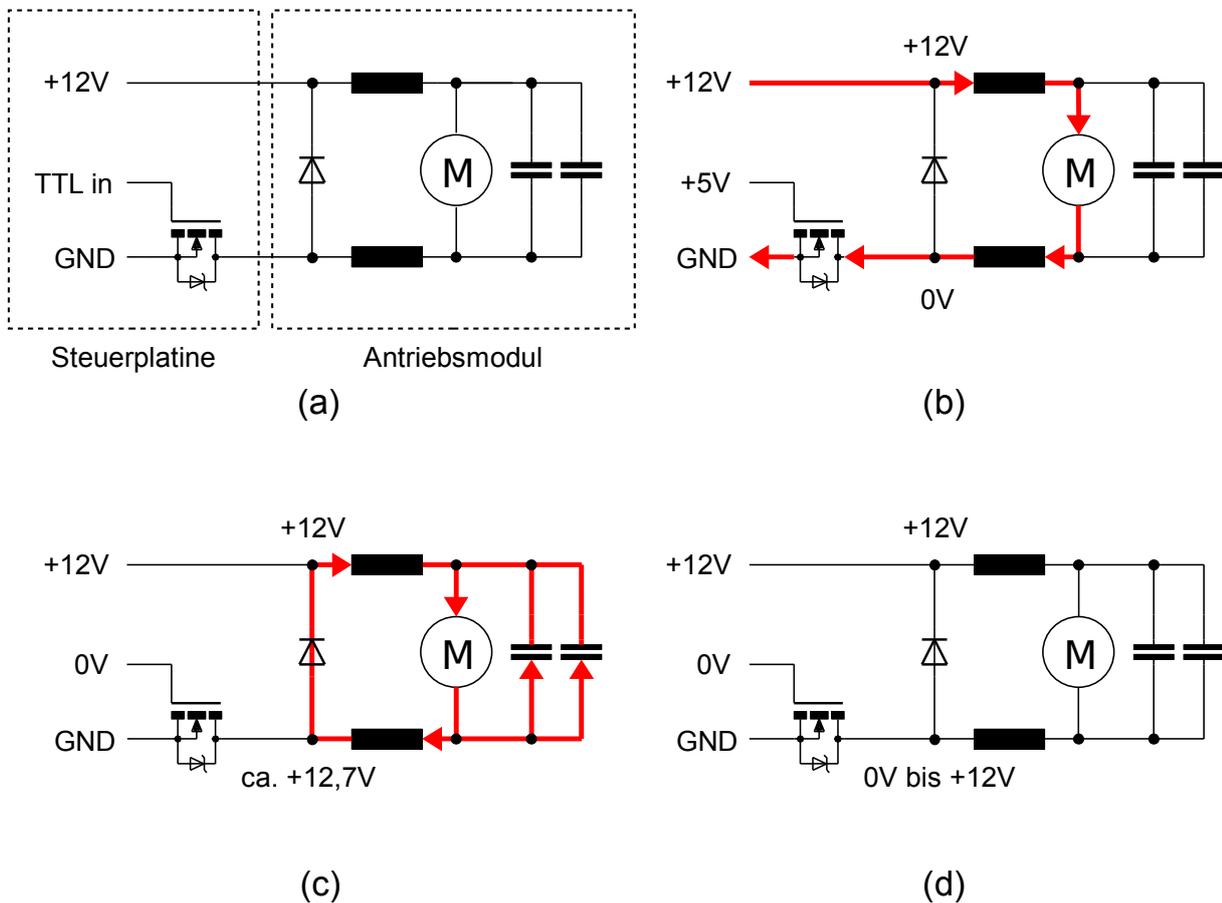


Abbildung 3.6.1.: (a) zeigt den elektrischen Aufbau eines Antriebsmoduls des X-Ufos sowie den Leistungstransistor auf der Steuerplatine. In (b) bis (d) ist der Stromfluss in verschiedenen Situationen durch Pfeile gekennzeichnet. Die Spannungen an der Freilaufdiode sind ebenfalls eingetragen. (b) zeigt den Zustand, der sich bei durchgeschaltetem Transistor einstellt, während (c) den Zustand direkt nach dem Sperren des Transistors darstellt. Der dort gezeigte Stromfluss endet allerdings nach kurzer Zeit, da er nur durch die gespeicherte Energie der Spulen und Kondensatoren aufrecht erhalten wird, und es stellt sich Zustand (d) ein, in dem kein Strom mehr fließt.

sich dadurch bei einer von Motor und Versorgungsspannung abhängigen Drehzahl ein Gleichgewichtszustand ein, bei dem (idealisiert betrachtet) kein Strom durch den Motor fließt.

### 3.6.2. Leistungs-MOSFETs

Zum Schalten hoher Ströme mit kleinen Spannungen werden heute oft Leistungs-MOSFETs eingesetzt. MOSFET steht für *Metal-Oxide-Semiconductor-Field-Effect-Transistor* (Metall-Oxid-Halbleiter-Feldeffekttransistor). Diese Transistoren besitzen meistens drei Anschlüsse: *Source*, *Drain* und *Gate*. Über die Spannung zwischen Gate und Source ( $V_{GS}$ ) kann der Widerstand zwischen Source und Drain gesteuert werden. Das Gate ist elektrisch von den restlichen Anschlüssen isoliert, zum Steuern eines MOSFETs ist daher anders als bei einem Bipolartransistor kein dauerhafter Strom erforderlich ([BGK<sup>+</sup>98, Seite 446]).

Neben der Isolation des Gates ist ein weiterer Vorteil von MOSFETs gegenüber Bipolar-Leistungstransistoren laut [Bar02] und [Loc88], dass MOSFETs sehr hohe Schaltgeschwindigkeiten erreichen können, so dass insbesondere bei Hochfrequenzanwendungen die Schaltverluste gering bleiben. Durch die Anwendung des PWM-Verfahrens zur Leistungssteuerung der Motoren ist dieser Vorteil ebenfalls für den Quadrokopter relevant.

Ein MOSFET ist für eine bestimmte Stromrichtung zwischen Source und Drain ausgelegt. Die Bezeichnungen Source und Drain (Quelle und Senke) beziehen sich dabei auf die Flussrichtung der Ladungsträger auf dem Halbleitermaterial, das Source und Drain verbindet. Bei n-Kanal-MOSFETs handelt es sich um negative Ladungsträger (Elektronen), bei p-Kanal-MOSFETs sind die Ladungsträger dagegen positiv (Defektelektronen). Folglich sind n-Kanal-MOSFETs für eine technische Stromrichtung von Drain zu Source ausgelegt, p-Kanal-MOSFETs für die entgegengesetzte Stromrichtung (vgl. [irp]). Wenn ein MOSFET in die entgegengesetzte Richtung betrieben wird, lässt er unabhängig von der Gate-Source-Spannung den Strom durch, da MOSFETs aufgrund ihres Aufbaus meistens eine parasitäre Diode enthalten, die Strom entgegen der Auslegungsrichtung leitet. Der Grund für die Existenz dieser Diode ist in [Loc88, Seite 3] beschrieben.

MOSFETs werden weiter zwischen selbstleitenden und selbstsperrenden Typen unterschieden. Ein selbstleitender MOSFET ist bei einer Spannung  $V_{GS} = 0$  Volt durchgeschaltet, ein selbstsperrender MOSFET gesperrt ([Sch]). Bei selbstsperrenden n-Kanal-MOSFETs und selbstleitenden p-Kanal-MOSFETs muss an Gate ein höheres Potential als an Source angelegt werden, um den Transistor durchzuschalten bzw. zu sperren. Bei den anderen beiden Kombinationen muss das Potential an Gate dazu niedriger sein als das an Source.

Auf der Steuerplatine sollen die MOSFETs die Spannung für die Motoren schalten. Dazu sind selbstsperrende n-Kanal-MOSFETs am passendsten geeignet. Wenn die MOSFETs als Low-Side-Schalter verwendet werden (Source an GND, Drain an der GND-Zuleitung der Antriebsmodule), können die Gate-Leitungen direkt über die Ausgänge des Mikrocontrollers gesteuert werden, die auf GND und +5V geschaltet werden können. Liegt der Ausgang auf GND, ist

### 3. Entwicklung einer Steuerplatine

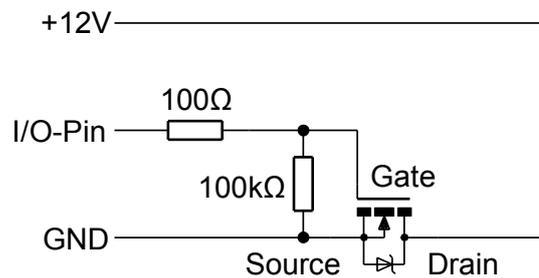


Abbildung 3.6.2.: Beispiel zur Verwendung eines n-Kanal-MOSFETs als Low-Side-Schalter, der eine an die Ausgangsleitungen (rechts) angeschlossene Last schalten kann.

die Spannung zwischen Gate und Source 0 Volt und der MOSFET sperrt. Liegt der Ausgang auf +5V, ist das Potential an Gate höher als an Source, und der Transistor schaltet wie oben beschrieben durch. Bei den anderen MOSFET-Typen wären zusätzliche Bauteile nötig, um das Gate mit der richtigen Spannung anzusteuern.

Eine mögliche Beschaltung der MOSFETs ist in Abbildung 3.6.2 gezeigt. Der 100kΩ-Pulldown-Widerstand sorgt dafür, dass das Gate auf einen Low-Pegel gezogen wird, wenn der I/O-Pin hochohmig geschaltet oder nicht verbunden ist. Dieser Zustand tritt beim Einschalten des Quadropters auf, da die I/O-Pins beim Reset des Mikrocontrollers zunächst automatisch als hochohmige Eingänge konfiguriert werden. Durch den Widerstand sperrt der MOSFET in dieser Situation zuverlässig. Ohne diesen Widerstand könnte der MOSFET ganz oder teilweise durchschalten, was zu einem unerwünschten Einschalten der Last und, falls der MOSFET nur teilweise durchschaltet, zu einem Überhitzen des MOSFETs führen könnte.

Der 100Ω-Widerstand zwischen I/O-Pin und Gate dient dem Schutz des I/O-Pins, da ohne diesen Widerstand beim Umschalten des Transistors kurzzeitig ein hoher Strom zum Umladen des Gates fließt, der den I/O-Baustein beschädigen könnte. Bei einer Spannung von 5V am I/O-Pin kann mit dem eingezeichneten Widerstand maximal ein Strom von 50mA auftreten. Um möglichst geringe Umschaltzeiten zu erreichen und Bauteile zu sparen, wurde dieser Widerstand auf der Steuerplatine jedoch nicht eingebaut. Der dadurch kurzzeitig entstehende hohe Stromfluss hat in der Praxis keine Probleme verursacht.

Bei der Auswahl eines geeigneten MOSFETs sind neben dem Typ (n- oder p-Kanal, selbstsperrend oder selbstleitend) auch andere Eigenschaften zu beachten, die meistens in den Datenblättern angegeben sind. Zunächst muss das Bauteil in der Lage sein, die maximal auftretende Spannung zu sperren (Drain-to-Source Voltage,  $V_{DS}$ ). Durch die auf den Motormodulen vorhandenen Freilaufdioden wird die Spannung an den Transistoren auf der Steuerplatine auf ca. 12,7 Volt begrenzt.

Im durchgeschalteten Zustand muss das Bauteil für die benötigte Stromstärke ausgelegt sein (Continuous Drain Current,  $I_D$ ). Jeder Motor des X-Ufos kann kurzzeitig Ströme in der Größenordnung von 5 Ampere ziehen, im Normalbetrieb fließt immer noch ca. 1 Ampere durch jeden Motor.

Eine weitere wichtige Angabe ist die Schwellspannung zwischen Gate und Source  $V_{GS(th)}$ , ab der der Transistor durchschaltet bzw. sperrt. Diese Spannung sollte für die Verwendung auf der Steuerplatine im Bereich unter 5V liegen, um eine direkte Ansteuerung durch den ATmega zu ermöglichen. Eine kleinere Schwellspannung ist hier vorteilhaft, da Verluste im Mikrocontroller oder in Leitungen zu einer geringeren Spannung zwischen Gate und Source führen können. Liegt die Schwellspannung nur knapp unter 5 Volt, kann dadurch der Widerstand im MOSFET ansteigen, was zu größeren Verlusten und Wärmeentwicklung führt.

Für einen effizienten Betrieb ist auch ein möglichst geringer Widerstand zwischen Source und Drain im durchgeschalteten Zustand ( $R_{DS(on)}$ ) erforderlich. Da dieser Wert von  $V_{GS}$  abhängt, sollte darauf geachtet werden, für welche Spannung  $V_{GS}$  diese Angabe im Datenblatt gemacht wird.

