

## Dissertation



# Autonome Quadrokopter zur Innenraumerkundung AQopterI8 Forschung und Entwicklung



Nils Gageik



Würzburg 2015

Autor: Nils Christian Gageik (geboren in Aachen)

Fertig gestellt am 24.09.20015

Eingereicht am: 02.10.2015

Bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr. Sergio Montenegro (Universität Würzburg, Aerospace Information Technology)
2. Gutachter: Prof. Dr. Arno Fischer (FH Brandenburg, Vizepräsident für Technologietransfer, Betriebssysteme / Rechnernetze)

Disputation:

1. Prüfer: Prof. Dr. Reiner Kolla
2. Prüfer: Prof. Dr. Sergio Montenegro
3. Prüfer: Prof. Dr. Arno Fischer

Beisitzer: Prof. Dr. Hakan Kayal

Druck: [www.druckterminal.de](http://www.druckterminal.de)

KDD Kompetenzzentrum Digital-Druck GmbH

Leopoldstraße 68 \* D-90439 Nürnberg

# Zusammenfassung

Diese Forschungsarbeit beschreibt alle Aspekte der Entwicklung eines neuartigen, autonomen Quadropters, genannt AQopterI8, zur Innenraumerkundung. Dank seiner einzigartigen modularen Komposition von Soft- und Hardware ist der AQopterI8 in der Lage auch unter widrigen Umweltbedingungen autonom zu agieren und unterschiedliche Anforderungen zu erfüllen. Die Arbeit behandelt sowohl theoretische Fragestellungen unter dem Schwerpunkt der einfachen Realisierbarkeit als auch Aspekte der praktischen Umsetzung, womit sie Themen aus den Gebieten Signalverarbeitung, Regelungstechnik, Elektrotechnik, Modellbau, Robotik und Informatik behandelt. Kernaspekt der Arbeit sind Lösungen zur Autonomie, Hinderniserkennung und Kollisionsvermeidung.

Das System verwendet IMUs (Inertial Measurement Unit, inertielle Messeinheit) zur Orientierungsbestimmung und Lageregelung und kann unterschiedliche Sensormodelle automatisch detektieren. Ultraschall-, Infrarot- und Luftdrucksensoren in Kombination mit der IMU werden zur Höhenbestimmung und Höhenregelung eingesetzt. Darüber hinaus werden bildgebende Sensoren (Videokamera, PMD), ein Laser-Scanner sowie Ultraschall- und Infrarotsensoren zur Hinderniserkennung und Kollisionsvermeidung (Abstandsregelung) verwendet. Mit Hilfe optischer Sensoren kann der Quadropter basierend auf Prinzipien der Bildverarbeitung Objekte erkennen sowie seine Position im Raum bestimmen. Die genannten Subsysteme im Zusammenspiel erlauben es dem AQopterI8 ein Objekt in einem unbekanntem Raum autonom, d.h. völlig ohne jedes externe Hilfsmittel, zu suchen und dessen Position auf einer Karte anzugeben. Das System kann Kollisionen mit Wänden vermeiden und Personen autonom ausweichen. Dabei verwendet der AQopterI8 Hardware, die deutlich günstiger und Dank der Redundanz gleichzeitig erheblich verlässlicher ist als vergleichbare Mono-Sensor-Systeme (z.B. Kamera- oder Laser-Scanner-basierte Systeme).

Neben dem Zweck als Forschungsarbeit (Dissertation) dient die vorliegende Arbeit auch als Dokumentation des Gesamtprojektes AQopterI8, dessen Ziel die Erforschung und Entwicklung neuartiger autonomer Quadropter zur Innenraumerkundung ist. Darüber hinaus wird das System zum Zweck der Lehre und Forschung an der Universität Würzburg, der Fachhochschule Brandenburg sowie der Fachhochschule Würzburg-Schweinfurt eingesetzt. Darunter fallen Laborübungen und 31 vom Autor dieser Arbeit betreute studentische Bachelor- und Masterarbeiten.

Das Projekt wurde ausgezeichnet vom Universitätsbund und der IHK Würzburg-Mainfranken mit dem Universitätsförderpreis der Mainfränkischen Wirtschaft und wird gefördert unter den Bezeichnungen „Lebensretter mit Propellern“ und „Rettungshelfer mit Propellern“ [1]. Außerdem wurde die Arbeit für den Gips-Schüle-Preis nominiert. Absicht dieser Projekte ist die Entwicklung einer Rettungsdrohne. In den Medien Zeitung, Fernsehen und Radio wurde über den AQopterI8 schon mehrfach berichtet.

Die Evaluierung zeigt, dass das System in der Lage ist, voll autonom in Innenräumen zu fliegen, Kollisionen mit Objekten zu vermeiden (Abstandsregelung), eine Suche durchzuführen, Objekte zu erkennen, zu lokalisieren und zu zählen. Da nur wenige Forschungsarbeiten diesen Grad an Autonomie erreichen, gleichzeitig aber keine Arbeit die gestellten Anforderungen vergleichbar erfüllt, erweitert die Arbeit den Stand der Forschung.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>7</b>
<b>2</b>	<b>Stand der Technik.....</b>	<b>11</b>
2.1	Überblick Stand der Technik .....	12
2.2	Begriffliches der Drohnentechnik.....	12
2.3	Sensorik .....	16
2.4	Orientierungsdarstellungen .....	26
2.5	Regelungstechnik für Quadrocopter .....	32
2.6	Algorithmen und Methoden.....	36
2.7	Verwandte Systeme ( <i>Related Works</i> ).....	44
2.8	Zusammenfassung.....	51
<b>3</b>	<b>Konzept.....</b>	<b>55</b>
3.1	Gesamtkonzept AQopter18.....	56
3.2	Konzeptvergleich: AQopter18 vs. Stand der Technik.....	59
3.3	Innovative Aspekte .....	61
3.4	Datenfusion mit Weighted Filter.....	65
3.5	Datenfusion mit Triple Awareness Filter (TAF).....	70
<b>4</b>	<b>Hardware .....</b>	<b>73</b>
4.1	Überblick Hardware.....	74
4.2	Elektronik.....	74
4.3	Mechanische Konstruktion.....	78
<b>5</b>	<b>Signalverarbeitung (Sensorik).....</b>	<b>85</b>
5.1	Überblick Signalverarbeitung .....	86
5.2	Orientierungsbestimmung .....	86
5.3	Höhenbestimmung.....	95
5.4	Positionsbestimmung.....	96
5.5	Hinderniserkennung und Situationserfassung .....	97
<b>6</b>	<b>Regel- und Steuerung (Aktuatorik).....</b>	<b>105</b>
6.1	Überblick Regel- und Steuerung.....	106
6.2	Überblick Regelung.....	106

6.3	Regelung (6 DOF).....	107
6.4	Superposition .....	109
6.5	Kollisionsvermeidung .....	110
6.6	Flight Management (Steuerung) .....	114
6.7	Weitere Funktionen zur Regel- und Steuerung.....	118
<b>7</b>	<b>Softwarearchitektur.....</b>	<b>129</b>
7.1	Überblick Softwarearchitektur .....	130
7.2	Klassisches Softwaredesign .....	130
7.3	Rodos-basiertes Softwaredesign.....	133
7.4	Operator- und Piloteninterface.....	134
<b>8</b>	<b>Evaluierung.....</b>	<b>141</b>
8.1	Überblick Evaluierung.....	142
8.2	Orientierungsbestimmung .....	142
8.3	Höhenbestimmung.....	147
8.4	Lageregelung .....	149
8.5	Höhenregelung.....	151
8.6	Kollisionsvermeidung und Abstandsregelung .....	152
8.7	Autonomer Flug.....	157
8.8	Objektsuche.....	161
8.9	Gerichteter Flug.....	165
<b>9</b>	<b>Quadrocopter in der Lehre.....</b>	<b>175</b>
9.1	Lehre Heute .....	176
9.2	Konzept der Lehre .....	176
9.3	Evaluierung der Lehre.....	177
<b>10</b>	<b>Diskussion und Ausblick .....</b>	<b>185</b>
10.1	Diskussion.....	186
10.2	Ausblick.....	187
<b>11</b>	<b>Anhang .....</b>	<b>189</b>
11.1	Weitere Graphen zur Evaluierung des Systems .....	189
11.2	Abkürzungsverzeichnis .....	195
11.3	Zitate und Kapitelbilder .....	195

11.4	Historie des AQopterI8-Projekts .....	196
11.5	Arbeiten am AQopterI8-Projekt .....	203
11.6	Wissenschaftliche Veröffentlichungen.....	205
11.7	Beiträge in Medien zum AQopterI8.....	206
11.8	Das AQopterI8 Team .....	207
11.9	Quellen- und Literaturverzeichnis .....	207
<b>12</b>	<b>Danksagung .....</b>	<b>219</b>

*Therefore the Lord God sent him forth from the Garden of Eden,  
he drove out the man, and at the east of the Garden of Eden  
he placed Cherubim and a flaming sword,  
which turned every way, to keep the way  
of the tree of life.  
[Genesis, 3:23-24]*

# 1 Einleitung



# Einleitung

Im letzten Jahrzehnt hat das Interesse an unbemannten Luftfahrzeugen (Drohnen) stetig zugenommen. Gleichzeitig wuchsen die Zahlen der wissenschaftlichen Arbeiten und Anwendungen auf diesem Gebiet. Unter den Drohnen erfreuen sich die Quadro- und Multikopter aufgrund ihrer Vorteile bestehend aus Einfachheit und Flexibilität von Konstruktion und Steuerung, Robustheit sowie geringen Kosten besonderer Beliebtheit.

Als Teilgebiet der Robotik kann die Drohnentechnik heute als eigenes wissenschaftliches Spezialgebiet verstanden werden, das aufgrund der Betrachtung eines fliegenden Systems besondere Anforderungen an den Agenten stellt. Die Unterschiede zwischen Drohnen und Robotern sind letztlich darauf zurückzuführen, dass sich im Allgemeinen Roboter am Boden bewegen, während Drohnen fliegen. Eine Verschiebung der Anforderungen erfolgt daher ganz natürlich. Dazu gehören Gewichtsreduktion und aerodynamische Optimierung, die Wahl und Anpassung von Sensorik, Aktuatorik, Signalverarbeitung, Navigation, Steuerung und Regelung für und auf fliegende Systeme sowie deren besonders effiziente Programmierung in Hinblick auf Limitierungen von Rechenkapazität und Speicher. Dies sind zwar größtenteils allgemeine Fragestellungen der Robotik, bei Drohnen treten diese Probleme jedoch in ein besonderes Licht oder sind verschärft.

Ein zentraler Aspekt der Drohnentechnik ist die Autonomie. Vergleichbar mit den Fortschritten und Vorteilen der industriellen Automatisierung können mit zunehmender Autonomie der Drohnen, wie dies allgemein in der Robotik angestrebt wird, Aufgaben effizienter, zuverlässiger, günstiger und schneller erledigt werden.