Ein weiteres Auswahlkriterium ist die Kapazität zwischen den verschiedenen Anschlüssen des Transistors, insbesondere zwischen Gate und Source sowie zwischen Gate und Drain, da diese Kapazitäten zum Schalten des Transistors umgeladen werden müssen. Je geringer die Kapazitäten sind, desto schneller kann der Transistor daher geschaltet werden, was wiederum geringere Verluste zur Folge hat. Viele Datenblätter geben auch die mit diesen Kapazitäten zusammenhängende Gesamtladung des Gates  $Q_G$  an, die zum Schalten des Transistors geladen werden muss. In [Bar02, Seite 10] ist beschrieben, dass diese Angabe einen besseren Vergleich der Schaltzeiten ermöglicht als die Kapazitätsangaben, insbesondere wenn Bauteile verschiedener Hersteller verglichen werden.

Für den Aufbau der Steuerplatine wurde der *IRLU3717* ausgewählt. Dieser Transistor kann laut Datenblatt [Int05] einen Strom bis 120 Ampere sowie eine Spannung von bis zu 20 Volt schalten. Der Widerstand  $R_{DS(on)}$  beträgt ca.  $5m\Omega$  bei einer Gatespannung  $V_{GS}$  von 4,5 Volt. Daraus ergibt sich im eingeschalteten Zustand für den normalen Betrieb im Quadrocopter eine Verlustleistung von ca.  $(1A)^2 \cdot 5m\Omega = 5mW$ . Die Verluste beim Umschalten sind ebenfalls gering, da die Zeit zum Umladen des Gates auf der Steuerplatine weniger als 500 Nanosekunden beträgt<sup>5</sup>. Selbst wenn während dieser Zeit die volle Leistung am Transistor abfließe, entspräche das nur einer Energie von  $12V \cdot 1A \cdot 500ns = 6\muWs$ . Bei einer PWM-Frequenz von 4kHz, wie sie auf der Steuerplatine verwendet wird, finden 8000 Umschaltvorgänge pro Sekunde statt, wodurch nach der obigen Rechnung maximal eine zusätzliche Verlustleistung von  $6\muWs \cdot \frac{8000}{s} = 48mW$  entsteht. Die gesamte Verlustleistung von maximal 53mW pro MOSFET ist so gering, dass die Bauteile keine zusätzliche Kühlung benötigen und sich trotz ihrer geringen Größe nicht merklich erhitzen.

<sup>5</sup>Nach dieser Zeit ist die Spannung bis auf 0,5 Volt gefallen bzw. bis auf 4,5 Volt gestiegen. Das Ergebnis wurde experimentell mit einem Oszilloskop ermittelt.

#### 3.6.3. Pulsweitenmodulation

Um die Bewegung des Quadropters zu steuern, muss es möglich sein, die Leistung der Motoren zu variieren. Eine naheliegende Idee besteht darin, die MOSFETs nur teilweise durchzuschalten, indem die Spannung an den Gate-Eingängen der MOSFETs mit Hilfe eines D/A-Wandlers eingestellt wird. Dadurch kann der Widerstand der MOSFETs beeinflusst werden, um so die Leistung der Motoren festzulegen. Diese Methode ist jedoch in der Praxis nicht brauchbar. Um die Spannung an einem Motor auf die Hälfte zu senken, müsste die andere Hälfte am MOSFET abfallen, so dass am MOSFET die gleiche Leistung in Hitze umgewandelt würde, die vom Motor aufgenommen wird. Dadurch würde der MOSFET schnell überhitzen. Selbst wenn für ausreichende Kühlung gesorgt werden kann, ist diese Methode insbesondere für ein batteriebetriebenes Modellfluggerät ungeeignet, da ein großer Teil der Energie als Wärme verloren geht und nicht für den Antrieb genutzt wird.

Die MOSFETs sollten daher so angesteuert werden, dass an ihnen möglichst wenig Leistung abfällt. Im voll durchgeschalteten Zustand ist der Widerstand der MOSFETs im Verhältnis zum Widerstand der Motoren sehr gering, es fällt daher auch nur ein kleiner Teil der Leistung an den MOSFETs ab. Im gesperrten Zustand ist der Widerstand dagegen so hoch, dass praktisch kein Strom mehr fließt und damit ebenfalls sehr wenig Leistung abfällt. Die MOSFETs sollten also entweder vollständig ein- oder ausgeschaltet sein, und der Übergang zwischen den beiden Zuständen sollte möglichst schnell erfolgen.

Um dennoch die Motorleistung zu beeinflussen, kann die Technik der Pulsweitenmodulation (PWM) benutzt werden. Dabei wird ein Rechtecksignal mit fester Frequenz erzeugt, welches die MOSFETs in schneller Folge ein- und ausschaltet. Das Verhältnis zwischen der Dauer der eingeschalteten Phase und der ausgeschalteten Phase wird dabei als Tastverhältnis bezeichnet. Durch Ändern des Tastverhältnisses (und damit der Pulsweite) werden die Motoren in jeder PWM-Periode für eine unterschiedlich lange Zeit mit Spannung versorgt, so dass die Leistung gesteuert werden kann.

Die PWM-Frequenz ist dabei von großer Bedeutung. Bei einer sehr niedrigen Frequenz würden die Motoren abwechselnd die volle Drehzahl erreichen und dann wieder stehenbleiben. Da die mechanische Trägheit der Motoren und Rotoren als Tiefpassfilter für die Drehgeschwindigkeit wirkt, wird diese bei höheren PWM-Frequenzen geglättet, so dass eine gleichmäßige Geschwindigkeit entsteht, die durch das Tastverhältnis variiert werden kann. Eine höhere PWM-Frequenz hat allerdings auch Auswirkungen auf die elektrischen Komponenten der Antriebsmodule, da die im Motor vorhandenen Spulen schnellen Änderungen im Stromfluss entgegenwirken. Bei einer ausreichend hohen PWM-Frequenz ist der Stromfluss durch den Motor daher nahezu konstant. In der Aus-Phase des PWM-Signals fließt der Strom dann wie in Unterabschnitt 3.6.1 beschrieben durch die Freilaufdiode weiter.

Ein Unterschied zwischen diesem kontinuierlichen Stromfluss bei hohen PWM-Frequenzen und dem gepulsten Stromfluss bei niedrigeren Frequenzen ist, dass bei kontinuierlichem

Stromfluss mit gleicher abgegebener mechanischer Leistung und gleicher Drehzahl geringere Kupferverluste<sup>6</sup> im Motor auftreten als bei gepulstem Stromfluss. Das ist dadurch zu erklären, dass die mechanische Leistung des Motors linear mit dem Stromfluss durch den Motor steigt, die (ohmschen) Kupferverluste dagegen quadratisch mit dem Stromfluss steigen.

Um diesen Effekt genauer zu beleuchten, soll die vom Motor auf den Rotor abgegebene mechanische Leistung  $P_m$  mit den Kupferverlusten  $P_{cu}$  verglichen werden. Nach [Dem94, Seite 129] ist die Rotationsenergie eines Körpers  $E_{rot}$  um eine Achse mit Trägheitsmoment  $J$  bei einer Winkelgeschwindigkeit  $\omega$ :

$$E_{rot} = \frac{1}{2} J \omega^2$$

Wenn man ein gleichbleibendes Trägheitsmoment voraussetzt, kann man daraus die mechanische Leistung bestimmen, die zur Änderung der Winkelgeschwindigkeit notwendig ist:

$$P_m = \dot{E}_{rot} = J\omega \cdot \dot{\omega} = M\omega$$

Dabei ist  $M$  das auf den Rotor wirkende Drehmoment. Dieses ist bei permanenterregten Elektromotoren wie denen im X-Ufo proportional zur Stärke des Stromflusses  $I$  durch den Motor. Dies ergibt sich aus der Formel für die Lorentz-Kraft  $\vec{F}_L = l\vec{I} \times \vec{B}$  (nach [BGK<sup>+</sup>98, Seite 227]), in der  $\vec{I}$  Richtung und Stärke des Stromflusses durch einen Leiter der Länge  $l$  ist, und  $\vec{B}$  das Magnetfeld, das diesen Leiter durchsetzt. Bei einem permanenterregten Elektromotor ist das Erregerfeld von der Stromstärke durch den Motor unabhängig, so dass sich die Lorentz-Kraft (und damit das Drehmoment) proportional zur Stromstärke verhält.

$$P_m = M \cdot \omega = I \cdot k \cdot \omega$$

Die Konstante  $k$  ist dabei vom verwendeten Motor abhängig. Bei einer zeitlich variablen Stromstärke  $I$  kann die mittlere Leistung  $\bar{P}_m$  über einen kurzen Zeitraum wie folgt ermittelt werden, solange die Winkelgeschwindigkeit  $\omega$  sich während dieses Zeitraums nur unwesentlich ändert:

$$\bar{P}_m = \bar{M} \cdot \omega = \bar{I} \cdot k \cdot \omega$$

Dieser mechanischen Leistung werden die Kupferverluste im Motor gegenübergestellt. Diese können aus dem Kupferwiderstand sowie der Stromstärke des durch den Motor fließenden Stroms berechnet werden:

$$P_{cu} = I^2 \cdot R$$

Die mittlere Verlustleistung bei variabler Stromstärke ist hier:

$$\bar{P}_{cu} = \bar{I}^2 \cdot R$$

---

<sup>6</sup>Kupferverluste bezeichnen die Leistung, die durch den elektrischen Widerstand der Spulen im Motor als Wärme verloren geht.

### 3. Entwicklung einer Steuerplatine

Daraus ergibt sich für das Verhältnis  $rel$  aus mittlerer Verlustleistung und mittlerer mechanischer Leistung:

$$rel = \frac{\overline{P}_{cu}}{\overline{P}_m} = \frac{\overline{I}^2}{\overline{I}} \cdot \frac{R}{k \cdot \omega}$$

Bei kontinuierlichem Stromfluss, der theoretisch bei hohen PWM-Frequenzen angenähert werden kann, gilt  $\overline{I} = I$ . Dies ermöglicht folgende Umformung:

$$rel_{cont} = \frac{\overline{P}_{cu,cont}}{\overline{P}_{m,cont}} = \frac{I^2}{I} \cdot \frac{R}{k \cdot \omega} = I \cdot \frac{R}{k \cdot \omega} = \overline{I} \cdot \frac{R}{k \cdot \omega}$$

In dem Fall, dass die PWM-Frequenz so gering ist, dass der Stromfluss durch den Motor als gepulst betrachtet werden kann, aber noch so schnell, dass sich die Winkelgeschwindigkeit über die Periode eines Pulses nicht wesentlich ändert<sup>7</sup>, fließt während der Aus-Phase kein Strom durch den Motor (vgl. Unterabschnitt 3.6.1). Damit wird in dieser Phase keine mechanische Leistung erbracht, und es treten auch keine Kupferverluste auf. Während der An-Phase fließt dagegen ein Strom der Stärke  $I_{on}$ . Die mittlere Stromstärke ist unter diesen Voraussetzungen  $\overline{I} = I_{on} \cdot tv$ , die mittlere quadratische Stromstärke ist  $\overline{I}^2 = I_{on}^2 \cdot tv$ , wobei  $tv$  das PWM-Tastverhältnis ist. Das Verhältnis der Verlustleistung zur mechanischen Leistung kann daher im gepulsten Fall wie folgt umgeformt werden:

$$rel_{puls} = \frac{\overline{P}_{cu,puls}}{\overline{P}_{m,puls}} = \frac{I_{on}^2 \cdot tv}{I_{on} \cdot tv} \cdot \frac{R}{k \cdot \omega} = \frac{\overline{I} \cdot I_{on}}{I_{on} \cdot tv} \cdot \frac{R}{k \cdot \omega} = \frac{\overline{I}}{tv} \cdot \frac{R}{k \cdot \omega} = \frac{rel_{cont}}{tv}$$

Da  $tv$  ein Wert zwischen 0 und 1 ist, gilt:

$$rel_{puls} = \frac{rel_{cont}}{tv} \geq rel_{cont}$$

Die relativen Kupferverluste sind bei gepulsten Motorstrom (und einem Tastverhältnis  $tv < 1$ ) also höher als bei einem kontinuierlichen Motorstrom. Es ist zu beachten, dass  $tv$  in dieser Formel für das Tastverhältnis bei langsamer PWM-Frequenz steht. Um bei gleichen Rahmenbedingungen die gleiche Leistung mit der hohen PWM-Frequenz einzustellen, ist im Allgemeinen ein anderes Tastverhältnis erforderlich, wie die Messungen am Ende dieses Abschnitts zeigen.

Die obenstehende Betrachtung suggeriert, dass eine höhere PWM-Frequenz in jedem Fall geringere Verluste verursacht. Diese Interpretation vernachlässigt jedoch einige wichtige Faktoren. Zum einen existieren neben den Kupferverlusten noch weitere Arten von Verlusten, die weiter unten behandelt werden. Zum anderen ist nicht garantiert, dass der Stromfluss durch den Motor mit steigender PWM-Frequenz monoton glatter wird, da das Zusammenspiel von

<sup>7</sup>Hier wird vorausgesetzt, dass der Rotor ausreichend träge ist, dass eine Frequenz existiert, für die eine solche Näherung angebracht ist. In den Versuchen am X-Ufo konnte dies bei PWM-Frequenzen von wenigen hundert Hertz beobachtet werden.