Kernaspekt der Autonomie ist die Signalverarbeitung zur Bestimmung der eigenen Position und der Positionen bzw. Abständen von und zu Hindernissen. Dabei hat jede Sensorik seine eigenen Schwächen, weshalb heute kein autonomer Quadrocopter existiert, der in beliebigen Situationen operieren kann. Hinzu kommt, dass oftmals Systeme als autonom bezeichnet werden, die gleichzeitig auf externe Hilfsmittel wie Signale und Ressourcen angewiesen sind. Da diese Systeme ohne die besagten Hilfsmittel nicht operabel sind, wäre die Bezeichnung solcher Systeme als semi-autonom zutreffender. Auf die konkreten Unterschiede wird im Stand der Technik mit Beispielen näher eingegangen. In dieser Arbeit wird der Begriff der Autonomie in Abgrenzung dazu enger gefasst.

Insbesondere der Betrieb in Innenräumen stellt eine besondere Herausforderung an das Gesamtsystem und dessen Autonomie dar, weil in Gebäuden meist wenig Platz zum Navigieren zur Verfügung steht. Gleichzeitig müssen Kollisionen aktiv vermieden werden. Daher und aufgrund der bereits erwähnten inhärenten Nachteile der verschiedenen Sensortechnologien sind Lösungen zur Hinderniserkennung und Kollisionsvermeidung ein zentraler Aspekt dieser Arbeit sowie der wissenschaftlichen Forschung im Bereich autonomer Drohnentechnik. Durch Verwendung multipler kostengünstiger Sensoren und deren Datenfusion präsentiert die Arbeit auf diesem Gebiet neue Lösungen.

Dabei befasst sich diese Arbeit mit allen Aspekten eines autonomen Systems, da die unterschiedlichen Bereiche ineinander greifen und einander beeinflussen. Die komplette Realisierung dieser innovativen Systemlösung wird aufgezeigt. Dabei wird der Fokus auf eine kostengünstige Umsetzung und eine aussagekräftige Evaluierung durch empirische Experimente gelegt.



Ein Anwendungsszenario für den AQopter18 ist der Feuerwehreinsatz, z.B. der Flug in verrauchten oder unzugänglichen Räumen. Dazu gehören nicht nur Häuser, sondern auch Tunnel, Höhlen, Schornsteine, Bergwerke und Katakomben. Damit ein Quadrocopter in beliebigen Umgebungen operieren kann, ist eine leistungsfähige und zuverlässige Sensorik und Situationserfassung notwendig. Zu dem beschriebenen Problem und den anschließenden Fragestellungen im Bereich Regelung und Steuerung sowie zur Realisierung des Gesamtsystems und Durchführung einer Objektsuche als Pendant zur Suche nach Verletzten zeigt diese Arbeit neue Lösungen auf.



*A moment comes, which comes but rarely in history,  
when we step out from the old to the new,  
when an age ends, and when the soul of a nation,  
long suppressed, finds utterance.*  
[Jawaharlal Nehru]

## 2 Stand der Technik



## 2.1 Überblick Stand der Technik

Das Kapitel Stand der Technik behandelt vorrangig terminologische, technische und mathematische Grundlagen, Methoden und Systeme zur Klärung der Begriffe und Hintergründe, die in der weiteren Arbeit verwendet werden und in dessen Gesamtkontext die Arbeit eingebettet ist. Der Inhalt dieses Kapitels ist eine Komposition aus dem auf diesem Forschungsgebiet gebräuchlichen Grundlagenwissen und weiterer spezieller Quellen.

## 2.2 Begriffliches der Drohnentechnik

In diesem Abschnitt werden die Begriffe Quadrocopter, Autonomie und Drohne sowie verwandte Terme näher erläutert. Da dies wesentliche Schlagwörter der behandelten Thematik sind und auch weil in den Medien und der Literatur nicht immer eine einheitliche Definition existiert, ist eine genaue Abgrenzung erforderlich.

### 2.2.1 Drohnen

Drohne ist der deutsche Begriff für ein unbemanntes Luftfahrzeug. Der Begriff Drohne hat in der öffentlichen Wahrnehmung aufgrund der militärischen Verwendung (Predator, Reaper) keinen guten Ruf. In Fachkreisen werden häufig die englischen Abkürzungen wie UAV (unmanned aerial vehicle), MAV (micro air vehicle), UAS (unmanned aerial system) und unbemannte VTOL (vertical take-off and landing) verwendet, um nur einige zu nennen. Eine fachlich begründete Abgrenzung zwischen Drohne und UAV existiert nicht, da beide Begriffe dasselbe meinen, wenngleich die Begriffe Drohne wie auch das englische *drones* aufgrund der militärisch-historischen Vorbelastung in einigen Kreisen verpönt sind [2].

### 2.2.2 Quadrocopter und Multikopter

Unter einem Quadrocopter, auch Quadcopter, Quadrocopter oder Quadcopter, im englischen meist Quadrotor genannt, versteht man ein Luftfahrzeug mit vier Propellern (vgl. Abbildung 2.1). Der Quadrocopter gehört wie der Helikopter zu den Drehflüglern, d.h. er erzeugt seinen Auftrieb im Gegensatz zu einem Flugzeug nicht durch starre Tragflächen sondern durch nach oben gerichtete drehende Tragflächen, sogenannte Rotoren bzw. Propeller.



Abbildung 2.1: AQuadrocopter (Quadrocopter)

Im Gegensatz zum Helikopter verwendet der Quadrocopter vier Rotoren bzw. vier Motoren. Dementsprechend sind die Rotoren beim Quadrocopter kleiner als beim Helikopter, besitzen also eine geringere Trägheit und sind an den vier Auslegern angebracht. Um das System in der Waagerechten zu halten, ist eine permanente Regelung der vier Motoren erforderlich. Der Helikopter wird durch Änderung des Anstellwinkels gesteuert, wozu eine zusätzliche Rotorkopfmechanik erforderlich ist. Somit ist der mechanische Aufbau des Helikopters komplexer und empfindlicher als der Aufbau eines Quadrocopters.

Die vier Propeller des Quadrocopters sind notwendig, um die vier Freiheitsgrade (DOF, degree of freedom) bestehend aus der Orientierung (Roll-Pitch-Yaw) und der Höhe unabhängig voneinander manipulieren zu können. Durch Manipulation der vier Freiheitsgrade lassen sich dann die beiden

translatorischen Freiheitsgrade in der z-Ebene beliebig erreichen. Mit weniger Aktuatoren lassen sich diese Freiheitsgrade nicht mehr unabhängig voneinander steuern. Die beiden Propeller jeweils einer Achse drehen sich in dieselbe Richtung, während sich die beiden Propeller der anderen Achse in die entgegengesetzte Richtung drehen. Somit kann das Drehmoment des Quadropters um die z-Achse ausgeglichen werden. Da dieser Aufbau mechanisch sowie steuerungs- und regelungstechnisch die einfachste Konfiguration darstellt, erfreuen sich Quadropters besonderer Beliebtheit, wengleich jede Konfiguration mit zusätzlichen Propellern darüber hinaus denkbar ist. All diese verschiedenen Bauformen werden unter dem Oberbegriff Multikofter zusammengefasst. Weitere verbreitete Bauformen von Multikoftern sind Hexakofter (Abbildung 2.2) und Octokofter (Abbildung 2.3). Die Vorteile dieser Bauformen sind die erhöhte Nutzlast durch den zusätzlichen Auftrieb sowie höhere Systemverlässlichkeit im Falle des Ausfalles eines Motors bzw. Propellers. Da der Quadropter nicht mehr flugfähig ist, sofern ein Motor ausfällt, stürzt er in einem solchen Szenario unweigerlich ab oder dreht sich unkontrolliert um eine Achse. Verglichen mit anderen Ursachen für Abstürze ist dies jedoch ein seltenes Szenario. Der AQopter18 ist während seiner Entwicklung aufgrund unterschiedlichster Ursachen (meist Bedienungsfehler, Softwarefehler oder Sensorausfall aufgrund von I<sup>2</sup>C Fehlern) dutzende, wenn nicht hunderte Male abgestürzt, aber keinmal wegen eines defekten Motors. Für ein Forschungsprojekt ist also der Quadropter aufgrund seiner Einfachheit und Robustheit das Mittel der Wahl.



Abbildung 2.2: Hexakofter MK-Hexa [3]



Abbildung 2.3: Octokofter MK ARF Okto XL 6S12 [3]

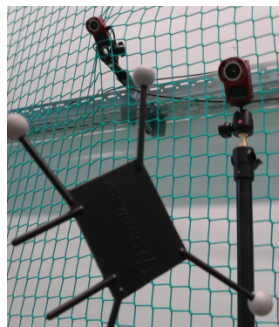
Eine Ausnahme sind Trikofter, die zwar lediglich drei Propeller besitzen, aber einen zusätzlichen Aktuatoren, z.B. einen Servomotor, benötigen, womit einer der Motoren gekippt wird. Somit ist es dem Trikofter ebenfalls möglich, die bereits erwähnten vier Freiheitsgrade unabhängig voneinander zu manipulieren. Da beim Trikofter im Gegensatz zu den Multikoftern wie dem Quadropter nicht alle Motoren starr sind, ist der Trikofter eher den Helikoftern zuzuordnen oder zumindest als Hybrid anzusehen [4] [3].

### 2.2.3 Autonomie und Position

Das Wort Autonomie stammt von dem altgriechischen Wort *autonomia* ab und lässt sich mit Selbstständigkeit übersetzen [5]. Ein autonomer Quadropter ist demnach selbstständig. Soweit sind sich die Wissenschaftler einig. Was aber nun zur Selbstständigkeit (Autonomie) dazu gehört und was nicht selbstständig ist, daran scheiden sich die Geister. Der Begriff Autonomie hat einen solchen Hype erfahren, dass beinahe jeder im Ansatz selbstständige Quadropter als autonom bezeichnet wird. Hier ist zum besseren Verständnis eine differenzierte Betrachtung erforderlich. In der

einschlägigen Fachliteratur spricht man daher auch von vollkommen autonom (fully autonomous) in Abgrenzung zu der verwässerten Verwendung des Begriffs für Systeme, die zwar Aspekte der Autonomie erfüllen, jedoch nicht „ganz so autonom“ sind wie vollkommen autonome Systeme. Mit autonom wird also im Allgemeinen eine ganze Bandbreite von Systemen bezeichnet, die auf der einen Seite von den ferngesteuerten Systemen (engl.: remotely piloted) und auf der anderen Seite von den voll autonomen Systemen eingegrenzt werden. Diese Verwendung der Begrifflichkeit erscheint inkonsequent, weshalb stattdessen im Weiteren eine Einteilung in die Kategorien *manuell*, *semi-autonom* und *autonom* Anwendung findet. An Hand einiger Beispiele soll diese Thematik im Folgenden näher erläutert werden.

In vielen Ansätzen wird die Position des Quadropters mittels optischem Tracking (OTS) bestimmt. Unter optischem Tracking (Abbildung 2.4) versteht man ein im Raum verteiltes Kamerasystem, das am Quadropters befestigte Marker detektiert und somit unter Zuhilfenahme der bekannten Kameraparameter die Position des Markers und damit des Quadropters im Raum ermittelt. Ein solcher Quadropters ist nur sehr eingeschränkt selbstständig, denn in einem unpräparierten Raum wäre diese Drohne ohne das OTS des Wissens um ihre eigene Position beraubt. Somit wäre keine Navigation möglich und das System wäre nicht in der Lage, seine Aufgabe zu erfüllen.



**Abbildung 2.4: Optisches Trackingsystem (OTS) bestehend aus 2 Kameras und einem Rigid Body mit 3 Markern**

Ein anderes Beispiel für ein semi-autonomes System ist ein Quadropters mit einer Kamera, der die Kamerabilder über Funk an einen Computer zur Verarbeitung schickt. Der Computer verarbeitet die Bilder, bestimmt die Position des Quadropters und steuert diesen entsprechend. Auch dieser Quadropters ist nicht autonom, da zwar im Gegensatz zum ersten Beispiel die Sensorik on-Board ist, er dennoch eine ständige Funkverbindung zum ortsfesten Computer benötigt und somit nur in der Nähe des Computers operieren kann.

Ein weiteres Beispiel ist ein Quadropters mit einem GPS (Global Positioning System) Receiver zur Positionsbestimmung und Navigation, wie er heute bereits gekauft werden kann. Auch dieses System ist permanent auf das funktionierende GPS angewiesen und versagt in Gebäuden, Tunneln und anderen Szenarien, wo kein GPS-Empfang möglich ist (engl.: GPS-denied) oder das GPS Signal gestört wird, wie z.B. durch GPS-Jammer oder GPS-Spoofing. Während ein GPS-Jammer das GPS Signal lediglich stört, wird beim GPS-Spoofing das GPS Signal manipuliert, wodurch aktiv auf die Steuerung des Quadropters eingewirkt werden kann.

Die eigene Position im Raum sowie die Umgebung stets und möglichst exakt zu bestimmen, sind die Wesentlichkeiten eines autonomen Systems und der differenzierende Aspekt zu semi-autonomen Systemen. Bei einem Quadropters ist dies umso schwieriger, da er als fliegendes Objekt stets in

Bewegung ist (instabiles System) und als autonomes System seine vollständige Sensorik und Rechenkapazität mit sich führt. Dementsprechend sind diese Ressourcen limitiert und mitgeführte Sensoren werden von den Bewegungen des Quadropters verfälscht, was wiederum über die Regelung rückgekoppelt wird und die Dynamik des Systems einschränkt.

Zusammenfassend kann festgestellt werden, dass das Wissen um die eigene Position und die Umgebung eine der größten Herausforderung bei der Autonomie eines Quadropters darstellt. Dies ist ein konzeptioneller Aspekt des Gesamtsystems und findet besondere Berücksichtigung bei der Implementierung. Folglich kann dies als substantieller Part nicht weggelassen werden. Die dafür benötigten Sensoren und Rechner erhöhen das Gewicht und verschlechtern die dynamischen Eigenschaften des Systems. Zudem sind aufgrund der verfügbaren Sensorik und Rechenkapazität andere Genauigkeiten mit autonomen als mit semi-autonomen Systemen zu erreichen. Ohne diese Hardware on-Board wäre die Drohne aber gar nicht erst in der Lage, autonom zu agieren [6] [7] [8] [9].

#### 2.2.4 Flight Director

In der Luftfahrt ist der Flight Director (FD) ein Fluginstrument, das dem Piloten hilft, eine gewünschte Flugbahn zu fliegen. Der Pilot bzw. das Flight Management System geben eine gewünschte Trajektorie vor und der Flight Director ermittelt daraus die benötigten Sollwerte für die Aktuatoren wie Lage und Geschwindigkeit des Flugzeuges. Dies erleichtert dem Pilot das Folgen einer gewünschten Flugbahn, sofern er manuell fliegt, oder aber der FD dient dem Autopiloten als Sollwertgeber [10].

Der Begriff Flight Director setzt sich zusammen aus Flight (Flug) und Director (techn. Englisch für Sollwertgeber) und lässt sich demnach mit Flugwertgeber übersetzen, wenngleich auch im Deutschen der englische Begriff Anwendung findet [11].

Übertragen auf einen autonomen Quadropten entspricht die Funktion des Flight Directors einem Wegpunkt- und Positionsregler, welcher ebenfalls Sollwerte zur Einhaltung einer bestimmten Trajektorie ermittelt, wobei sich dies nur bedingt übertragen lässt, da in einem Quadropten zur Positionsregelung nicht notwendigerweise eine Trajektorienregelung gehört, zumindest nicht im stationären Fall und der Positionsregler sich zudem stets der Positionsbestimmung bedient und nicht nur auf Lageinformationen zurückgreift. Dieser Sachverhalt ist der Tatsache geschuldet, dass das Flugzeug als nicht-holonomes System nicht jede Position aus jeder Richtung anfliegen kann und somit die Trajektorienregelung bei einem Flugzeug wesentlich komplexer als bei einem Quadropten ist.

#### 2.2.5 Flight-Management System

Ein Flight Management System (FMS) ist allgemein eine elektronische Hilfe für den Piloten, um dessen Arbeit im Bereich Navigation und Flugsteuerung zu erleichtern und setzt somit auf einer Ebene oberhalb eines FDs auf. Das FMS ermittelt Trajektorien, welche wiederum dem Flight Director als Eingabe dienen. Es vereinfacht zudem die Planung und Optimierung während des Fluges. Dabei sorgt das Flight Management System dafür, dass ein vorgegebener Flugplan eingehalten wird, indem es diesen mit Hilfe von Sensorik zur Positionsbestimmung (INS, Radar, GPS) überprüft und ggf. die Trajektorie anpasst [10] [11].

Auf einen autonomen Quadropten übertragen entspricht diese Funktion weitestgehend der Wegpunktgenerierung und Navigation, wobei die Trajektoriengenerierung, wie schon erwähnt, bei

einem Quadropter wesentlich leichter fällt.

Somit ist das FMS das Analogon zu einem GPS basierten Navigationssystem, wie man es vom Auto oder Handy her kennt, während die Vorgaben, ob man geradeaus fahren oder abbiegen soll, dem FD nahe kommen. Der Autofahrer bzw. Pilot macht dann nur noch die „Lenkbewegungen“ und steuert das Gas. Im Fall eines Autopiloten oder autonomen Fahrzeugs wird ihm auch das abgenommen.

### 2.2.6 Nomenklatur

Hier stehen ein paar Hinweise zur Klärung von Formelzeichen, Begriffen und Abkürzungen [11]:

- 1.) Im Folgenden werden Matrizen **M** fettgedruckt, Vektoren  $\vec{v}$  mit Pfeil dargestellt, Quaternionen  $q$  unterstrichen und Skalare  $s$  ohne Fettdruck etc. versehen.
- 2.) XYZ steht für die Koordinatenachsen: X = Vorne, Y = Rechts, Z = Oben.
- 3.) RPY meint die Euler-Winkel Roll, Pitch und Yaw.
- 4.) INS steht für inertiales Navigationssystem (Trägheitsnavigation).
- 5.) PMD steht für Photomischdetektor (engl.: Photonic Mixing Device, S.25).
- 6.) EKF steht für erweiterten Kalman Filter (engl.: Extended Kalman Filter, S. 31).
- 7.) OD steht für Hinderniserkennung (engl.: Obstacle Detection). OD Sensoren sind Sensoren zur Hinderniserkennung.
- 8.) **R** steht für eine (3x3)-Rotationsmatrix.
- 9.)  $t$  steht für die Zeit (als Parameter / Variable).
- 10.)  $T$  steht für die Sampledauer.
- 11.)  $f$  steht für die Frequenz.
- 12.) Der Begriff Pose meint das 6-Tupel aus Position und Orientierung eines Objekts im dreidimensionalen Raum. Dies wird auch im Weiteren mit Lage im Raum bezeichnet.
- 13.)  $\Delta$  wird für die Ableitung oder die Änderung einer Größe verwendet, z.B.  $\Delta x$  die Änderung in x-Richtung oder  $\Delta t$  die Zeitdifferenz.
- 14.) SV steht für stereo-optische Abstandsmessung (engl.: Stereo-Vision, S. 23)
- 15.) TM steht für Telemetrie.
- 16.) TC steht für Telekommandos (engl: Telecommands).
- 17.) Lidar steht für optische Abstandsmessung (engl.: Light detection and ranging).
- 18.) I<sup>2</sup>C (engl. Inter-Integrated Circuit), auch TWI (engl. Two Wire Interface) genannt, ist ein serieller Bus.
- 19.) PWM (Pulsweitenmodulation) ist eine Modulationsart.
- 20.) SPI (engl. Serial Peripheral Interface) ist ein serieller Bus.
- 21.) ADC steht für Analog-Digital-Konverter (engl. Analog-Digital-Converter).
- 22.) USART (engl. Universal Synchronous/Asynchronous Receiver Transmitter) meint eine serielle Schnittstelle.

### 2.2.7 Quellen

[2] [3] [4] [6] [7] [8] [9] [10] [11]

## 2.3 Sensorik

Die Sensoren sind die Sinne einer autonomen Drohne. Wie der Mensch benötigt auch eine autonome Drohne eine Vielzahl unterschiedlicher Sinne, um sich in der Welt zu Recht zu finden. Typischerweise werden Sensoren eingeteilt in externe Sensoren wie Kameras, die die Umwelt der Drohne erfassen



und Daten über Hindernisse liefern, sowie interne Sensoren, die systemeigene Zustandsinformationen, wie z.B. die Prozessortemperatur, messen. Die Trennung, was ein systemeigener Zustandssensor ist und was nicht, ist dabei nicht immer ganz einfach. So werden IMUs (Inertial Measurement Unit) wie beinahe alle hier betrachteten Sensoren klassischerweise zu den externe Sensoren gezählt, wenngleich IMUs primär der Bestimmung der Orientierung dienen und die Orientierung keine Aussage über die Beschaffenheit der Umgebung liefert. Eine Differenzierung, also unterschiedliche Einteilung von Accelerometer und Gyroskop, erfolgt bisweilen auch, wenngleich diese Sensoren heute häufig auf einem Chip anzutreffen sind.

Daher wird im Weiteren eine Einteilung nach der räumlichen Lage des Messsubjekts verwendet. Als Umgebungssensoren werden solche verstanden, die ihre Informationen aus der entfernten Umgebung erfassen, womit das gemessene Objekt also räumlich vom Quadrokopter getrennt ist, während die Zustandssensoren ihre Informationen unmittelbar ohne einen externen „Informationsträger“ bestimmen. Damit erstellen Umgebungssensoren ein direktes Abbild des Umfeldes der Drohne, d.h. wie sieht die Außenwelt wirklich aus, wo sind Hindernisse, wie wurde sich in Bezug dazu bewegt, etc., während Zustandssensoren zunächst nur eine Aussage über Zustände des Systems selbst machen. Zustände können die Orientierung im Schwerfeld der Erde, der an der aktuellen Position herrschende Luftdruck und die Detektion von Rauch sein. Diese Zustände interferieren natürlich mit der Umgebung, die Sensoren aber messen nicht unmittelbar die Umgebung, sondern z.B. im Falle eines Rauchsensors lediglich den Rauch, der sich im Sensor befindet und welcher nicht zwangsläufig identisch mit den Rauchverhältnissen in der Umgebung ist.