Kapazitäten und Induktivitäten auf den Motormodulen zu Resonanzeffekten führen kann (vgl. [BGK<sup>+</sup>98, ab Seite 274]). Wird die PWM-Frequenz jedoch ausreichend hoch gewählt, sollte sich aufgrund der Trägheit der elektrischen Elemente dennoch ein gleichmäßiger Stromfluss durch den Motor einstellen.

Eine höhere Frequenz bedeutet aber auch ein häufigeres Umschalten der MOSFETs. Diese Umschaltung kann nicht beliebig schnell erfolgen. Ein MOSFET ist daher bei jedem Umschaltvorgang für eine kurze Zeit nur teilweise durchgeschaltet, und in dieser Zeit fällt eine hohe Leistung am MOSFET ab. Bei jedem Zyklus aus Ein- und Ausschalten geht also eine (bei gleicher Last) konstante Energiemenge verloren<sup>8</sup>. Die Umschaltverluste steigen daher proportional mit der PWM-Frequenz.

Ein weiterer Faktor ist, dass die Hardware, die die Leistungstransistoren ansteuert, in der Lage sein muss, PWM-Signale mit der gewünschten Frequenz und einer sinnvollen Auflösung des Tastverhältnisses zu erzeugen. Aus den Angaben im ATmega-Datenblatt [Atm09] ist ersichtlich, dass mit dem auf der Steuerplatine eingesetzten ATmega168 bei einer Auflösung von 256 Abstufungen und der CPU-Frequenz von 8MHz eine maximale PWM-Frequenz von ca. 32kHz möglich ist.

Kabel, die mit hohen Frequenzen wechselnde Ströme führen, strahlen elektromagnetische Wellen ab, welche Radioempfänger und andere Geräte stören können. Da jede schnelle Umschaltung, wie sie bei PWM erfolgt, einen eigenen elektromagnetischen Puls erzeugt, steigt die abgestrahlte Energie mit steigender PWM-Frequenz an.

Abgesehen von diesen möglichen Problemen erzeugen PWM-getriebene Motoren oft Piepsgeräusche in Höhe der PWM-Frequenz. Die Lautstärke dieser Geräusche bei verschiedenen PWM-Frequenzen ist von den Motoren abhängig.

Es existieren also verschiedene frequenzabhängige Effekte, die die Effizienz und das Verhalten der Motormodule auf teilweise nur schwierig vorhersagbare Weise beeinflussen. Die PWM-Frequenz für die Steuerplatine wurde daher anhand von Messreihen von Schub und Effizienz bei verschiedenen Frequenzen ausgewählt. Dabei wurde vorwiegend auf eine hohe Effizienz geachtet.

Die Messreihen geben auch darüber Auskunft, welcher Schub sich bei unterschiedlichen Tastverhältnissen einstellt, was ebenfalls von der PWM-Frequenz abhängt. Diese Information wird in der Steuersoftware verwendet, um die richtige Pulsweite für einen vom Regler vorgegebenen Schub zu berechnen.

In jeder Messreihe wurde eine feste PWM-Frequenz verwendet, das Tastverhältnis wurde in 40 Schritten von 0 bis 1 variiert. Die gewählten PWM-Frequenzen entsprechen den höchsten drei Frequenzen, die bei einer Taktfrequenz des ATmega von 8MHz auf beiden verwendeten

<sup>8</sup>Das gilt natürlich nur, solange die Frequenz nicht so groß ist, dass der nächste Umschaltvorgang bereits beginnt, bevor der letzte abgeschlossen ist.

### 3. Entwicklung einer Steuerplatine

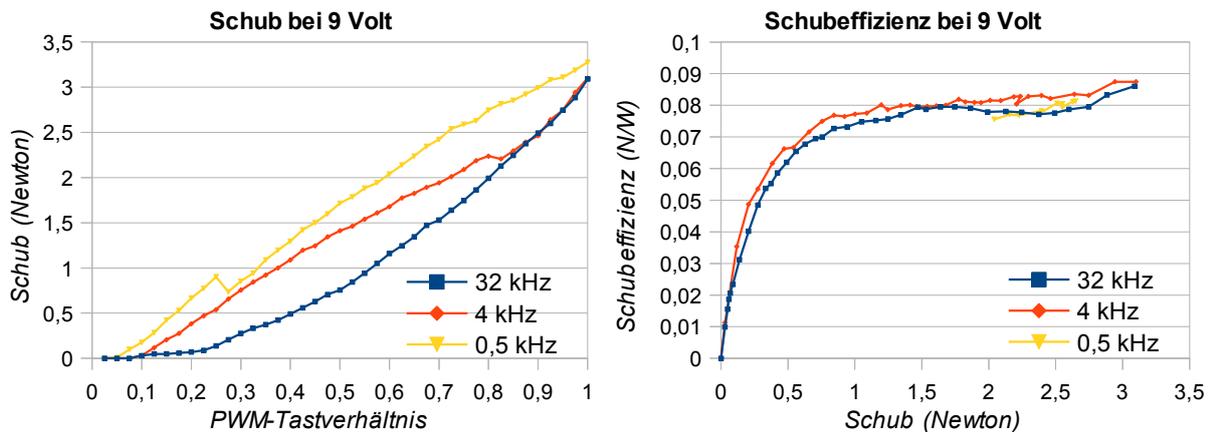


Abbildung 3.6.3.: Messreihen zu Schub und Schubeffizienz bei verschiedenen PWM-Frequenzen und Tastverhältnissen

Timern eingestellt werden können. Alle Motoren wurden dabei mit den gleichen Werten betrieben, so dass ein gleichmäßiger Aufwärtsschub entstand. Zum Erfassen des Schubs wurde der Quadrocopter mit einem Zusatzgewicht auf einer Waage befestigt. Um zu verhindern, dass die Messung durch den in Abschnitt 2.2.2 angesprochenen Bodeneffekt beeinflusst wird, welcher den Schub eines Rotors in Bodennähe verstärkt, wurde der Quadrocopter mit einem Abstandhalter auf der Waage befestigt. Die Waage wurde zu Beginn jeder Messreihe bei stehenden Rotoren tariert, so dass der Schub als negatives Gewicht abgelesen werden konnte.

Der Quadrocopter wurde über ein geregeltes Netzteil versorgt, das eine konstante Spannung liefern kann. Mit einem Multimeter wurde der Stromfluss durch den gesamten Quadrocopter bestimmt. Die ermittelten Werte für den Schub wurden für jede Kombination aus PWM-Frequenz und Pulsweite aufgeschrieben und sind in Abbildung 3.6.3 grafisch dargestellt. Gleichzeitig wurde die Stromstärke notiert, um die Schubeffizienz<sup>9</sup> zu bestimmen. Diese ist in einem zweiten Graphen gegen den Schub aufgetragen.

Da die Motoren bei einem Tastverhältnis von 1 dauerhaft mit Strom versorgt werden, sollte der Schub in diesem Fall unabhängig von der verwendeten PWM-Frequenz sein. Die sichtbare Abweichung verdeutlicht daher die begrenzte Genauigkeit der Messung. Es ist dennoch klar zu erkennen, dass das Schubverhalten bei verschiedenen PWM-Frequenzen deutlich voneinander abweicht. Die Messung der Schubeffizienz bei 500Hz ist leider unvollständig, da die Stromstärke nicht für alle Tastverhältnisse erfasst wurde. Da die erfassten Werte jedoch keine bessere Effizienz als bei 4kHz erwarten lassen, und wie unten beschrieben mehrmals Probleme bei der Messung mit 500Hz auftraten, wurde auf eine Wiederholung der Messreihe verzichtet.

Rechnet man aus der gesamten durchschnittlichen Stromstärke zurück auf die Stromstärke, die während der Ein-Phase durch jeden Motor geflossen sein muss, ergeben sich für die PWM-

<sup>9</sup>Mit Schubeffizienz ist hier der Quotient aus Schub und elektrischer Leistung gemeint.

Frequenz 500Hz bei niedrigen Tastverhältnissen über 2 Ampere pro Motor, während der höchste gemessene Wert bei den höheren Frequenzen 1,1 Ampere beträgt. Dies ist dadurch zu erklären, dass der Stromfluss durch den Motor aufgrund dessen Induktivität träge reagiert, so dass der Stromfluss bei hohen Frequenzen während der Ein-Phase nur geringfügig ansteigt. Bei niedrigen Frequenzen ist die Ein-Phase jedoch deutlich länger, so dass auch bei kleineren Tastverhältnissen ein hoher Stromfluss entstehen kann. Da ein hoher Stromfluss einem großen Drehmoment entspricht, erklärt dies auch den vergleichsweise hohen Schub bei kleinen Tastverhältnissen in der 500Hz-Messreihe.

Die gepulsten hohen Ströme, die bei dieser Frequenz auftreten, erschienen jedoch ungünstig, da sie Schwankungen in der Versorgungsspannung hervorrufen können. Da bei den Versuchen bereits Abstürze des Mikrocontrollers beobachtet wurden, die auf Spannungseinbrüche zurückgeführt werden konnten, wurde eine höhere PWM-Frequenz ausgewählt. Bei den beiden verbleibenden Frequenzen war die Schubeffizienz in der 4kHz-Messreihe konsistent höher als in der 32kHz-Messreihe. Daher wurde 4kHz als PWM-Frequenz für die Steuerplatine festgelegt.

#### 3.6.4. Antriebssoftware

Der Softwareteil des Antriebs ist dafür verantwortlich, die PWM-Signale für die Motoren zu erzeugen. Als Eingabe dient dabei die Vorgabe des Ausrichtungsreglers und eine Angabe des gewünschten Gesamtschubs. Der Regler betrachtet nicht die einzelnen Motoren, sondern gibt die gewünschte Leistung vor, die für eine Nick-, Roll- und Gierbewegung aufgewendet werden soll. Die Antriebssoftware berechnet aus diesen Angaben den Sollschub für die einzelnen Motoren und bestimmt das PWM-Tastverhältnis, das zum Erzielen dieses Schubs erforderlich ist. Dadurch werden die Details der Rotorkonfiguration und der Motoren vor dem Rest der Software verborgen. Daher kann z.B. die gleiche Regelsoftware auch für einen Hexakopter mit bürstenlosen Motoren eingesetzt werden, solange die Antriebssoftware entsprechend angepasst wird.

Der Zusammenhang zwischen dem Schub der Rotoren und den Winkelbeschleunigungen kann mit Hilfe der Überlegungen aus Abschnitt 2.2 bestimmt werden. Es werden jedoch nicht alle Effekte beachtet, die nach Gleichung 2.2.20 die Winkelgeschwindigkeit ändern können, da viele davon in den häufigsten Flugsituationen nahe des Schwebeflugs nur einen vergleichsweise geringen Einfluss auf die Bewegung des Quadropters ausüben oder von Größen abhängen, die nicht von Sensoren auf der Steuerplatine gemessen werden.

So werden Rollmoment und Horizontalkraft vernachlässigt, da die Geschwindigkeit des Quadropters relativ zur umgebenden Luft nicht gemessen wird. Weiterhin werden auch die gyroskopischen Effekte des Rahmens nicht beachtet, da der Quadropters im kontrollierten Flug meistens keine schnellen Rotationsbewegungen ausführt.

### 3. Entwicklung einer Steuerplatine

Eine Beachtung der gyroskopischen Effekte der Rotoren könnte dagegen während starker Gierbewegungen ein besseres Verhalten ermöglichen, da die Geschwindigkeitsdifferenz zwischen links- und rechtsdrehenden Rotoren  $\Omega_r$  dann einen besonders hohen Betrag hat. Dazu muss jedoch die Geschwindigkeit der Rotoren gemessen oder zumindest aus den abgegebenen Steuerimpulsen geschätzt werden. Das gleiche gilt für die Übertragung von Drehimpuls zwischen Rotoren und Rahmen, wenn die Geschwindigkeit der Rotoren gegen den Rahmen geändert wird.

Es ist möglich, dass die Antriebssoftware die eingestellten Vorgaben insbesondere bei anspruchsvollen Flugmanövern genauer umsetzen kann, wenn diese Effekte beachtet werden. Dies würde jedoch die Komplexität der Steuersoftware erhöhen, und es konnte bereits ohne explizite Berücksichtigung von gyroskopischen Effekten, Rollmoment und Horizontalkraft eine zufriedenstellende Flugstabilität erreicht werden. Daher wurde auf eine Beachtung dieser Effekte verzichtet.

Damit werden aktuell lediglich die Schubkraft und das Bremsmoment beachtet. Bei diesem vereinfachten Modell wird die Nickbeschleunigung nur durch die Schubdifferenz zwischen dem vorderen und hinteren Rotor erklärt, die Rollbeschleunigung durch die Schubdifferenz zwischen dem linken und rechten Rotor, und die Gierbeschleunigung durch das Bremsmoment. Die vereinfachte Formel für die Winkelbeschleunigung lautet dann:

$$I_Q \dot{\vec{\omega}}_B = \begin{pmatrix} l(T_L - T_R) \\ l(T_V - T_H) \\ D_L + D_R - D_V - D_H \end{pmatrix}$$

Dabei ist  $I_Q$  der (diagonale) Trägheitstensor des gesamten Quadropters,  $\dot{\vec{\omega}}_B$  ist die Winkelbeschleunigung des Rahmens,  $l$  der horizontale Abstand der Rotoren zum Quadropterschwerpunkt,  $T_i$  der Schub des Rotors  $i \in \{V, H, L, R\}$  und  $D_i$  das Bremsmoment des Rotors  $i$ .

Es werden nun zwei weitere vereinfachende Annahmen getroffen: Erstens, dass der Quadropters an Roll- und Nickachse eine ähnliche Gewichtsverteilung besitzt, so dass  $I_{Q,h} = I_{Q,xx} \approx I_{Q,yy}$  das Trägheitsmoment des Quadropters um jede horizontale Achse ist. Zweitens, dass sich das Bremsmoment proportional zum Schub verhält und damit  $D_i = k \cdot T_i$  entspricht.

Dies wird in [Bou07] indirekt vorausgesetzt, da dort sowohl Schub als auch Bremsmoment als quadratisch zur Rotordrehzahl angenommen werden. Nimmt man noch den Gesamtschub  $F_T$  hinzu, ergibt sich mit den genannten Annahmen das folgende Gleichungssystem:

$$\begin{cases} \dot{\omega}_{B,x} &= \frac{l}{I_{Q,h}}(T_L - T_R) \\ \dot{\omega}_{B,y} &= \frac{l}{I_{Q,h}}(T_V - T_H) \\ \dot{\omega}_{B,z} &= \frac{k}{I_{Q,zz}}(T_L + T_R - T_V - T_H) \\ F_T &= T_V + T_H + T_L + T_R \end{cases}$$

$a = \frac{I_{Q,h}}{l}$  und  $b = \frac{I_{Q,zz}}{k}$  sind konstante Parameter des Quadropters. Setzt man diese in das Gleichungssystem ein und löst es nach dem Schub der einzelnen Rotoren auf, erhält man:

$$\begin{cases} T_V &= \frac{F_T}{4} + \frac{a\dot{\omega}_y}{2} - \frac{b\dot{\omega}_z}{4} \\ T_H &= \frac{F_T}{4} - \frac{a\dot{\omega}_y}{2} - \frac{b\dot{\omega}_z}{4} \\ T_L &= \frac{F_T}{4} + \frac{a\dot{\omega}_x}{2} + \frac{b\dot{\omega}_z}{4} \\ T_R &= \frac{F_T}{4} - \frac{a\dot{\omega}_x}{2} + \frac{b\dot{\omega}_z}{4} \end{cases}$$

Auf diese Weise wird aus den Vorgaben der Sollschub jedes Rotors errechnet. Zu diesem muss jedoch noch das PWM-Tastverhältnis berechnet werden. Das ist die Aufgabe der Funktion  $\text{pwm}(\text{relSchub})$ , welche als Parameter den gewünschten Schub relativ zum maximal möglichen Schub erhält. In der einfachsten Variante kann diese Funktion als  $\text{pwm}(\text{relSchub}) = \text{relSchub}$  definiert werden. Wie in Abbildung 3.6.3 zu sehen ist, weicht der tatsächlich erzeugte Schub dann allerdings teilweise deutlich vom Sollschub ab.

Diese Messwerte können aber auch verwendet werden, um eine bessere Funktion für  $\text{pwm}$  zu entwickeln. Dazu wird eine Näherungsfunktion aufgestellt, die nah an den gemessenen Datenpunkten liegt. Diese muss nicht alle gemessenen Punkte exakt enthalten, eine gute Entsprechung sowohl in den absoluten Werten als auch in der Steigung sind aber wünschenswert, damit sich die Ausgaben des Ausrichtungsreglers möglichst linear auf den Schub auswirken. Für die gemessenen Werte wurde die folgende Näherung entwickelt:

$$\text{pwm}(\text{relSchub}) = \begin{cases} (\text{relSchub} + 0,1) \cdot 0,891 & \text{falls } 0 \leq \text{relSchub} < 0,461 \\ (\text{relSchub} - 0,0129) \cdot 1,1153 & \text{falls } 0,461 \leq \text{relSchub} < 0,71 \\ (\sqrt{(\text{relSchub} - 0,71) \cdot 0,179} + 0,775) & \text{falls } 0,71 \leq \text{relSchub} \leq 1,0 \end{cases}$$

Wie in Abbildung 3.6.4 zu sehen ist, ist diese Funktion eine gute Näherung für die gemessene Schubkurve.

Die angegebene Funktion wurde in der Motorsteuerungssoftware implementiert. Daraufhin wurde eine weitere Messreihe durchgeführt, bei der der relative Sollschub in kleinen Schritten von 0 bis 1 variiert wurde. Das Ergebnis ist im rechten Graphen in Abbildung 3.6.4 dargestellt, zusammen mit dem idealen Verhalten (Sollschub=Schub) sowie dem Verhalten bei der direkten Verwendung des relativen Sollschubs als PWM-Tastverhältnis.

Eine weitere Aufgabe der Steuerungssoftware ergab sich erst aus praktischen Erfahrungen mit der Steuerplatine. Wenn ein Motor sich nicht oder nur langsam dreht und eine hohe Sollleistung für diesen Motor eingestellt wird, fließt zunächst ein sehr hoher Strom durch den Motor, wodurch die Versorgungsspannung auf der Steuerplatine stark absinken kann. Das führte bei Versuchen mehrfach dazu, dass der Mikrocontroller beim anfänglichen Beschleunigen der Rotoren abstürzte oder durch einen Brown-out Reset zurückgesetzt wurde. In anderen Fällen lief der Mikrocontroller zwar korrekt, erhielt aber während des Starts verfälschte Messwerte von den

### 3. Entwicklung einer Steuerplatine

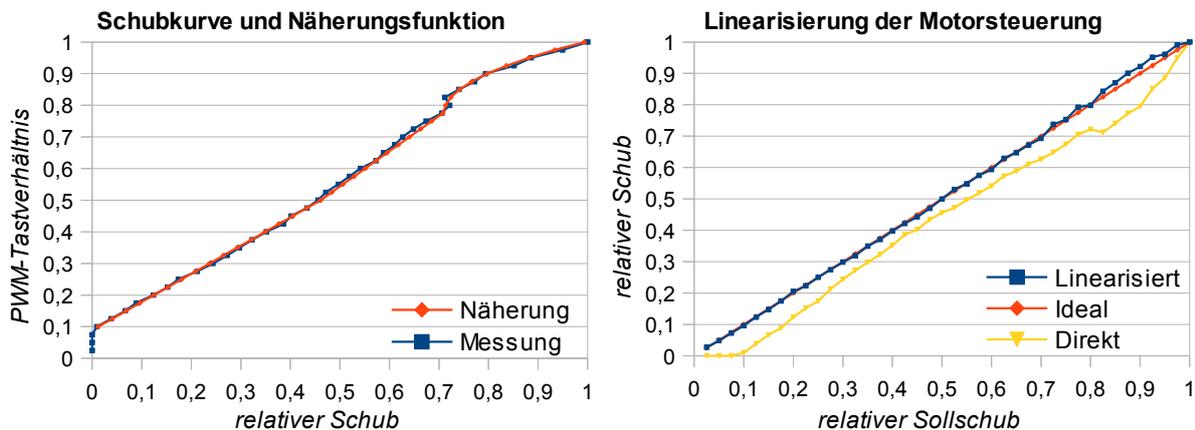


Abbildung 3.6.4.: Der linke Graph illustriert die Ähnlichkeit der entwickelten Näherungsfunktion zu den gemessenen Datenpunkten bei der verwendeten PWM-Frequenz von 4 kHz und 9 Volt. Der rechte Graph zeigt den gemessenen Schub aufgetragen gegen den Sollschub, vor und nach der Anpassung der Motorsteuerung. Das alte Verhalten ist mit „Direkt“ bezeichnet, da die Sollleistung direkt als PWM-Tastverhältnis ausgegeben wird. „Linearisiert“ bezeichnet das neue Verhalten. Zum Vergleich ist das ideale Verhalten mit abgebildet.

Sensoren. Um dieses Problem zu vermeiden, wurde die Motorsteuerung so angepasst, dass der erwartete Gesamtschub nur mit einer begrenzten Geschwindigkeit angehoben werden kann. Dadurch konnten die beschriebenen Probleme behoben werden.

### 3.7. Zustandsfilter

Bisher wurde besprochen, wie mit den verwendeten Sensoren die aktuellen Beschleunigungen und Winkelgeschwindigkeiten ermittelt werden können. Um den Quadrocopter in eine horizontale Ausrichtung für einen Schwebeflug zu regeln, oder auf einen festgelegten Winkel für einen Vorwärts- oder Seitwärtsflug, muss jedoch aus diesen Messwerten die aktuelle Ausrichtung des Quadrocopters gegenüber der Erdoberfläche bestimmt werden.

Diese Aufgabe wird vom expliziten komplementären Filter übernommen, das von Mahony et al. in [MHP08] beschrieben wurde. Eine Übersicht des Filters wurde bereits in Unterabschnitt 2.4.3 gegeben. Das Filter wurde genau für den hier auftretenden Einsatzzweck entwickelt: Die Schätzung der Ausrichtung aus gemessenen Winkelgeschwindigkeiten und weiteren Sensordaten, die von ungenauen Sensoren ermittelt werden und auf einem Embedded-System ausgewertet werden, auf dem wenig Rechenleistung zur Verfügung steht.

Da auf der Steuerplatine neben den Winkelgeschwindigkeiten nur die Schwerkraftrichtung gemessen wird, kann lediglich die Auslenkung des Quadrocopters aus der horizontalen Lage um

die Nick- und Rollachse absolut bestimmt werden. Die Ausrichtung um die Schwerkraftachse kann ohne weitere Informationen nicht ermittelt werden, da eine andere Ausrichtung auf dieser Achse zu den gleichen Messwerten führen würde. Zur vollständigen Bestimmung der Ausrichtung im erdfesten Koordinatensystem müsste daher mindestens eine zweite bekannte Richtung gemessen werden. Die absolute Ausrichtung um die Schwerkraftachse wird jedoch nicht immer benötigt, insbesondere ist sie für die Stabilisierung des Quadropters für einen Schwebeflug nicht erforderlich. Daher wurde darauf verzichtet, einen weiteren Sensor in die Steuerplatine zu integrieren.

Die Implementierung des expliziten komplementären Filters für die Steuersoftware basiert auf der Quaternionenformulierung, die im Anhang von [MHP08] beschrieben ist. Die implementierte Variante des Filters wird im Folgenden in mehreren Schritten beschrieben. Im ersten Schritt wird lediglich die Messung des Gyrosensors zum Schätzen der Ausrichtung verwendet. Im nächsten Schritt wird die Korrektur durch die Messwerte des Beschleunigungssensors hinzugefügt. Schließlich wird die Offsetkorrektur beschrieben, welche die Restabweichung zwischen geschätzter und tatsächlicher Ausrichtung eliminieren soll, die durch die Gyrodreift entsteht.

### 3.7.1. Schätzung nur nach Winkelgeschwindigkeiten

Wenn nur der Gyrosensor für die Abschätzung der Ausrichtung verwendet wird, kann die absolute Ausrichtung im erdfesten Koordinatensystem nicht bestimmt werden, da nur Änderungen der Ausrichtung gemessen werden können. Die vom Filter ermittelte Ausrichtung ist dann immer relativ zu der Ausrichtung, mit der das Filter initialisiert wird.