Somit betreffen zunächst Störungen im Umfeld zwischen Drohne und Messsubjekt , z.B. Rauch der einen Laserscanner stört, vorrangig die externen Sensoren und nur solche Störungen, die an der Drohne anliegen oder bis an diese Drohne heranreichen, die internen Sensoren. Ein Beispiel hierzu sind Vibrationen, die den Accelerometer verfälschen. Damit lassen sich die Kategorien auch an Hand der Lage und dem Einflussbereich der Störungen voneinander abgrenzen. Diese Einteilung kann bei der weiteren Fehleranalyse und Fehleroptimierung des Sensorsystems hilfreich sein.

Autonome Drohnen besitzen verschiedene typische Sensortypen, die entsprechend obiger Definition wie folgt eingeteilt werden können (Tabelle 2.1). Alle diese Sensortypen wurden in den AQopter18 integriert bzw. im Rahmen der Arbeit erprobt.

Umgebungssensoren	Zustandssensoren	
Infrarot Abstandssensoren	Gyroskop	IMU ( Inertial Measurement Unit)
Ultraschall Abstandssensoren	Accelerometer	
PMD Kamera	Magnetsensor	
Kamera (RGB)	Drucksensor (Luftdruck)	
Laserscanner	Gassensor	
Wärmestrahlungssensoren	Rauchsensor	
Optischer Flusssensor	Temperatursensor	
Radarsensoren	Akkusensor (Spannung, Strom, Ladung)	

Tabelle 2.1: Sensortypen

Diese Einteilung kommt der Unterteilung in propriozeptive und exterozeptive Sensoren [12] sehr nahe. Propriozeptiv ist ein Begriff aus der Biologie und meint „Reize aus dem Körper wahrnehmend“

[13].

Die Sensoren lassen sich auch einteilen anhand ihres Preises, wobei in dieser Arbeit fast ausschließlich preiswerte, sogenannte Low-Cost Sensoren, eingesetzt werden. Weitere technische Einteilungen unterscheiden die Sensoren nach der Art der Datenübertragung (digital, analog) oder dem Messprinzip. Relevant für die weitere Arbeit ist besonders die Einteilung nach der Datenmenge oder Komplexität der Verarbeitung (Signalverarbeitung, Signal Processing). Sensoren, üblicherweise bildgebende Sensoren, die eine hohe Datenmenge liefern, benötigen eine aufwändige Auswertung, um aus der Fülle an Daten die Informationen zu ermitteln, die interessant sind. Solch eine Verarbeitung muss von einem leistungsstarken Prozessor, einem sogenannten High Level System wie z.B. einem Intel Atom mit 1.6 GHz Taktung, durchgeführt werden. Daher werden diese Sensoren als High Level Sensoren bezeichnet. Für Low Level Sensoren genügen typischerweise Mikrocontroller mit Taktungen im Megahertz Bereich. Diese Prozessoren liefern gleichzeitig auch die benötigten Interfaces (Schnittstellen) zu solchen Sensoren, die bei leistungsstarken Systemen unüblich sind.

Sofern man die verschiedenen Sensortypen (Sensortechnologien) angemessen vergleichen möchte, sind zusätzlich zu den Kosten des jeweiligen Sensors noch die Kosten des Rechners hinzuzurechnen. Diese Überlegungen sind auch relevant bei der Auslegung des Systems, bei der Konstruktion und Planung der Anschlüsse sowie bei der Entscheidung, welche Routinen wo integriert und damit welche Software und Funktionen von welchem Prozessor verarbeitet wird.

Die Funktionsweise dieser Sensortypen wird im Weiteren näher erläutert. Dabei wird insbesondere auf die Messgüte sowie Vor- und Nachteile eingegangen.

### 2.3.1 Infrarot Abstandssensoren

Infrarot Abstandssensoren (kurz IR-Sensoren, vgl. Abbildung 2.6) sind optische Sensoren, die einen infraroten Lichtstrahl aussenden, empfangen und üblicherweise durch Triangulation aus der Position des empfangenen Lichtstrahls auf dem Empfänger den Abstand zum Messsubjekt bestimmen. Dabei fällt der am Messobjekt diffus reflektierte Strahl auf ein PSD (Position-Sensitive-Device). PSD sind Array-Lichtsensoren (1D oder 2D), die der Bestimmung der Position eines Lichtpunkts auf dem Bildbereich des Empfängers dienen. Somit kann die Strecke  $x$ , die der Position auf dem PSD entspricht, ermittelt werden. Aus dem Abstand  $L$  zwischen Sender und Empfänger sowie einem Skalierungsfaktor  $f$  kann dann mit Hilfe von Formel 2.1 der Abstand  $D$  des Objekts ermittelt werden [12].

$$D = f \frac{L}{x} \quad \text{Formel 2.1}$$

Vorteile von Infrarot Abstandssensoren liegen in dem geringen Preis und der einfachen Ansteuerung. Die Sensoren liefern eine analoge Spannung, die im gültigen Messbereich proportional zum Abstand abfällt (vgl. Abbildung 2.5). Diese Funktion zwischen Ausgangsspannung und Abstand ist jedoch nicht bijektiv. Für den GP2Y0A710K0F [14] entspricht eine Spannung von 2.5V einem Abstand von entweder ca. 50cm oder 1m. Aufgrund dieser Mehrdeutigkeiten können später Fehlmessungen auftreten, die durch eine geeignete Verarbeitung auszuschließen sind. Dies erfolgt entweder durch Beschränkung des Messbereichs oder Fusion mit weiterer Sensorik.

Aufgrund des beschränkten Messbereichs kommen im AQopterI8 zwei Sensoren zum Einsatz. Der

GP2Y0A710K0F von Sharp, im Weiteren langer IR-Sensor (kurz IRL) genannt, hat einen Messbereich von ca. 1m bis 5,5m. Der GP2Y0A02YK [15] hat einen Messbereich zwischen 20cm und 150cm und wird daher im Weiteren kurzer IR-Sensor (kurz IRS) genannt.

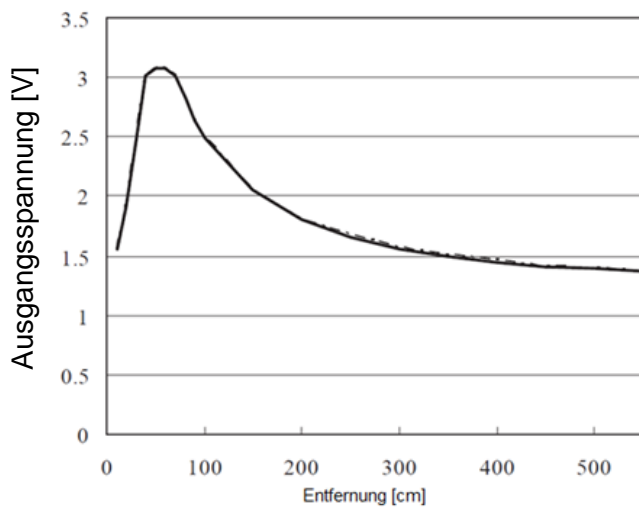


Abbildung 2.5: IR-Sensor (GP2Y0A710K0F) Ausgangsspannung [V] in Abhängigkeit zur Entfernung L in cm [Sharp, 2006]

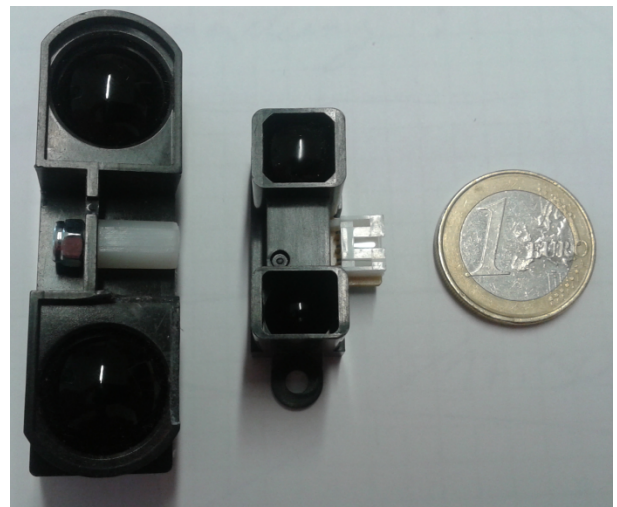


Abbildung 2.6: IR-Sensoren GP2Y0A710K0F (1m – 5,5m) und GP2Y0A02YK (20cm – 150cm) (v.l.)

Neben dem Nachteil der Mehrdeutigkeit sind die Messungen der IR-Sensoren oftmals eher unzuverlässig. Lichtdurchlässige Oberflächen, wie z.B. Wasser und Glas, werden nicht korrekt erkannt [vgl. Tabelle 11.1 A01, A02]. Für eine korrekte Messung ist eine ausreichend reflektierende Oberfläche notwendig, was in aller Regel auf größere, glatte senkrechte Flächen zutrifft, nicht aber auf abgewinkelte und unebene Oberflächen. Schon bei geringem Neigungswinkel versagen die Sensoren. Zudem haben die Sensoren einen sehr geringen Öffnungswinkel, was bei der Auswertung zu berücksichtigen ist und dazu führen kann, dass Hindernisse übersehen werden. Wie alle optischen Sensoren versagen die Sensoren bei Einschränkungen des Sichtbereichs, z.B. durch Rauch.

### 2.3.2 Ultraschall Abstandssensoren

Ultraschallsensoren (kurz US-Sensoren, vgl. Abbildung 2.8), auch Sonarsensoren genannt, bestimmen Entfernungen zu Objekten mittels Schallwellen. Die Sensoren senden Schallwellen aus, empfangen die Reflektion dieser Schallwellen und bestimmen an Hand der Laufzeit die Entfernung von Objekten. Ultraschallsensoren haben häufig einen großen Öffnungswinkel (vgl. Abbildung 2.7), was im Vergleich zu Sensoren wie IR-Sensoren Vorteile bei der Abdeckung mit sich bringt. Gleichzeitig kann aber bei diesen Sensoren nicht eindeutig bestimmt werden, wo genau sich das Objekt im „Messkegel“ befindet. Ein Vorteil von Ultraschallsensoren besteht darin, dass diese Sensoren durch Rauch messen und durchsichtige Oberflächen wie Glas detektieren können. Zudem sind die Sensoren preiswert und einfach anzusteuern. Der Messbereich von typischen Ultraschallsensoren beginnt bei ca. 20cm und reicht bis ca. 4-6m. Im Vergleich zu IR-Sensoren tolerieren US-Sensoren einen größeren Neigungswinkel. Dieser beträgt für den SRF02 ca. +/-20°, während der Öffnungswinkel ca. 55° beträgt. Die Sensoren versagen bei Objekten, die nicht ausreichend Reflektionsfläche liefern, sowie bei schallschluckenden Oberflächen wie z.B. Stoffen. Eine genaue und verlässliche Detektion von Personen mit Ultraschallsensoren ist daher schwierig, womit sich Ultraschallsensoren dazu nicht eignen [12][A01].