Die aktuelle Schätzung der Ausrichtung wird als Quaternion  $\hat{q}$  gespeichert, welche die geschätzte Rotation zwischen Richtungen im körperfesten und im erdfesten Koordinatensystem beschreibt. Ist  $\vec{v}$  ein im körperfesten Koordinatensystem ausgedrückter Vektor, erhält man die geschätzte Darstellung dieses Vektors im erdfesten Koordinatensystem durch Anwendung der in Abschnitt 2.5 definierten Funktion `rotate`:

$$\hat{\vec{v}} = \text{rotate}(\vec{v}, \hat{q})$$

Die Schätzung erfolgt schrittweise. In jedem Schritt  $n$  werden zunächst die Winkelgeschwindigkeiten um die körperfesten Achsen des Quadropters als Vektor  $\vec{\omega}_n$  gemessen. Auf dieser Grundlage wird dann aus der Schätzung des letzten Schritts  $\hat{q}_{n-1}$  die neue Schätzung  $\hat{q}_n$  berechnet. Diese Berechnung ist in [MHP08] wie folgt beschrieben, wobei  $\mathbf{p}$  die in Abschnitt 2.5 definierte Funktion ist, die einen Vektor in eine rein imaginäre Quaternion überführt:

$$\dot{\hat{q}} = \frac{1}{2} \hat{q} \cdot \mathbf{p}(\vec{\omega})$$

### 3. Entwicklung einer Steuerplatine

Daraus ergibt sich für kleine Zeitabstände  $\Delta t$  zwischen zwei Schritten:

$$\hat{q}_n \approx \hat{q}_{n-1} + \frac{\Delta t}{2} \hat{q}_{n-1} \cdot \mathbf{p}(\vec{\omega}_n)$$

Ein Nachteil dieser Näherung besteht darin, dass sie mit größeren Zeitabständen zunehmend ungenauer wird, und dass die Länge der Quaternion  $\hat{q}$  aufgrund dieser Ungenauigkeiten häufig normalisiert werden muss.

Es ist jedoch mit geringem Zusatzaufwand auch möglich, die Aktualisierung genauer durchzuführen. Unter der Annahme, dass  $\vec{\omega}$  zwischen zwei Schritten nahezu konstant bleibt, ist  $\Delta t \cdot \vec{\omega}_n$  eine Näherung für den Rotationsvektor, der die Rotation seit dem letzten Schritt beschreibt. Durch eine Verkettung mit der Rotation, die die Ausrichtung im letzten Schritt darstellt ( $\hat{q}_{n-1}$ ), erhält man dann die Ausrichtung im aktuellen Schritt ( $\hat{q}_n$ ). Um die Verkettung als Quaternionenmultiplikation berechnen zu können, wird dabei zuerst die dem Rotationsvektor entsprechende Quaternion ermittelt:

$$\hat{q}_n = \hat{q}_{n-1} \cdot \text{quaternionAusRotationsvektor}(\Delta t \cdot \vec{\omega}_n)$$

Die Funktion `quaternionAusRotationsvektor` ist in Gleichung 2.5.3 definiert.

Diese Formel wurde auf der ersten Version der Steuerplatine eingesetzt, welche noch keinen Beschleunigungssensor enthielt. Damit konnte bereits für kurze Zeit ein stabiler Schwebeflug erreicht werden. Da die Gyrodraft jedoch zu einem immer größeren Fehler führt, stellte sich der Quadrocopter allmählich schräg, so dass Testflüge in der Regel nicht länger als ca. 30 Sekunden dauerten.

#### 3.7.2. Schätzung nach Winkelgeschwindigkeiten und gemessenen Richtungen

Um die Drift auszugleichen und die absolute Ausrichtung gegenüber dem erdfesten Koordinatensystem bestimmen zu können, wird nun zusätzlich die Schwerkraftrichtung gemessen, die im erdfesten Koordinatensystem gemäß dessen Definition als  $\vec{g}_E = (0 \ 0 \ 1)^T$  bekannt ist. Der normierte Vektor  $\vec{\tilde{g}}$  stellt die gemessene Schwerkraftrichtung im Quadrocopter-Koordinatensystem dar, die aus der Ausgabe des Beschleunigungssensors ermittelt wird.

Zunächst wird berechnet, welche Richtung der Schwerkraftvektor im Quadrocopter-Koordinatensystem besäße, falls die geschätzte Ausrichtung  $\hat{q}$  der tatsächlichen Ausrichtung entspräche. Dazu wird die Richtung  $\vec{g}_E$  vom erdfesten in das Quadrocopter-Koordinatensystem transformiert, indem die inverse der von  $\hat{q}$  beschriebenen Rotation auf  $\vec{g}_E$  angewendet wird. Das Ergebnis wird als erwartete Schwerkraftrichtung  $\vec{\hat{g}}$  bezeichnet:

$$\vec{\hat{g}} = \text{rotate}(\vec{g}_E, \hat{q})$$

Der Unterschied zwischen  $\tilde{\vec{g}}$  und  $\hat{\vec{g}}$  wird als Fehler in der Schätzung der Ausrichtung interpretiert. Die Schätzung könnte nun direkt um den Rotationsvektor zwischen  $\tilde{\vec{g}}$  und  $\hat{\vec{g}}$  korrigiert werden, so dass eine erneute Berechnung von  $\hat{\vec{g}}$  mit der korrigierten Schätzung der Ausrichtung genau  $\tilde{\vec{g}}$  entspräche.

Die Messung der Erdbeschleunigung würde auf diese Weise direkt die Schätzung der Ausrichtung bestimmen, nur die Rotation um die Schwerkraftachse wäre noch von der Eingabe des Gyrosensors beeinflusst. Wie bereits in Abschnitt 2.4 beschrieben wurde, sind die Messwerte des Beschleunigungssensors jedoch von (vorwiegend hochfrequenten) Störungen überlagert, die nicht ungefiltert in die Schätzung der Ausrichtung eingehen sollten.

Die Schätzung wird daher in jedem Schritt nur um einen kleinen Anteil in Richtung des Rotationsvektors zwischen  $\tilde{\vec{g}}$  und  $\hat{\vec{g}}$  korrigiert. Die Größe dieses Anteils wird durch die wählbare Konstante  $k_P$  beeinflusst. Je kleiner  $k_P$  gewählt wird, desto geringer ist der Einfluss hochfrequenter Störungen auf die Schätzung, aber desto schwächer ist die Korrektur auch insgesamt.

In [MHP08] wird statt des Rotationsvektors zwischen  $\tilde{\vec{g}}$  und  $\hat{\vec{g}}$  das Kreuzprodukt zwischen diesen Vektoren als Grundlage für den Korrekturvektor verwendet, das im Folgenden als  $\vec{c}_P$  bezeichnet wird:

$$\vec{c}_P = \tilde{\vec{g}} \times \hat{\vec{g}}$$

$\vec{c}_P$  besitzt die gleiche Richtung wie der Rotationsvektor zwischen  $\tilde{\vec{g}}$  und  $\hat{\vec{g}}$ . Die Länge von  $\vec{c}_P$  ergibt sich nach der allgemeinen Formel für die Länge des Kreuzprodukts aus dem Winkel  $\alpha$  zwischen den Vektoren, als  $|\vec{c}_P| = |\tilde{\vec{g}}| \cdot |\hat{\vec{g}}| \cdot \sin(\alpha) = \sin(\alpha)$ , da sowohl  $\tilde{\vec{g}}$  als auch  $\hat{\vec{g}}$  normiert sind. Aufgrund der Näherung  $\sin(\alpha) \approx \alpha$  für kleine Winkel  $\alpha$  ist  $\vec{c}_P$  eine gute Näherung für den Rotationsvektor, falls der Richtungsunterschied zwischen  $\tilde{\vec{g}}$  und  $\hat{\vec{g}}$  nur gering ist.

Das Kreuzprodukt wird mit der oben eingeführten Konstante  $k_P$  skaliert. Um die Korrektur anzuwenden, wird das Ergebnis auf die gemessenen Winkelgeschwindigkeiten addiert, bevor mit diesen die Schätzung der Ausrichtung aktualisiert wird.

Die bisher besprochenen Schritte sind in den folgenden Formeln zusammengefasst:

$$\begin{aligned} \hat{q}_n &= \hat{q}_{n-1} \cdot \text{quaternionAusRotationsvektor}(\Delta t \cdot (\vec{\omega}_n + k_P \cdot \vec{c}_{P,n-1})) \\ \vec{c}_{P,n} &= \tilde{\vec{g}}_n \times \text{rotate}(\vec{g}_E, \tilde{q}_n) \end{aligned}$$

Für weitere Sensoren  $i$ , die bekannte Richtungen  $\vec{d}_{i,E}$  messen, würde ebenfalls das Kreuzprodukt zwischen gemessener und erwarteter Richtung berechnet. Die Ergebnisse aller Sensoren können dann gewichtet addiert werden, um  $\vec{c}_P$  zu erhalten:

$$\vec{c}_{P,n} = \sum_i k_i \vec{d}_{i,n} \times \text{rotate}(\vec{d}_{i,E}, \tilde{q}_n)$$

#### 3.7.3. Driftausgleich

Ein typischer Messfehler bei Gyrosensoren ist ein langsam schwankender Offset, der aufgrund von Änderungen in Temperatur, Luftdruck und ähnlichen Faktoren entsteht. Eine solche Drift kann mit den oben aufgeführten Formeln nicht vollständig ausgeglichen werden, da die Korrektur der Winkelgeschwindigkeiten direkt aus der ermittelten Abweichung zwischen geschätzter und realer Ausrichtung berechnet wird. In einem Gleichgewichtszustand, in dem die angewendete Korrektur genau dem Offset des Gyrosensors entspricht, muss daher eine bleibende Restabweichung existieren.

Um diese bleibende Abweichung zu überwinden, wird das Zeitintegral von  $\vec{c}_P$  mit einer weiteren wählbaren Konstante  $k_I$  skaliert und als zusätzlicher Korrekturwert auf die gemessenen Winkelgeschwindigkeiten addiert. Die Wirkung ähnelt der des I-Anteils in einem PI-Regler, der ebenfalls eingesetzt wird, um Restabweichungen auszugleichen.

Da die Umsetzung auf der Firmware der Steuerplatine in diskreten Schritten mit Zeitabstand  $\Delta t$  erfolgt, wird das Integral als  $\vec{c}_I$  numerisch angenähert. Das gesamte Filter wird dann durch die folgenden Gleichungen beschrieben:

$$\begin{aligned}\hat{q}_n &= \hat{q}_{n-1} \cdot \text{quaternionAusRotationsvektor}(\Delta t \cdot (\vec{\omega}_n + k_P \cdot \vec{c}_{P,n-1} + k_I \cdot \vec{c}_{I,n-1})) \\ \vec{c}_{P,n} &= \vec{g}_n \times \text{rotate}(\vec{g}_E, \hat{q}_n) \\ \vec{c}_{I,n} &= \vec{c}_{I,n-1} + \Delta t \cdot \vec{c}_{P,n}\end{aligned}$$

#### 3.7.4. Besonderheiten der Implementierung

Die Implementierung des Filters in der Steuersoftware enthält mehrere Besonderheiten, die nicht direkt aus der Beschreibung des Filters ersichtlich sind. So wird die Quaternion  $\hat{q}$  alle 256 Schritte normalisiert, um eine Längenänderung der Quaternion durch Ungenauigkeiten bei den Berechnungen auszugleichen.

Eine weitere Besonderheit ist, dass die Länge des Integrals  $\vec{c}_I$  so beschränkt wird, dass  $k_I \cdot \vec{c}_I$  nicht deutlich größer als der maximale Wert werden kann, der für die Gyrodrift erwartet wird. Der Grund dafür ist, dass in manchen Flugsituationen ein systematischer Fehler durch echte Beschleunigungen entstehen kann, der nicht auf Gyrodrift zurückzuführen ist, vom Filter aber als solche interpretiert würde.

Als Beispiel fliege der Quadrocopter bei einer Auslenkung um  $45^\circ$  nach rechts eine konstante Rechtskurve, so dass er sich dauerhaft auf gleicher Flughöhe um die Schwerkraftachse dreht und dabei eine Kreisbahn beschreibt. In dieser Konstellation entspricht der Betrag der Zentripetalbeschleunigung dem der Erdbeschleunigung, da der Schub der Rotoren durch den Winkel von  $45^\circ$  gleichermaßen auf beiden Achsen wirkt. Der Beschleunigungssensor auf dem Quadrocopter misst eine konstante Beschleunigung von  $\sqrt{2}g$ , welche direkt in Richtung der z-Achse des Quadrocopters zeigt.

Da das explizite komplementäre Filter nur die Richtung des gemessenen Vektors berücksichtigt, kann dieses Filter nicht zwischen dieser Situation und einer solchen unterscheiden, in der der Quadrocopter in horizontaler Lage steht und der Gyrosensor eine sehr hohe Drift aufweist. Ein Simulationsversuch hat bestätigt, dass das Filter in der beschriebenen Situation tatsächlich zu einer solchen horizontalen Schätzung der Ausrichtung gelangen kann. Eine sinnvoll gewählte Beschränkung von  $\vec{c}_I$  verringert diesen unerwünschten Effekt, ohne den Driftausgleich in normalen Flugsituationen zu beeinträchtigen.

Schließlich kann der Rechenaufwand für die Transformation des Schwerkraftvektors in das Quadrocopter-Koordinatensystem gegenüber einer normalen Vektorrotation deutlich verringert werden, da der Schwerkraftvektor  $\vec{g}_E$  fest bekannt ist:

$$\begin{aligned}
 \hat{q} &= (\hat{q}_a, \hat{q}_b, \hat{q}_c, \hat{q}_d) \\
 \hat{g} &= \text{rotate} \left( \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \bar{q} \right) \\
 &= \mathbf{p}^{-1}((\hat{q}_a, -\hat{q}_b, -\hat{q}_c, -\hat{q}_d) \cdot (0, 0, 0, 1) \cdot (\hat{q}_a, \hat{q}_b, \hat{q}_c, \hat{q}_d)) \\
 &= \mathbf{p}^{-1}((\hat{q}_d, -\hat{q}_c, \hat{q}_b, \hat{q}_a) \cdot (\hat{q}_a, \hat{q}_b, \hat{q}_c, \hat{q}_d)) \\
 &= \mathbf{p}^{-1}((0, 2(\hat{q}_b\hat{q}_d - \hat{q}_a\hat{q}_c), 2(\hat{q}_a\hat{q}_b + \hat{q}_c\hat{q}_d), \hat{q}_a^2 - \hat{q}_b^2 - \hat{q}_c^2 + \hat{q}_d^2)) \\
 &= \begin{pmatrix} 2(\hat{q}_b\hat{q}_d - \hat{q}_a\hat{q}_c) \\ 2(\hat{q}_a\hat{q}_b + \hat{q}_c\hat{q}_d) \\ \hat{q}_a^2 - \hat{q}_b^2 - \hat{q}_c^2 + \hat{q}_d^2 \end{pmatrix}
 \end{aligned}$$

### 3.8. Ausrichtungsregler

Aus den Informationen über die Ausrichtung, die mit Hilfe des Zustandsfilters ermittelt werden, sowie der Zielausrichtung, die z.B. von der Fernsteuerung vorgegeben wird, müssen nun Stellwerte für den Antrieb erzeugt werden. Dabei soll die Zielausrichtung nach Möglichkeit schnell erreicht und auch unter Störeinflüssen (z.B. durch Wind oder Kollisionen) gehalten werden.

Die verwendete PID-Regeltechnik wurde bereits in Abschnitt 2.6 beschrieben. Diese Technik soll nun zur Stabilisierung des Quadrocopters auf allen drei Rotationsachsen eingesetzt werden. Dazu werden drei PID-Regler verwendet, von denen jeder die Leistung für die Winkelbeschleunigung um eine der Achsen des Quadrocopters steuert. Ein Vorteil dieser Zuordnung besteht darin, dass die Regelparameter auf die Eigenschaften der entsprechenden Rotationsachsen angepasst werden können. Dies kann hilfreich sein, wenn Unterschiede zwischen den Motoren bestehen, welche für die Beschleunigung um die entsprechende Achse benötigt werden.

### 3. Entwicklung einer Steuerplatine

Für einen stabilen Schwebeflug ist insbesondere die Auslenkung aus der horizontalen Lage von Bedeutung. Daher wird die kürzeste Rotation, die eine beliebige horizontale Ausrichtung herstellt, als Grundlage für die Regelung verwendet. Da es sich dabei genau um die Rotation zwischen dem lokalen z-Vektor des Quadropters und dem globalen z-Vektor handelt, steht der Rotationsvektor senkrecht auf diesen beiden Richtungen und besitzt daher keinen Gier-Anteil. Er beschreibt daher eine gleichzeitige Drehung um die Roll- und Nickachse, die von den beiden entsprechenden Reglern ausgeführt werden kann.

Dazu werden die Projektionen des Rotationsvektors auf x- und y-Achse des Quadropters als Regelgröße für Roll- und Nickregler verwendet. Der Sollwert für die Regler ist für den Schwebeflug 0, kann aber auch durch den rechten Steuerhebel der Fernsteuerung vorgegeben werden, die dadurch die gewünschte Neigung bestimmt.

Die Rotation um die Gierachse muss ebenfalls geregelt werden. Dabei ist eine möglichst geringe Winkelgeschwindigkeit vorteilhaft, da eine Gierbewegung bei Schräglage des Quadropters die Regelgrößen von Nick- und Rollregler beeinflusst. Eine schnelle Rotation um die Gierachse würde auch zu Wechselwirkungen zwischen Nick- und Rollbewegung durch gyroskopische Effekte führen, die in der Regelung nicht berücksichtigt werden.

Eine absolute Regelung der Gierausrichtung erscheint allerdings nicht als sinnvoll, da diese mit den vorhandenen Sensoren nicht gemessen werden kann. Daher verwendet der Gierregler die Winkelgeschwindigkeit um die Gierachse als Regelgröße und bestimmt diese direkt aus den Messwerten des Gyrosensors. Dies entspricht ebenfalls der Interpretation der Fernsteuerungseingabe, so dass diese auch hier proportional als Sollwert für den Regler benutzt werden kann.

Nach dem experimentellen Ermitteln geeigneter Regelparameter wurden verschiedene Versuche zur Stabilität durchgeführt. Ein Video zu einem solchen Testflug ist auch auf der CD enthalten, die dieser Arbeit beiliegt.

Im ungestörten Schwebeflug werden sowohl die horizontale Ausrichtung als auch der Gierwinkel beibehalten. Durch kleine Abweichungen von der horizontalen Lage ergeben sich geringe Beschleunigungen in wechselnde Richtungen. Ein Störeinfluss durch ungleichmäßige Beladung des Quadropters wird durch den I-Anteil der Roll- und Nickregler ausgeglichen, solange die notwendige Leistung für diesen Ausgleich unter der I-Begrenzung der Regler liegt.

Eine plötzlicher manuelle Auslenkung aus der horizontalen Lage wird schnell und mit wenig Nachschwingen ausgeglichen. Nach kleinen Störungen auf der Gierachse dreht sich der Quadropter in die Ausgangsrichtung zurück, obwohl der Regler auf dieser Achse nicht die Ausrichtung, sondern die Winkelgeschwindigkeit regelt. Dies ist durch den I-Anteil des Reglers begründet, welcher das Integral der Winkelgeschwindigkeiten auf dieser Achse auf 0 regelt. Wenn durch eine stärkere Auslenkung um die Gierachse die Begrenzung des I-Anteils des Reglers erreicht wird, wird die ursprüngliche Ausrichtung nicht wiederhergestellt.

## 4. Fazit

### 4.1. Zusammenfassung

Es wurde ein physikalisches Modell für einen Quadrocopter beschrieben, wobei die wichtigsten Kräfte und Drehmomente erläutert wurden, die auf diesen wirken. Auf dieser Grundlage wurden Bewegungsgleichungen aufgestellt, die die Bewegung eines Quadrocopters im freien Flug beschreiben. Darüber hinaus wurde ein Überblick über wichtige Grundlagen der Ausrichtungsmessung gegeben, da die Erfassung der aktuellen Ausrichtung eine wichtige Voraussetzung für das Erreichen eines autonomen Schwebeflugs ist.

Unter Verwendung kostengünstiger Bauteile wurde eine funktionsfähige Steuerplatine entwickelt, die sowohl einen Schwebeflug als auch die Kontrolle durch eine Fernsteuerung unterstützt. Der Quelltext der entwickelten Steuersoftware ist modular aufgebaut und kommentiert. Die Steuerung kann mit einfachen Mitteln nachgebaut und nachvollzogen werden und eignet sich damit als Ausgangspunkt für eigene Weiterentwicklungen und Experimente.

Die teuersten Komponenten der Steuerplatine sind die Sensoren. Eine kurze Suche ergab, dass die Wii MotionPlus für 17,99 Euro erhältlich ist, während der Beschleunigungssensor MMA7260QT auf einer Breakout-Platine (vgl. Abbildung 3.4.1) für 9,95 Euro gekauft werden kann. Insgesamt kosten die Komponenten der vorgestellten Steuerplatine ca. 40 Euro, wobei der Aufwand der Platinenherstellung und -bestückung nicht mitgerechnet ist, da dieser je nach vorhandenem Werkzeug und Herstellungstechnik sehr unterschiedlich ausfallen kann. Der verwendete Fernsteuerungsempfänger ist in diesem Preis ebenfalls nicht enthalten und würde die Kosten beachtlich erhöhen. Die Entwicklung einer alternativen Elektronik zur Fernsteuerung, z.B. auf der Basis kostengünstiger Digitalfunkmodule, könnte in einer weiteren Arbeit untersucht werden.

Die Liste der Bauteile für die Steuerplatine ist im Anhang der Arbeit angegeben, der auch den Schaltplan und das Platinenlayout enthält. Diese sind außerdem als Eagle-Dateien auf dem Datenträger enthalten, der dieser Arbeit beiliegt. Dort befindet sich auch der Quelltext der Steuersoftware, der mit dem avr-gcc-Compiler bzw. WinAVR und dem enthaltenen Makefile übersetzt werden kann. Für die Software gilt die freie und einfach verständliche ISC-Lizenz. Dadurch ist jedem die Weiterentwicklung und Anpassung der Steuerung für eigene Projekte erlaubt.

#### 4. Fazit

Die Stabilität des Schwebeflugs und die Robustheit gegenüber Störeinflüssen sind in einem Video demonstriert, das ebenfalls auf dem Datenträger enthalten ist.

## 4.2. Vergleich mit verwandten Projekten

Es existieren bereits einige andere Projekte, die sich mit dem Eigenbau von Quadrokoptern und passenden Steuerungen befassen, und bestehende Platinen werden oft von einzelnen Bastlern abgeändert. Daher kann hier nur auf eine kleine Auswahl von Projekten hingewiesen werden.

### 4.2.1. MikroKopter

Eines dieser Projekte ist MikroKopter (siehe [HiS]). Dort wurde eine Steuerung für Quadro-Hexa- und Oktokopter entwickelt. Beschleunigungs- und Gyrosensoren sowie ein Luftdrucksensor sind standardmäßig auf der Steuerplatine enthalten, ein Magnetfeldsensor sowie ein GPS-Empfänger können modular ergänzt werden. Die Software unterstützt viele Funktionen, z.B. kann der MikroKopter automatisch Höhe und Position halten, oder automatisch vorgegebene Koordinaten anfliegen.

Die Steuerplatine „FlightControl“ für den MikroKopter ist vierlagig aufgebaut und enthält eine große Zahl von SMD-Bausteinen. Es ist deshalb nur schwierig möglich, die Platine selbst nachzubauen. Der im Shop auf der MikroKopter-Homepage erhältliche vorbestückte Bausatz für die Platine kostet derzeit 359,95 Euro und ist damit deutlich teurer als die in dieser Arbeit vorgestellte Platine. Die Firmware für die FlightControl-Platine ist zwar im Quelltext verfügbar, die Lizenz verbietet jedoch einen Einsatz auf Hardware, die nicht von [www.mikrokopter.de](http://www.mikrokopter.de) stammt.

### 4.2.2. MultiWii

MultiWii ([mul]) ist eine Software, die auf unterschiedlichen Helikopter-Steuerplatinen eingesetzt werden kann. Der Name stammt daher, dass die Software ursprünglich nur für die Verwendung der Wii MotionPlus und der Nunchuk-Erweiterung als Gyro- und Beschleunigungssensor ausgelegt war. Inzwischen werden aber auch mehrere andere Sensoren unterstützt. Als Mikrocontroller kommt ein ATmega zum Einsatz. Damit weist die MultiWii-Steuerung eine starke Ähnlichkeit zu der Steuerung auf, die in der vorliegenden Arbeit entwickelt wurde. Für den Aufbau der Steuerung werden jedoch fertig bestückte „Arduino Pro Mini“-Mikrocontrollerplatinen vorgeschlagen. Der Preis dieser Platinen ist mit ca. 15 Euro nicht hoch, übersteigt jedoch den Preis des einzelnen Mikrocontrollers, der auf der Steuerplatine in dieser Arbeit eingesetzt wurde.

Die Software ermöglicht einen stabilisierten Flug nach absoluten Neigungswinkeln sowie einen Kunstflugmodus, in dem die Steuerung über die Winkelgeschwindigkeiten erfolgt. Der Quelltext der MultiWii-Software ist unter der GPLv3-Lizenz verfügbar und kann daher auch für eigene Projekte eingesetzt werden. Die in der vorliegenden Arbeit verwendete ISC-Lizenz bietet jedoch größere Freiheit für die Weiterverwendung der Software.

#### 4.2.3. AeroQuad

Die AeroQuad-Steuerung ([aer]) ist ebenfalls für Helikopter mit unterschiedlichen Rotorzahlen und -konfigurationen geeignet. Die günstigste verfügbare Hardwarevariante ist der AeroQuad Mini v1.0, der wie auch die MultiWii-Hardware einen ATmega-Mikrocontroller auf einem Arduino Pro Mini-Board einsetzt. Als Gyro- und Beschleunigungssensor werden jedoch die SMD-ICs ITG3200 von Invensense und ADXL345 von Analog Devices auf eigenen Breakout-Platinen eingesetzt. Der Bausatz ist für 99,95 Dollar erhältlich.