Die Firma Tritech vertreibt für ca. 12.000€ einen 360° Sonar Scanner (Micron Sonar: Ultra Compact

Imaging Sonar) [16] zur Anwendung in ferngesteuerten Unterwasserfahrzeugen (engl.: ROV, Remotely Operated Vehicle). Der Sensor hat mit einem Gewicht von ca. 324g einen Messbereich von 30 cm bis 7500 cm bei einer Auflösung von ca. 7,5mm.

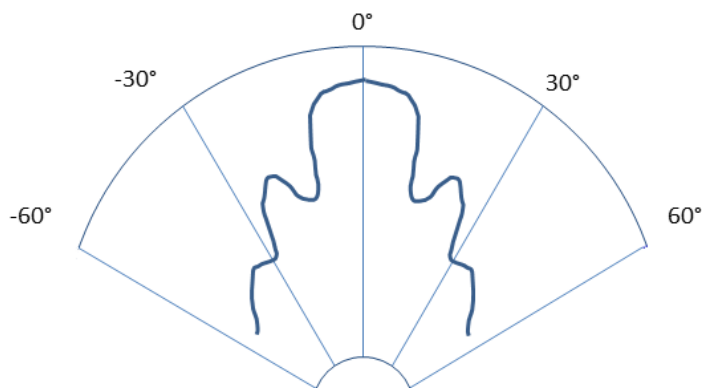


Abbildung 2.7: Charakteristische winkelabhängige Reflektionsintensität von Ultraschallsensoren

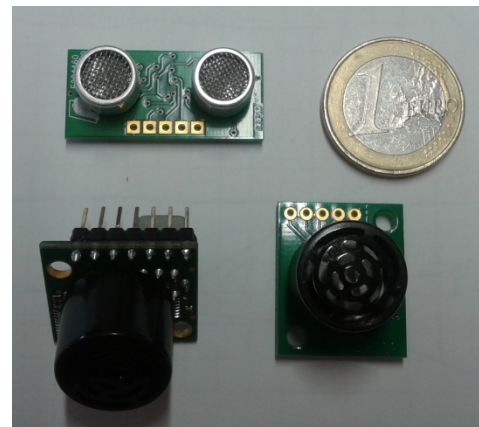


Abbildung 2.8: US-Sensoren SRF-10, MB 1202 und SRF-02 (v.o.l.)

### 2.3.3 Inertiale Messeinheit (IMU, Inertial Measurement Unit)

Eine inertielle Messeinheit, auch Inertialsystem oder Trägheitssensorsystem genannt, besteht aus einem Gyroskop (Gyrometer, Drehratensensor) und einem Accelerometer (Beschleunigungssensor). Gyroskope messen Winkelgeschwindigkeiten und somit rotatorische Bewegungen, während Accelerometer Beschleunigungen messen und damit translatorische Bewegungen detektieren können. Da die Gravitationskraft eine konstante Beschleunigung ist, kann diese als Referenz zur Orientierungsbestimmung genutzt werden. Dies funktioniert am Boden sehr gut, ist im fliegenden Quadrokopter jedoch erheblich erschwert, da Vibrationen der Motoren und die Eigenbewegung des Quadrokopters die Messungen verfälschen.

Typischerweise werden dreiaxige Sensoren zur Messung von Beschleunigungen und Drehraten aller drei Raumachsen verwendet. Mit einer 6DOF IMU ist somit ein Sensor gemeint, der sowohl Beschleunigungen als auch Drehraten in allen drei Raumachsen messen kann.

Es existieren unterschiedliche Messprinzipien, die in einem Gyroskop bzw. Accelerometer genutzt werden. Mechanische Sensoren erzielen hohe Genauigkeiten, sind aber sehr groß, schwer und teuer. Für Quadrokopter eignen sich sogenannte MEMS-Sensoren. Die Abkürzung MEMS meint ein Mikro-Elektro-Mechanisches-System und damit Chips im 1-Zentimeterbereich. Diese Sensoren sind preiswert, klein und leicht, kommen aber trotz der großen Fortschritte der letzten Jahre nicht an die Genauigkeit größerer, z.B. mechanischer Sensoren heran. Ein Problem insbesondere von MEMS-Sensoren ist der Drift, d.h. das Rauschen der Sensoren ist nicht mittelwertfrei. Dies äußert sich im sogenannten Integrationsdrift, wodurch die Messungen der Sensoren zur Bestimmung der integralen Größe nach kurzer Zeit einem erheblichen Fehler unterliegen.

Mit Hilfe eines geeigneten Systems zur Datenfusion kann aus den Rohdaten der Sensoren die Orientierung und Position ermittelt werden. Solche Systeme, die mit Hilfe der Änderung der Lage im Raum (Position und Orientierung) ausgehend von einer initialen Lage die aktuelle Orientierung und Position bestimmen, werden inertielle Navigationssysteme (INS) oder Trägheitsnavigationssysteme genannt. Da es im Gegensatz zur Orientierung für die Positionsbestimmung in einem IMU-basierten INS keine Referenz gibt, aufgrund der doppelten Integration der Messfehler exponentiell anwächst,

weil Orientierungsfehler durch falsche Gravitationskompensation unauslöschlich Positionsfehler verursachen und wegen Vibrationen, ist ein System zur Positionsbestimmung basierend auf einem INS für einen Quadrocopter ungeeignet.

Im AQopter18 finden die folgenden fünf 6 DOF IMUs Anwendung, weshalb diese kurz vorgestellt werden (vgl. Abbildung 2.9):

- 6 DOF IMU Digital Combo Board [17]
- IMU Fusion Board [17]
- MinIMU-9 [18]
- MinIMU-9 v2 [18]
- MPU-6000 EVB [19]

Das Combo Board vereint einen ITG3200 [20] und einen ADXL345 [21] auf einer Platine. Das IMU Fusion Board ist eine Weiterentwicklung des Combo Boards und ersetzt das Gyroskop durch die IMU3000 [22]. Die MinIMU-9 ist ein sogenannter 9 DOF Sensor und hat (als Besonderheit) zusätzlich zu den 6 DOF IMUs noch einen 3 DOF Magnetsensor auf derselben Platine vereint. Das MinIMU-9 besteht aus dem Gyroskop L3G4200D [23] und dem LSM303DLM [24], einer Kombination aus dreiaxsigem Accelerometer und dreiaxsigem Magnetsensor. Die Weiterentwicklung v2 kombiniert den L2GD20 (Gyroskop) [25] mit dem LSM303DLHC (Accelerometer und Magnetsensor) [26]. Das MPU-6000 EVB kombiniert auf einer Platine die 6 DOF IMUs MPU-6000 und MPU-6050 [27].

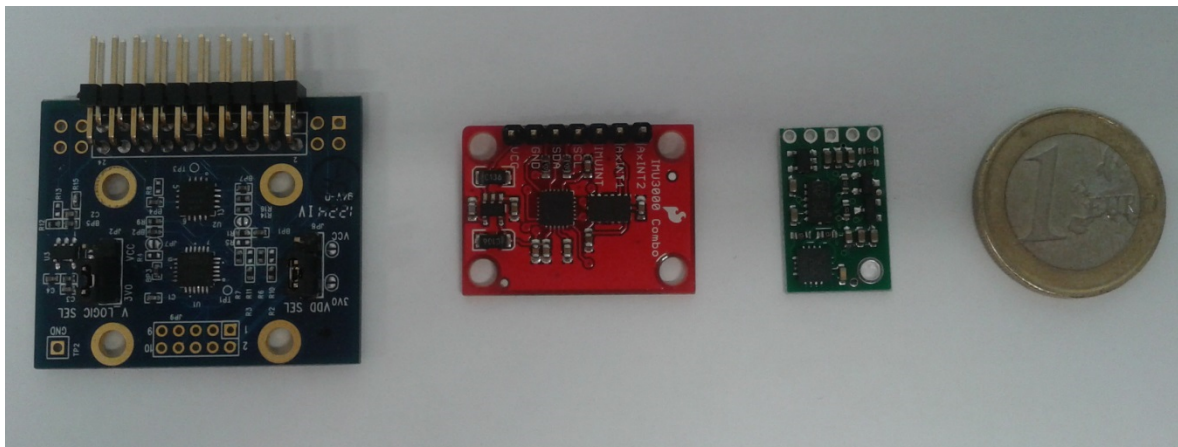


Abbildung 2.9: IMUs MPU-6000 EVB, Fusion Board, MinIMU-9 v2 (v.l.)

### 2.3.4 Magnetsensoren

Magnetsensoren werden in Quadrocoptern zur Orientierungskompensation um die z-Achse (Yaw, Gierachse) eingesetzt. Da das Gieren, d.h. drehen um die z-Achse, keinen Einfluss auf den gemessenen Gravitationsvektor hat, lässt sich ein Driften um diese Achse grundsätzlich nicht mit dem Beschleunigungssensor kompensieren. Ein Magnetsensor ist im Grunde nichts anderes als ein Kompass und misst Magnetfelder wie das Erdmagnetfeld. Dieses kann ähnlich wie der Gravitationsvektor beim Accelerometer als Referenz verwendet werden, um Drehungen zu detektieren und Fehler zu kompensieren. Zu beachten dabei ist, dass das Erdmagnetfeld durch Magnetfelder von Permanent- oder Elektromagneten (Hard-Iron Effekt) sowie durch ferromagnetische Stoffe (Soft-Iron Effekt) gestört bzw. überlagert wird und damit die Orientierungskompensation verfälscht. Eine Voraussetzung für eine korrekte Orientierungs-

kompensation ist ein weitestgehend homogenes Magnetfeld. Insbesondere innerhalb von Gebäuden ist dies nicht immer gegeben. Konstante Störungen lassen sich durch eine geeignete Kalibrierung kompensieren, während dies bei willkürlichen Verfälschungen erheblich erschwert wird. Diese Effekte stellen eine Herausforderung an ein geeignetes System zur Fehlerkompensation dar [28] [29][A21].

### 2.3.5 Kamera (digital)

Digitale Kameras sind bildgebende Sensoren (CCD, CMOS), die Fotos sowie im kontinuierlichen Fall Videos der Umgebung aufnehmen und digitalisieren können. Mit Methoden der Bildverarbeitung lassen sich aus diesen Bildern wertvolle Informationen (Abstände, Positionen) extrahieren, womit diese Sensoren ähnlich den Augen bei einem Menschen zu den mächtigsten Sensoren gehören. Es lassen sich mit Hilfe von Kameras Bewegungen, Drehungen und Objekte erkennen sowie Abstände stereo-optisch bestimmen. Da es sich bei Bildern im Gegensatz zu Abstandssensoren, die nur einen Wert liefern, um ein ganzes Array an Daten handelt, das je nach Auflösung sehr umfangreich sein kann, brauchen diese Sensoren einen Rechner mit vergleichsweise viel Speicher und Rechenleistung [30].