Durch den modularen Aufbau der AeroQuad-Software, die unter der GPLv3-Lizenz verwendet werden kann, können neue Sensoren in eigenen Bibliotheken angebunden werden. Die Schätzung der Ausrichtung und die Flugsteuerung sind ebenfalls in eigenen austauschbaren Modulen umgesetzt. Eine interessante Parallele zu der Steuerungssoftware in dieser Arbeit ist, dass das in AeroQuad enthaltene Modul „ARG“ (Accelerometer, Rate Gyro) zur Schätzung der Ausrichtung ebenfalls auf der Quaternionenformulierung des expliziten komplementären Filters basiert.

### 4.3. Ausblick

Auf der Basis der entwickelten Steuerung können Experimente zu alternativen Techniken durchgeführt werden. Beispielsweise kann der Regelalgorithmus der Software ausgetauscht werden, um einen Vergleich in der Flugstabilität zwischen unterschiedlichen Regeltechniken zu untersuchen. Hierbei wäre es insbesondere interessant, einen Regler zu entwerfen, welcher weitere Effekte des in Abschnitt 2.2 besprochenen physikalischen Modells berücksichtigt. Auch ein Vergleich verschiedener Zustandsfilter bietet sich an, wobei für die sinnvolle Implementierung eines erweiterten Kalman-Filters oder eines Partikelfilters vermutlich eine größere Rechenleistung benötigt wird, als der verwendete ATmega zur Verfügung stellt.

Durch eine Erweiterung der Hardware um einen Magnetfeldsensor würde die Messung einer zweiten erdfesten Richtung ermöglicht, so dass eine Bestimmung der absoluten Ausrichtung auch um die Gierachse möglich wird. Ferner könnten auch Sensoren zur Bestimmung der absoluten Position eingesetzt werden, z.B. ein Luftdrucksensor oder ein nach unten gerichteter Abstandssensor zur Bestimmung der Höhe bzw. der Höhe über dem Boden. Ein GPS-Empfänger würde schließlich eine vollständige Positionserfassung ermöglichen, so dass der Quadrocopter autonom eine Position halten oder festgelegte Wegpunkte ansteuern könnte.



## A. Schaltplan und Platinenlayout

Im Verzeichnis *Schaltplan und Platinenlayout* auf der dieser Arbeit beiliegenden CD befinden sich die Eagle-Dateien ([Cad]) zu Version 1 und Version 2 der Steuerplatine, so wie sie ursprünglich aufgebaut wurden. In beiden Fällen wurden jedoch nachträglich Korrekturen durch das Durchtrennen von Leiterbahnen und Herstellen neuer Verbindungen durchgeführt, die hier beschrieben werden.

Bei Version 1 der Platine wurde eine fehlende Verbindungsleitung zur Debug-Schnittstelle ergänzt, die aufgrund eines Fehlers beim Erstellen des Schaltplans nicht im Layouteditor angezeigt wurde.

Version 2 enthält Verbindungen zwischen dem Mikrocontroller und den Pins *g-Select1*, *g-Select2* und *sleep* des Beschleunigungssensors, die jedoch nachträglich entfernt wurden, da die Ausgänge des Mikrocontrollers aufgrund der unterschiedlichen Betriebsspannungen der beiden Bauteile nur auf Low-Pegel oder hochohmig eingestellt werden konnten. In der Praxis war es jedoch erforderlich, diese Pins des Beschleunigungssensors auf einen High-Pegel zu legen, um einen Messbereich von 6g auszuwählen und den Sleep-Modus konstant zu unterbinden. Die Pins wurden daher nachträglich konstant mit der Versorgungsspannung des Beschleunigungssensors verbunden.

Die Stromversorgung des Beschleunigungssensors wurde ebenfalls geändert. Ursprünglich wurde die 3,6V-Versorgungsspannung direkt aus der 5V-Versorgung erzeugt, dies führte jedoch zu Problemen beim Einschalten, da die Spannung der 5V-Versorgung nicht ausreichend schnell ansteigt, um die im Datenblatt des Sensors angegebenen Vorgaben zu erfüllen. Der Sensor lieferte daher manchmal grob falsche Messwerte. Daher wird der Beschleunigungssensor nun über den I/O-Pin PD3 des Mikrocontrollers versorgt, der erst nach Erreichen einer stabilen 5V-Versorgungsspannung eingeschaltet wird.

Die ebenfalls in diesem Verzeichnis enthaltene Version 3 wurde nicht im Rahmen dieser Arbeit aufgebaut, es handelt sich statt dessen um ein optimiertes Layout auf der Grundlage des Schaltplans der nachträglich korrigierten Version 2. Dieses Layout ist daher für den Aufbau einer neuen Platine geeignet, die der Steuerplatine im finalen Zustand des Quadropters entspricht. Der Schaltplan der dritten Version sowie das Platinenlayout sind in diesem Anhang abgedruckt.

## A. Schaltplan und Platinenlayout

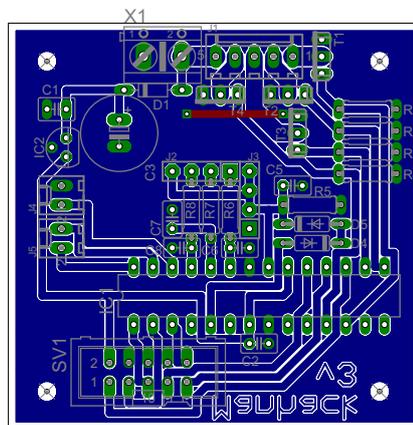


Abbildung A.1.: Die dritte Revision des Platinenlayouts in Originalgröße. Die rote Verbindung konnte nicht auf der Platinenunterseite eingeplant werden und wurde durch eine Drahtbrücke realisiert.

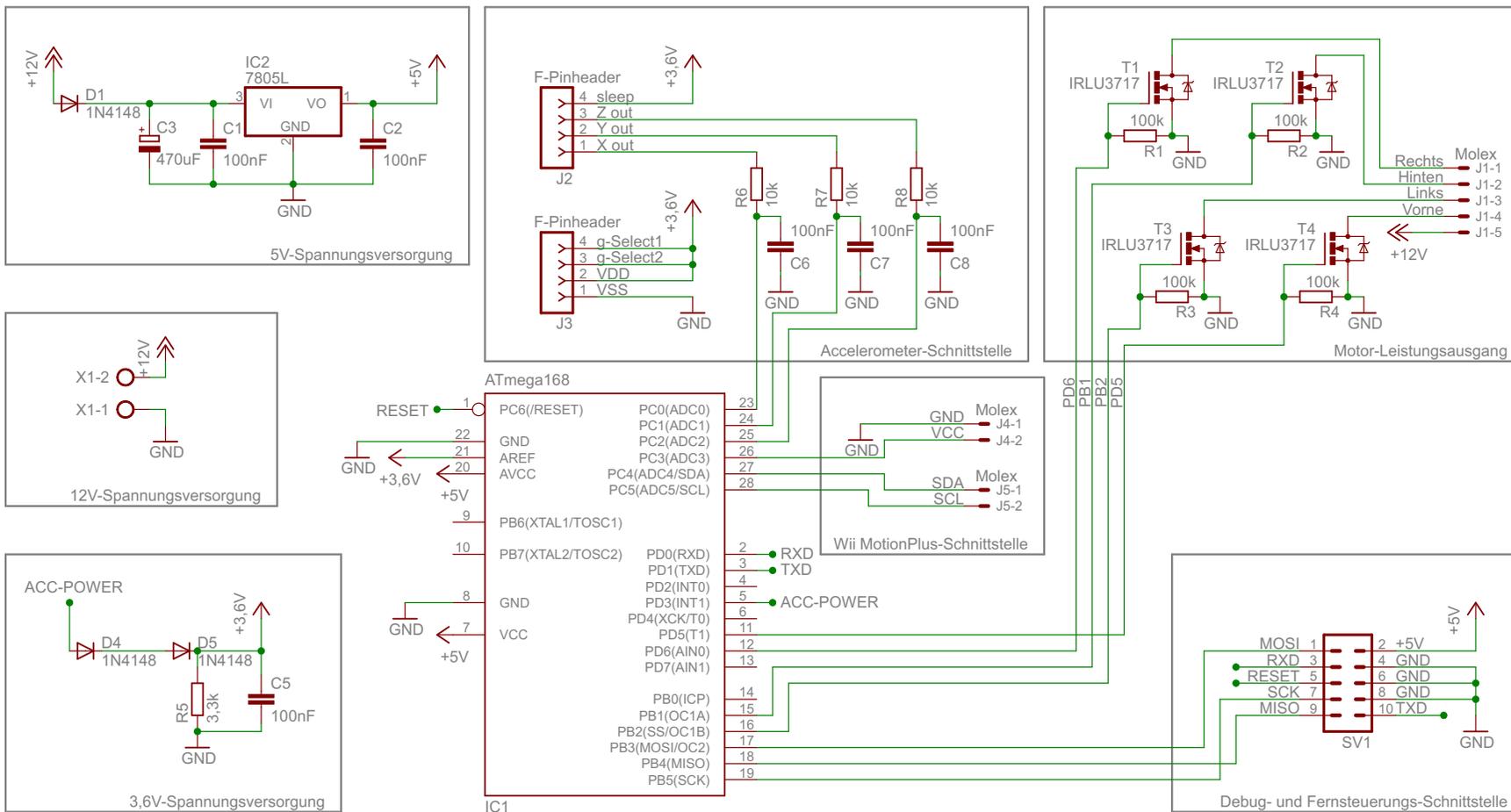


Abbildung A.2.: Der Schaltplan des Quadropters in der finalen Version. Dies entspricht dem Stand des Platinenlayouts in Version 3.

## A. Schaltplan und Platinenlayout

Bezeichnung	Name	Anzahl
J1	Molex-Buchse 5-Pol, Raster 2,54mm	1
J2, J3	F-Pinheader 4-Pol, Raster 2,54mm	2
J4, J5	Molex-Buchse 2-Pol, Raster 2,54mm	2
SV1	Wanne 10-Pol, Raster 2,54mm	1
X1	Anschlussklemme 2-Pol, Raster 5,08mm	1
C1, C2, C5, C6, C7, C8	Kondensator 100nF	6
C3	Kondensator 470µF	1
D1, D4, D5	Diode 1N4148	3
R1, R2, R3, R4	Widerstand 100kΩ	4
R5	Widerstand 3,3kΩ	1
R6, R7, R8	Widerstand 10kΩ	3
T1, T2, T3, T4	MOSFET IRLU3717	4
IC1	Mikrocontroller ATmega168	1
(IC1)	IC-Sockel 28-Pol (optional)	1
IC2	Spannungsregler 78L05	1
	Wii MotionPlus	1
	Beschleunigungssensor MMA7260QT auf Breakout-Board	1
	Platine einseitig, 55mm x 55mm	1
	Molex-Stecker 5pol, Raster 2,54mm	1
	Molex-Stecker 2pol, Raster 2,54mm	2
	M-Pinheader 4-Pol, Raster 2,54mm	2

Tabelle A.1.: Materialliste für den Aufbau der Quadrocoptersteuerung. Die Bauteile, die direkt auf der Platine eingesetzt werden, sind mit der entsprechenden Bezeichnung aus Schaltplan und Platinenlayout versehen. Die Wii MotionPlus wird wie in Abschnitt 3.3 beschrieben mit Anschlussleitungen versehen, die in den beiden 2-Pol Molex-Steckern enden, die mit J4 und J5 verbunden werden. Der 5-Pol Molex-Stecker wird mit der Stromversorgung der Motoren verbunden, wobei ein Pin gemeinsam für die +12V-Versorgung aller Motoren verwendet wird. Die beiden 4-Pol M-Pinheader werden an das Breakout-Board des Beschleunigungssensors MMA7260QT angelötet, um ein steckbares Modul zu erhalten, das in die beiden F-Pinheader J2 und J3 eingesteckt werden kann (siehe Abbildung 3.4.1). Die Leitungen der Wii MotionPlus und der Motoren können auch direkt angelötet werden. Dadurch können die Molex-Steckverbindungen eingespart werden, was Kosten und Gewicht senkt und die Zuverlässigkeit erhöht.