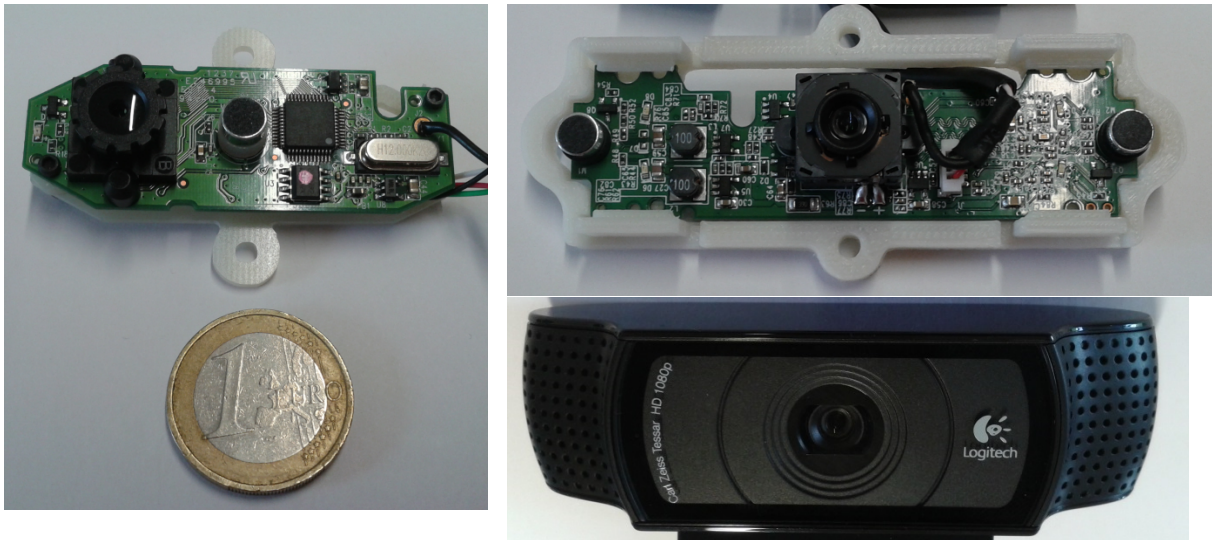


Abbildung 2.10: C270 ohne Gehäuse mit Halterung (links), C920 ohne Gehäuse mit Halterung bzw. mit Gehäuse (rechts)

Im AQopterI8 kommen die Kameras C270 und C920 von Logitech (Abbildung 2.10), sogenannte WebCams, zum Einsatz. WebCams sind vergleichsweise günstige Kameras, die primär zur Aufnahme von am PC sitzenden Personen während der Videoübertragung (z.B. über Skype) gedacht sind. Entsprechend sind die Kameras nicht optimiert zur Aufnahme beweglicher Bilder. Dies führt zu Problemen im Flug, da die Bilder durch die Bewegung „verwischen“. Dieser Effekt fällt bei der günstigeren C270 erheblich stärker ins Gewicht als bei der C920. Hauptvorteil von WebCams ist neben dem Preis die Tatsache, dass sie in der Regel sehr einfach per USB an einem PC anzuschließen und anzusteuern sind. Dies ist für Kameras, die z.B. per paralleler Schnittstelle an einen Mikrocontroller anzuschließen sind, nicht immer gegeben, zumal Mikrocontroller aufgrund des geringeren Speichers und der niedrigeren Rechenleistung die Möglichkeiten bei der weiteren Bildverarbeitung oft stark einschränken.

## Stereo-optische Abstandsmessung (SV, stereo vision)

Die stereo-optische Abstandsmessung nutzt zwei Kamerabilder derselben Szene, die aus unterschiedlichen Perspektiven aufgenommen worden sind. Mit Hilfe von Triangulation und bekannten konstanten Parametern bezüglich Kameraoptik und -abstand lässt sich so die Entfernung zu Objekten bestimmen. Die dazu benötigte Bildverarbeitung ist sehr aufwendig und entsprechend rechenintensiv. Damit überhaupt eine Berechnung über Triangulation möglich ist, müssen die korrespondierenden Bildbereiche erkannt und korrekt zugeordnet werden. Dies ist bei kontrastarmen Bildern und einfarbigen, glatten Oberflächen erheblich erschwert, enorm aufwändig oder unmöglich. Daher eignet sich die stereo-optische Abstandsmessung im Gegensatz zur Messung mit Hilfe von Abstandssensoren (IR, US) weniger für glatte, folglich oft eintönige Objekte, sondern viel besser für unebene bzw. chaotisch strukturierte Oberflächen, die kontrastreiche Bilder liefern. Deutlich wird dies in den beiden Fotos aus Abbildung 2.11. Die Abbildung zeigt links eine weiße Wand, die wegen der glatten Oberfläche gut mit Abstandssensoren zu messen ist, aber bei der ein SV-System Probleme hat und bestenfalls noch die Steckdosen und den Zettel zugeordnet bekommt, sofern diese im gemeinsamen Bildbereich liegen. Rechts dagegen befindet sich ein Beispiel für ein kontrastreiches Bild mit Regalen und Gegenständen, womit ein SV-System gut zurechtkommt und wo die genannten Abstandssensoren Probleme bekommen, wenn es keine ausreichend großen, glatten Oberflächen gibt, die gut reflektieren. Daher ergänzen sich SV-Sensorsysteme und Abstandssensoren sehr gut und jedes System vermag die Schwachstellen des anderen komplementär auszugleichen.



Abbildung 2.11: Links kontrastarme, einfarbige glatte Oberfläche (weiße Wand), rechts unstrukturierte, kontrastreiche Oberfläche (Schrank, Regal mit Werkzeugen, Kisten, Boxen und sonstigen Gegenständen)

Genauigkeit und Verlässlichkeit bzw. Tiefenauflösung und Öffnungswinkel (Messbreite) hängen unter anderem vom Kameraabstand ab (vgl. Formel 2.22). Je weiter die Kameras auseinander liegen, desto besser ist aufgrund trigonometrischer Zusammenhänge die Tiefenauflösung, d.h. die Messgenauigkeit entfernter Objekte. Gleichzeitig reduziert sich aber automatisch die Messbreite bzw. der Öffnungswinkel des SV-Sensorsystems, da der gemeinsame Bildbereich beider Kameras abnimmt. Da nur Abstände von Objekten, die im gemeinsamen Bildbereich liegen und von beiden Kameras erkannt werden, auch bestimmt werden können, kann der Abstand der Kameras nicht beliebig groß gewählt werden. Hinzu kommt, dass die Verlässlichkeit abnimmt, da in einem kleineren Bildbereich eine Zuordnung der Bilder erfolgen muss und dies unter Umständen aus bereits erwähnten Ursachen nicht möglich sein mag. Folglich ist die Wahl eines geeigneten Kameraabstandes ein Trade-Off Problem. Experimentell hat sich ein Kameraabstand als optimal herausgestellt, der in etwa dem Augenabstand beim Menschen entspricht.

## Optischer Flusssensor

Ein Spezialfall unter den Kameras sind die optischen Flusssensoren (Optical Flow Sensor, kurz OF). Diese Sensoren vereinen Kamera und bildverarbeitende Logik auf einer Platine und bestimmen basierend auf dem Prinzip des optischen Flusses Positionsänderungen (Abbildung 2.12). Somit lässt sich in einem Quadrocopter die Position relativ zur Startposition bestimmen. Käuflich erwerblich sind die beiden Modelle ADNS-3080 und PX4FLOW (Abbildung 2.13).



Abbildung 2.12: Prinzip optischer Fluss: Die korrespondierenden Bildpunkte sind nach links gerückt, was einer Kamerabewegung bzw. einem optischen Fluss nach rechts entspricht.

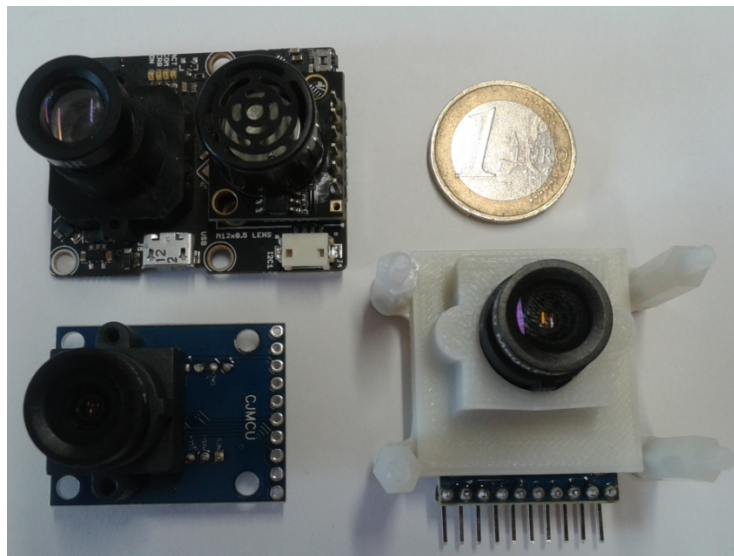


Abbildung 2.13: Optische Flusssensoren PX4FLOW (oben) und ADNS-3080 ohne/mit 3D gedrucktem Gehäuse

Der ADNS-3080 wird im AQuadrocopter eingesetzt und besteht aus einer Kamera und einem DSP (Digitaler Signalprozessor). Zu den Vorteilen dieses Sensors gehört die hohe Wiederholrate von bis zu 6400 fps (frames per second, Bilder pro Sekunde). Entsprechend liefert der Sensor selbst bei schnellen Bewegungen (und unter guten Bedingungen) qualitativ gute Ergebnisse. Der Sensor versagt bei schlechten Lichtverhältnissen, kontrastarmen Oberflächen oder Oberflächen mit dem Blenderproblem, d.h. z.B. symmetrische Oberflächen, bei denen eine Bildtranslation nicht korrekt einer Bewegung zuzuordnen ist. Ein Nachteil dieses Sensors ist die Beschränkung auf Translationen. Rotationen, insbesondere Drehungen um die z-Achse, führen zu Fehlern. Ein weiterer Nachteil des ADNS-3080 besteht in der Empfindlichkeit gegenüber Staub, der sich auf dem Lichtsensor festsetzt und zum Ausfall von Pixeln führt, worunter die Messergebnisse bis hin zum Totalausfall leiden. Daher ist es wichtig, diesen Sensor vor Staub und anderen feinen Partikeln zu schützen. Dies kann man z.B. durch ein 3D gedrucktes Gehäuse (Abbildung 2.13, rechts unten) erreichen [Tabelle 11.1 A04] [31].