# Literaturverzeichnis

- [aer] Webseite des AeroQuad-Projekts. <http://www.aeroquad.com>, Abruf: April 2012
- [Atm09] ATMEL CORPORATION (Hrsg.): *ATmega48/88/168 Datasheet*. 2545R. Atmel Corporation, Juli 2009. [http://www.atmel.com/dyn/resources/prod\\_documents/doc2545.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf)
- [Bar02] BARKHORDARIAN, Vrej: *Power MOSFET Basics*. <http://www.irf.com/technical-info/appnotes/mosfet.pdf>. Version: August 2002
- [Ber03] BERNSTEIN, Jonathan: *An Overview of MEMS Inertial Sensing Technology*. <http://www.sensorsmag.com/sensors/acceleration-vibration/an-overview-mems-inertial-sensing-technology-970>. Version: Februar 2003
- [BGK<sup>+</sup>98] BOLZ, Dr. J. ; GREHN, Joachim ; KRAUSE, Joachim ; KRÜGER, Herwig ; SCHMIDT, Dr. Herbert K. ; SCHWARZE, Dr. H.: *Metzler Physik*. 3. Auflage. Schroedel Verlag, 1998. – ISBN 3-507-10700-7
- [Bor11] BORNSCHEIN, Bernhard: *X-UFO Varianten*. <http://www.bornschein-online.de/modellbau/x-ufo/x-ufo.html>. Version: April 2011, Abruf: Dezember 2011
- [Bou07] BOUABDALLAH, Samir: *Design and control of quadrotors with application to autonomous flying*. 2007
- [BS05a] BOUABDALLAH, Samir ; SIEGWART, Roland: Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (2005)*, April, S. 2259–2264
- [BS05b] BOUABDALLAH, Samir ; SIEGWART, Roland: Towards Intelligent Miniature Flying Robots. In: *Proceedings of Field and Service Robotics (2005)*
- [Buc09] BUCHNER, Steffen: *Beschleunigungssensoren*. <http://userpages.uni-koblenz.de/~physik/informatik/Sensoren/beschleunigung.pdf>. Version: Juli 2009
- [Cad] CadSoft Computer GmbH: *CadSoft EAGLE PCB Design Software*. <http://www.cadsoft.de/eagle-pcb-design-software/?language=de>, Abruf: März 2012

- [Dem94] DEMTRÖDER, Prof. Dr. W.: *Experimentalphysik 1 - Mechanik und Wärme*. Bd. 1. Springer-Verlag, 1994. – ISBN 3–540–56543–4
- [DJ08] DOUCET, Arnaud ; JOHANSEN, Adam M.: A Tutorial on Particle Filtering and Smoothing: Fifteen years later. (2008), Dezember. [http://www.cs.ubc.ca/~arnaud/doucet\\_johansen\\_tutorialPF.pdf](http://www.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf)
- [Dog09] DOGP: *Wii Controller Objects (Nunchuck, Classic Controller, MotionPlus)*. <http://forums.parallax.com/showthread.php?p=833324#post833324>. Version: August 2009, Abruf: Mai 2012
- [Eps] EPSON TOYOCOM CORP. (Hrsg.): *XV-3500 - Ultra Small Vibration Gyro Sensor*. Epson Toyocom Corp., <http://www.epsontoyocom.co.jp/english/product/Sensor/set01/xv3500cb/index.html>, Abruf: Mai 2012
- [Fay01] FAY, Gary: Derivation of the Aerodynamic Forces for the Mesicopter Simulation. (2001), Februar
- [Fis10] FISHER, Christopher J.: *Using an Accelerometer for Inclination Sensing*. [http://www.analog.com/static/imported-files/application\\_notes/AN-1057.pdf](http://www.analog.com/static/imported-files/application_notes/AN-1057.pdf). Version: 2010
- [Fre08] FREESCALE SEMICONDUCTOR, INC. (Hrsg.): *±1.5g - 6g Three Axis Low-g Micromachined Accelerometer*. 5. Freescale Semiconductor, Inc., Februar 2008. [http://cache.freescale.com/files/sensors/doc/data\\_sheet/MMA7260QT.pdf](http://cache.freescale.com/files/sensors/doc/data_sheet/MMA7260QT.pdf)
- [GWA07] GREWAL, Mohinder S. ; WEILL, Lawrence R. ; ANDRES, Angus P.: *Global Positioning Systems, Inertial Navigation, and Integration*. 2. Auflage. Wiley-Interscience, 2007. – ISBN 978–0470041901
- [Ham44] HAMILTON, Sir William R.: On Quaternions; or on a new System of Imaginaries in Algebra. In: *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science* 25 (1844), Juli, S. 10–13
- [HHS<sup>+</sup>10] HARTUNG, Stefan ; HUMBERT, Yohan ; SCHLEINING, Eduard ; SCHÖNFELD, Nicolas ; THEISEN, Dirk: *Projektausarbeitung - Headtrackersteuerung des RollerCoaster 2000*. 2010
- [HiS] HiSystems GmbH: *Webseite des MikroKopter-Projekts*. <http://mikrokopter.de>, Abruf: April 2012
- [HTLMB06] HINDRICHSEN, C. C. ; THOMSEN, E. V. ; LOU-MØLLER, R. ; BOVE, T.: MEMS Accelerometer with Screen Printed Piezoelectric Thick Film. In: *5th IEEE Conference on Sensors* (2006), Oktober, S. 1477–1480
- [Int05] INTERNATIONAL RECTIFIER (Hrsg.): *IRLR/U3717 Datasheet*. International Rectifier, Januar 2005. <http://www.irf.com/product-info/datasheets/data/irlr3717.pdf>, Abruf: Mai 2012

- [Inv] INVENSENSE, INC. (Hrsg.): *IDG-650 Integrated Dual-Axis Gyroscope*. InvenSense, Inc., <http://invensense.com/mems/gyro/idg650.html>, Abruf: Mai 2012
- [IOI<sup>+</sup>] IWATA, Satoru ; OTA, Keizo ; ITO, Kuniaki ; WAKITANI, Noboru ; TAKAMOTO, Junji: *Iwata Asks - Wii MotionPlus*. [http://www.nintendo.co.uk/NOE/en\\_GB/news/iwata/iwata\\_asks\\_wii\\_motionplus\\_16791\\_17205.html](http://www.nintendo.co.uk/NOE/en_GB/news/iwata/iwata_asks_wii_motionplus_16791_17205.html), Abruf: Mai 2012
- [irp] International Rectifier: *Application Note AN-940: How P-Channel MOSFETs Can Simplify Your Circuit*. <http://www.irf.com/technical-info/appnotes/an-940.pdf>
- [Ká60] KÁLMÁN, Rudolf E.: A New Approach to Linear Filtering and Prediction Problems. In: *Transactions of the ASME, Journal of Basic Engineering* 82D (1960), März, S. 35–45
- [Kes99] KESTER, Walt: *Practical Design Techniques for Sensor Signal Conditioning*. Analog Devices, Inc., 1999. – ISBN 0–916550–20–6
- [KG04] KIM, Anthony ; GOLNARAGHI, M.F.: A Quaternion-Based Orientation Estimation Algorithm Using an Inertial Measurement Unit. In: *Position Location and Navigation Symposium* (2004), April, S. 268–272
- [Kis] Kistler Instrumente GmbH: *Vorteile piezoelektrischer Messplattformen*. [http://www.kistler.com/de\\_de-de/642\\_Biomechanics\\_Advantages/Vorteile-piezoelektrischer-Messplattformen.html](http://www.kistler.com/de_de-de/642_Biomechanics_Advantages/Vorteile-piezoelektrischer-Messplattformen.html), Abruf: Dezember 2011
- [Lei06] LEISHMAN, J. G.: *Principles of helicopter aerodynamics*. 2. Ausgabe. Cambridge University Press, 2006
- [Loc88] LOCHER, Ralph: *Introduction to Power MOSFETs and Their Applications*. <http://www.national.com/an/AN/AN-558.pdf>. Version: Dezember 1988
- [LPL<sup>+</sup>03] LYNCH, Jerome P. ; PARTRIDGE, Aaron ; LAW, Kincho H. ; KENNY, Thomas W. ; KIREMIDJIAN, Anne S. ; CARRYER, Ed: Design of Piezoresistive MEMS-Based Accelerometer for Integration with Wireless Sensing Unit for Structural Monitoring. In: *Journal of Aerospace Engineering* 16 (2003), Juli, Nr. 3
- [MHP08] MAHONY, Robert ; HAMEL, Tarek ; PFLIMLIN, Jean-Michel: Nonlinear Complementary Filters on the Special Orthogonal Group. In: *IEEE Transactions on Automatic Control* 53 (2008), Juni, Nr. 5, S. 1203–1218
- [MLD11a] MÜLLER, Mark ; LUPASHIN, Sergei ; D'ANDREA, Raffaello: Quadrocopter Ball Juggling. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2011). <http://www.idsc.ethz.ch/people/staff/mueller-m/IrosJugglingPaper.pdf>

- [MLD11b] MÜLLER, Mark ; LUPASHIN, Sergei ; D'ANDREA, Raffaello: Quadrocopter Ball Juggling, ETH Zurich. (2011). <http://youtu.be/3CR5y8qZf0Y>
- [Moo09] MOODY, Miles: *Wii Motion Plus + Arduino = Love*. <http://randomhacksofboredom.blogspot.com/2009/06/wii-motion-plus-arduino-love.html>. Version: Juni 2009, Abruf: Mai 2012
- [mul] *Webseite des MultiWii-Projekts*. <http://www.multiwii.com>, Abruf: April 2012
- [NEC07] NEC ELECTRONICS CORPORATION (Hrsg.): *78K0/KC2 User's Manual*. 5. NEC Electronics Corporation, März 2007
- [roba] ROBBE MODELLSPORT GMBH (Hrsg.): *Empfaenger R118F 40 MHz*. robbe Modellsport GmbH, <http://www.robbe.de/empfaenger-r118f-40-mhz.html>, Abruf: Mai 2012
- [robb] ROBBE MODELLSPORT GMBH (Hrsg.): *F-14 4/8/1 40 MHz*. robbe Modellsport GmbH, <http://www.robbe.de/f-14-4-8-1-40-mhz.html>, Abruf: Mai 2012
- [RW12] RN-WISSEN: *Regelungstechnik — RN-Wissen, die Wissensbasis der Roboternetz Community*. <http://rn-wissen.de/index.php?title=Regelungstechnik&oldid=18862>. Version: 2012, Abruf: Mai 2012
- [Sch] SCHNABEL, Patrick: *MOS-Feldeffekttransistor (MOS-FET)*. <http://www.elektronik-kompodium.de/sites/bau/0510161.htm>, Abruf: November 2011
- [WB06] WELCH, Greg ; BISHOP, Gary: *An Introduction to the Kalman Filter*. (2006), Juli
- [wii] *WiiBrew, a wiki dedicated to homebrew on the Nintendo Wii*. [http://wiibrew.org/wiki/Main\\_Page](http://wiibrew.org/wiki/Main_Page), Abruf: Mai 2012
- [Wii11a] WIIBREW: *Wiimote/Extension Controllers*. [http://wiibrew.org/w/index.php?title=Wiimote/Extension\\_Controllers&oldid=98087](http://wiibrew.org/w/index.php?title=Wiimote/Extension_Controllers&oldid=98087). Version: 2011, Abruf: Mai 2012
- [Wii11b] WIIBREW: *Wiimote/Extension Controllers/Wii Motion Plus*. [http://wiibrew.org/w/index.php?title=Wiimote/Extension\\_Controllers/Wii\\_Motion\\_Plus&oldid=97399](http://wiibrew.org/w/index.php?title=Wiimote/Extension_Controllers/Wii_Motion_Plus&oldid=97399). Version: 2011, Abruf: Mai 2012
- [Wik12a] WIKIPEDIA: *I<sup>2</sup>C — Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=I%C2%B2C&oldid=103062246>. Version: 2012, Abruf: Mai 2012
- [Wik12b] WIKIPEDIA: *PID controller — Wikipedia, The Free Encyclopedia*. [http://en.wikipedia.org/w/index.php?title=PID\\_controller&oldid=492852338](http://en.wikipedia.org/w/index.php?title=PID_controller&oldid=492852338). Version: 2012, Abruf: Mai 2012

- [Wik12c] WIKIPEDIA: *Quaternion* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Quaternion&oldid=93015245>.  
Version: 2012, Abruf: Mai 2012
- [Wik12d] WIKIPEDIA: *Quaternion* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Quaternion&oldid=492041711>.  
Version: 2012, Abruf: Mai 2012
- [Wik12e] WIKIPEDIA: *Quaternions and spatial rotation* — *Wikipedia, The Free Encyclopedia*. [http://en.wikipedia.org/w/index.php?title=Quaternions\\_and\\_spatial\\_rotation&oldid=492009183](http://en.wikipedia.org/w/index.php?title=Quaternions_and_spatial_rotation&oldid=492009183). Version: 2012, Abruf: Mai 2012
- [Wik12f] WIKIPEDIA: *Roll-Nick-Gier-Winkel* — *Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Roll-Nick-Gier-Winkel&oldid=92715789>. Version: 2012, Abruf: Mai 2012
- [Wik12g] WIKIPEDIA: *Servomechanism* — *Wikipedia, The Free Encyclopedia*. <http://en.wikipedia.org/w/index.php?title=Servomechanism&oldid=487599280>. Version: 2012, Abruf: Mai 2012
- [Xev] XEVEL: *Slow and fast modes*. [http://www.assembla.com/wiki/show/alicewiimotionplus/slow\\_and\\_fast\\_modes](http://www.assembla.com/wiki/show/alicewiimotionplus/slow_and_fast_modes), Abruf: Mai 2012
- [Zie08] ZIEGLER, Peter-Michael: *Geschäft mit Überwachungs-Flugdrohnen boomt*. (2008), Januar. <http://heise.de/-178765>