### 2.3.6 PMD Kamera

Zur Aufnahme von Bildern mit Tiefeninformationen, sogenannten 3D Bildern (Abbildung 2.14), können PMD Kameras eingesetzt werden. PMD steht für Photonic Mixing Device (Photomischdetektor). PMD-Sensoren basieren auf dem Lichtlaufzeitverfahren. Dabei werden Lichtwellen in Richtung Messobjekt emittiert und im Empfänger (Receiver) absorbiert. Zusätzlich wird das gleiche Signal auf direktem Wege elektrisch als Vergleichsquelle dem Empfänger zugeführt (Abbildung 2.15). An Hand des Phasenunterschieds der beiden Signale kann nun auf die Laufzeit und den Abstand des Objekts geschlossen werden.

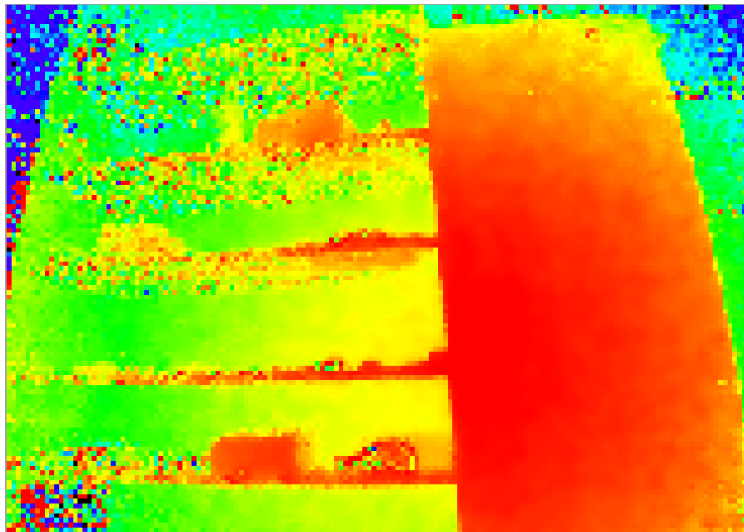


Abbildung 2.14: Tiefenbild (3D Bild) mit PMD Kamera aufgenommen

Im Vergleich zu SV-Systemen haben PMD Kameras bei gleicher Optik einen größeren effektiven Sichtbereich (FOV, field of view) und leiden nicht unter dem Problem, zwei Bildbereiche einander zuordnen zu müssen (Korrespondenz-Problem). Daher können PMD Kameras an einigen Stellen eingesetzt werden, wo SV-Systeme versagen. Darüber hinaus kann die Bildverarbeitung in SV-Systemen sehr rechenintensiv sein. Dies erhöht die Latenz und schränkt die Wiederholrate ein. Wird ein schnellerer, aber einfacherer Algorithmus verwendet, so geht dies oftmals auf Kosten der Zuverlässigkeit und kann je nach Oberfläche zu falschen Messungen führen. Diese Problematik gibt es bei PMD Kameras nicht. Ein Nachteil von PMD Kameras besteht jedoch darin, dass diese Licht emittieren und somit der Stromverbrauch deutlich höher ist. Zudem versagen PMD-Sensoren bei dunklen oder anderen Oberflächen, die Licht nicht ausreichend reflektieren [32].

Bis vor einigen Jahren waren PMD-Kameras sehr teuer, schwer und groß, was den Einsatz in Quadroptern erschwerte bis ausschloss. Die PMD-nano (Abbildung 2.16) ist ein kleines, leichtes und erschwingliches Modell, das im AQopterI8 zum Einsatz kommt. Der Sensor arbeitet mit einer Modulationsfrequenz von 20 MHz, woraus sich eine Eindeutigkeitsentfernung (NAR, non-ambiguity range) von 7.5m ergibt. Er hat eine Auflösung im Millimeterbereich und eignet sich besonders für nahe Messobjekte bis zu maximal 2m Entfernung. Abstände von über 5m sind aufgrund der abnehmenden Lichtintensität mit diesem Sensor kaum mehr korrekt zu detektieren. Um Störsignale wie Sonnenlicht oder sonstiges künstliches Licht zu unterdrücken, kommt der 4 Phasenalgorithmus zum Einsatz (vgl. Kapitel 2.6.5).

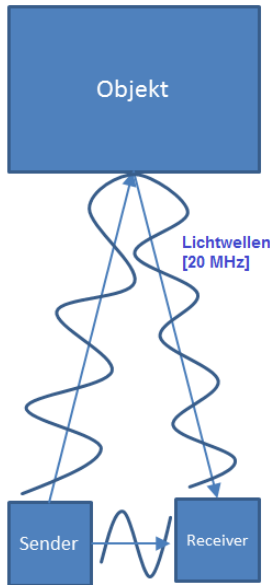


Abbildung 2.15: Prinzip der Lichtlaufzeitmessung, TOF (time of flight)

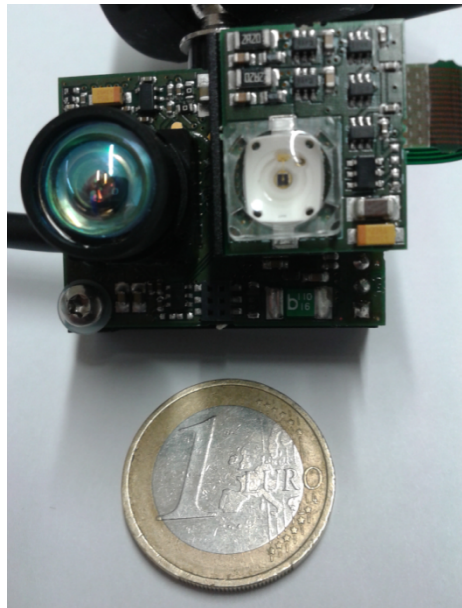


Abbildung 2.16: PMD-nano

### 2.3.7 Quellen

[12] [14] [15] [17] [18] [20] [21] [22] [24] [26] [27] [28] [29] [30] [A01-A30]

## 2.4 Orientierungsdarstellungen

Die Orientierungsbestimmung ist das Herz eines jeden Quadropters. Als instabiles System regelt der Quadropter permanent seine Orientierung. Alle Bewegungen im Flug hängen von der Orientierung ab. Fast jede Situation und jede Messung wird durch die Orientierung beeinflusst, womit die Orientierung an vielen Stellen einfließt. Ohne die aktuelle Orientierung ist der Quadropter nicht in der Lage, irgendeine sinnvolle Handlung auszuführen und versagt die Orientierungsbestimmung aufgrund von Fehlern (Software, Hardware), so muss der Quadropter fast unweigerlich abstürzen. Die Anforderungen sind im autonomen Flug noch höher, da es keinen menschlichen Operateur gibt, der kleine Fehler der Orientierung kompensiert. Schon ein Fehler von wenigen Grad kann dazu führen, dass die Drohne sich in eine ungewünschte Richtung bewegt.

Das Wissen um die Orientierung spielt neben der Anwendung in der Robotik und Drohnentechnik noch in vielen weiteren Anwendungsbereichen wie Navigation, Medizintechnik aber auch in Mobilfunkgeräten und Spielzeugen eine wichtige Rolle. Entsprechend umfangreich ist die Bandbreite existierender Darstellungsformen und Lösungsansätze. Die üblichsten Darstellungsformen sowie deren Vor- und Nachteile werden nun näher erläutert.

Euler-Winkel, insbesondere die Roll-Pitch-Yaw-Winkel (RPY), Quaternionen und Rotationsmatrizen sind die gebräuchlichsten Darstellungsformen von Orientierungen. RPY-Winkel sind eine sehr anschauliche Darstellungsform, die jede Orientierung über drei Drehungen um die drei Raumachsen XYZ beschreibt. Euler-Winkel leiden aber unter den beiden Nachteilen des hohen Rechenaufwandes und der singulären Stellen. Singuläre Stellen sind Unstetigkeitsstellen der Darstellung und entstehen üblicherweise durch Divisionen durch Null. In der numerischen Betrachtung am Computer muss dabei der Wert Null gar nicht erreicht werden, sondern es ist schon problematisch, wenn der Wert hinreichend klein wird. Singularitäten treten insbesondere bei einem Gimbal Lock auf. Gimbal Lock

meint den Fall, dass in bestimmten Orientierungen ein Freiheitsgrad verloren geht, womit auch ein Verlust an Eindeutigkeit verbunden ist. Man stelle sich ein Gimbal (kardanische Aufhängung) vor, bei dem alle Achsen zur Deckung kommen. Nun kann nicht mehr um alle Freiheitsgrade gedreht werden, da nun zwei Achsen um dieselbe Achse rotieren und entsprechend um eine Achse nicht gedreht werden kann. Die genannten beiden Probleme besitzen Quaternionen und Rotationsmatrizen nicht, was im Gegenzug auf Kosten der Anschaulichkeit geht.

Da jede Orientierung im dreidimensionalen Raum an Hand von drei Zahlen eindeutig definiert ist, haben Quaternionen ein redundantes Element und Rotationsmatrizen sechs. Dies macht das Arbeiten, insbesondere das Debugging, etwas aufwändiger und kostet (meist vernachlässigbar) mehr Speicher. Darüber hinaus sind Berechnungen mit Rotationsmatrizen etwas aufwändiger als mit Quaternionen. Aufgrund dieser Ursachen haben sich RPY-Winkel zur externen Darstellung für den Menschen und Quaternionen zur internen Darstellung und zur Verarbeitung weitestgehend durchgesetzt. Rotationsmatrizen eignen sich dabei oftmals als Zwischengröße bei der Umrechnung. Die Eigenschaften der Darstellungsformen sind nochmals in Tabelle 2.2 zusammengefasst [33].

**Tabelle 2.2: Vergleich von Orientierungsdarstellungen [33]**

Vergleich Orientierungsdarstellungen	Redundante Elemente	Singuläre Stellen	Anschaulichkeit	Performance
Rotationsmatrix	6	Nein	Gering	Gut
Euler-Winkel	0	Ja	Gut	Schlecht
Quaternionen	1	Nein	Gering	Sehr gut

Im Weiteren werden zunächst ein paar mathematische Grundlagen erläutert, bevor einige ausgewählte Lösungsansätze unter Unzähligen (vgl. Kapitel 11.9) zur Orientierungskompensation und Datenfusion zur Orientierungsbestimmung diskutiert werden.

### 2.4.1 Quaternionen und Euler-Winkel

Der folgende Abschnitt klärt ein paar Begriffe sowie notwendige mathematische Grundlagen, die bei der Implementierung zum Einsatz kommen. Dazu gehören Formeln zu Quaternionen und Euler-Winkeln.

Quaternionen, nach ihrem Erfinder auch Hamilton-Zahlen genannt, beschreiben einen eigenen Zahlenbereich, bei dem jede Zahl durch ein Quadrupel  $\underline{q}$  bestehend aus einem dreidimensionalen Vektorteil  $\vec{q}$  und einem Skalar  $q_0$  dargestellt wird (vgl. Formel 2.2):

$$\underline{q} = (q_0 \quad \vec{q}) = (q_0 \quad q_1 \quad q_2 \quad q_3) \quad \text{Formel 2.2}$$

Die genannte Schreibweise entspricht einem Hamilton Quaternion. Befindet sich der Skalarteil am Ende des Quaternions, so spricht man von einem Rodriguez Quaternion. Rodriguez ist der zweite Erfinder, dem die Quaternionen zugesprochen werden.

Mit Hilfe von Quaternionen lassen sich Orientierungen bzw. Rotationen beschreiben. Dazu wird mit einem Quaternion die aktuell gültige Orientierung des Systems oder auch eine Rotation dargestellt. Zur Veranschaulichung dient die Tatsache (vgl. Abbildung 2.17), dass der Vektorteil der Drehachse und der Skalarteil dem Drehwinkel  $\varphi$  entspricht.

Bei der Verwendung von Quaternionen sind ein paar spezielle Rechenvorschriften wie die nicht-kommutative Multiplikation zu beachten. Diese wird verwendet, um zwei Drehungen miteinander zu verrechnen. Dazu wird das aktuelle Orientierungsquaternion mit dem Rotationsquaternion, welches der Rotation entspricht, multipliziert. Weitere Drehungen können dabei beliebig folgen. Da jedoch auch Drehungen nicht kommutativ sind, gilt dies ebenso für die Quaternionenmultiplikation, d.h. die Reihenfolge der Multiplikation ist relevant. Soll der Vektor  $\vec{v}$  um den Winkel  $2 \cdot \varphi$  gedreht werden, so ist dieser Vektor mit dem Quaternion  $\underline{q}$  sowie seinem konjugiert Komplexen zu multiplizieren, wobei Formel 2.3 gilt:

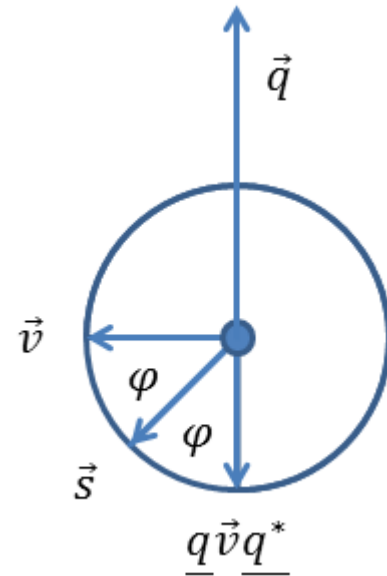


Abbildung 2.17: Quaternionen anschaulich

$$\underline{q} = (q_0 \quad \vec{q}) = (\cos(\varphi) \quad \sin(\varphi) * \vec{q}_n)$$

Formel 2.3

Während der Sinus (hinten) lediglich der Skalierung des Betrages auf Eins Rechnung trägt, stellt der Cosinus (vorne) die Länge der Drehung bzw. den korrekten Drehwinkel sicher. Entsprechend ist der Vektor  $\vec{s}$  durch Drehung um den halben Winkel  $\varphi$  zu erreichen. Es geht der halbe Winkel in die Formel ein, damit Drehungen von +/- 180° möglich sind. Andernfalls wären Drehung jenseits der +/- 90° aufgrund der Symmetrieeigenschaften der Sinus/Cosinus-Funktion nicht unterscheidbar. Allgemein gilt daher Formel 2.4 für den Fall einer Drehung um den Winkel  $\varphi$ :

$$\underline{q} = (q_0 \quad \vec{q}) = \left( \cos\left(\frac{\varphi}{2}\right) \quad \sin\left(\frac{\varphi}{2}\right) * \vec{q}_n \right)$$

Formel 2.4

Mit Hilfe der Quaternionenmultiplikation (Formel 2.5) wird aus zwei Quaternionen, einem Orientierungsquaternion  $\underline{r}$  und einem Rotationsquaternion  $\underline{d}$ , die aktuelle neue Orientierung  $\underline{q}$  ermittelt. Formel 2.5 fasst zwei alternative Darstellungsformen (Schreibweisen) der Quaternionenmultiplikation zusammen.

$$\underline{q} = \underline{r} \odot \underline{d} = \begin{pmatrix} r_0 & -r_1 & -r_2 & -r_3 \\ r_1 & r_0 & -r_3 & r_2 \\ r_2 & r_3 & r_0 & -r_1 \\ r_3 & -r_2 & r_1 & r_0 \end{pmatrix} \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} r_0 \cdot d_0 - \vec{r} \cdot \vec{d} \\ r_0 \cdot \vec{d} + d_0 \cdot \vec{r} + \vec{r} \times \vec{d} \end{pmatrix}$$

Formel 2.5

Es kommt vor, dass bestimmte Informationen, z.B. die Winkelgeschwindigkeitsmessungen der Gyrometer, nicht als Quaternion vorliegen, sondern als Skalar der drei Raumachsen bzw. als

dreidimensionaler Vektor. Dann ist es erforderlich, diese infinitesimalen Winkeländerungen bzw. Winkel in ein Quaternion zu überführen oder sonst wie mit dem Orientierungsquaternion verrechnen zu können. Für Ersteres, d.h. zur Umrechnung in ein Quaternion, können die infinitesimalen Winkeländerungen der Sensoren als Euler-Winkel interpretiert und dann in ein Quaternion umgerechnet werden. Formel 2.6 beschreibt die Umrechnung der Eulerwinkel

$$\overrightarrow{RPY} = \begin{pmatrix} roll \\ pitch \\ yaw \end{pmatrix}$$

in ein Quaternion  $\underline{q}$  [34]:

$$\underline{q} = \begin{pmatrix} \cos(yaw) \cdot \cos(pitch) \cdot \cos(roll) + \sin(yaw) \cdot \sin(pitch) \cdot \sin(roll) \\ \cos(yaw) \cdot \cos(pitch) \cdot \sin(roll) - \sin(yaw) \cdot \sin(pitch) \cdot \cos(roll) \\ \cos(yaw) \cdot \sin(pitch) \cdot \cos(roll) + \sin(yaw) \cdot \cos(pitch) \cdot \sin(roll) \\ \sin(yaw) \cdot \cos(pitch) \cdot \cos(roll) - \cos(yaw) \cdot \sin(pitch) \cdot \sin(roll) \end{pmatrix} \quad \begin{array}{l} \text{Formel} \\ 2.6 \end{array}$$

Nach Normierung erfolgt anschließend die Fusion mittels Quaternionenmultiplikation.

Eine zweite, alternative Möglichkeit bietet der Ansatz mit Hilfe eines Änderungs- oder Ableitungsquaternion beruhend auf einer Approximation für kleine Winkel. Dies entspricht einer Taylorreihenentwicklung 1. Ordnung. Für das Änderungsquaternion  $\underline{d}$  gilt dann Formel 2.7, wobei  $\underline{q}$  das aktuelle Quaternion und  $x,y,z$  die infinitesimalen Winkeländerungen der drei Raumachsen in Radiant darstellen.

$$\underline{d} = \begin{pmatrix} -q_1 \cdot x - q_2 \cdot y - q_3 \cdot z \\ q_0 \cdot x + q_2 \cdot z - q_3 \cdot y \\ q_0 \cdot y - q_1 \cdot z + q_3 \cdot x \\ q_0 \cdot z + q_1 \cdot y - q_2 \cdot x \end{pmatrix} \quad \text{Formel 2.7}$$

Nun wird das neue Orientierungsquaternion  $\underline{n}$  durch Addition der beiden Quaternionen bestimmt (Formel 2.8). Anschließend ist das Quaternion  $\underline{n}$  zu normieren.

$$\underline{n} = \underline{q} + \underline{d} \quad \text{Formel 2.8}$$

Im Allgemeinen ist es nicht sinnvoll, zwei Orientierungsquaternionen zu addieren. Schon allein deshalb ist diese Vorgehensweise auf kleine Winkel beschränkt. Die erstgenannte Methode versagt ebenfalls bei größeren Winkeln, da dann die gemessenen körperfesten Winkeländerungen nicht mehr exakt den mitdrehenden Euler-Winkeln entsprechen. Abgesehen von dieser Tatsache kommt die Berechnung über Euler-Winkel ohne Approximation aus, weshalb diese Methode zur Drehung eines Quaternions verwendet wird. Beide Methoden wurden implementiert und es konnte bei einer Samplezeit von 5-10ms kein signifikanter Unterschied festgestellt werden. Die zweite Methode kommt ohne trigonometrische Funktionen aus und reduziert somit den Rechenaufwand.

Formel 2.9 beschreibt die Umkehrfunktion von Formel 2.6, d.h. die Bestimmung der Euler-Winkel  $\overrightarrow{RPY}$  aus dem Quaternion  $\underline{q}$ . Diese Methode hat sich zur Bestimmung der Euler-Winkel bewährt und

kommt daher in der Implementierung zum Einsatz.

$$\begin{aligned} \overrightarrow{RPY} &= \begin{pmatrix} roll \\ pitch \\ yaw \end{pmatrix} \\ &= \begin{pmatrix} atan2(2 \cdot (q_0 \cdot q_1 + q_2 \cdot q_3), 1 - 2 \cdot (q_1 \cdot q_1 + q_2 \cdot q_2)) \\ asin(2 \cdot (q_0 \cdot q_2 - q_1 \cdot q_3)) \\ atan2(2 \cdot (q_0 \cdot q_3 + q_2 \cdot q_1), 1 - 2 \cdot (q_3 \cdot q_3 + q_2 \cdot q_2)) \end{pmatrix} \end{aligned} \quad \begin{array}{l} \text{Formel} \\ 2.9 \end{array}$$

Eine alternative Berechnung, die gänzlich auf den Arkussinus verzichtet, liefert Formel 2.10.

$$\begin{aligned} \overrightarrow{RPY} &= \begin{pmatrix} roll \\ pitch \\ yaw \end{pmatrix} \\ &= \begin{pmatrix} atan2(2 \cdot (q_0 \cdot q_1 + q_2 \cdot q_3), 1 - 2 \cdot (q_1 \cdot q_1 + q_2 \cdot q_2)) \\ atan2\left(\frac{2 \cdot (q_0 \cdot q_2 - q_1 \cdot q_3)}{\sqrt{(2 \cdot (q_0^2 + q_1^2) - 1)^2 + (2 \cdot (q_0 \cdot q_3 + q_2 \cdot q_1))^2}}\right) \\ atan2(2 \cdot (q_0 \cdot q_3 + q_2 \cdot q_1), 1 - 2 \cdot (q_3 \cdot q_3 + q_2 \cdot q_2)) \end{pmatrix} \end{aligned} \quad \begin{array}{l} \text{Formel} \\ 2.10 \end{array}$$

Zwei Vektoren definieren eine Drehung, die zur Orientierungskompensation herangezogen werden kann. Entsprechend lässt sich mit Hilfe von Kreuz- und Skalarprodukt aus den beiden Vektoren  $\vec{v}_1$  und  $\vec{v}_2$  ein Rotationsquaternion  $\underline{q}$  bestimmen (Formel 2.11 - Formel 2.13).

$$\vec{c} = \vec{v}_1 \times \vec{v}_2 = (c_1 \quad c_2 \quad c_3) \quad \text{Formel 2.11}$$

$$s = \vec{v}_1 \cdot \vec{v}_2 \quad \text{Formel 2.12}$$

$$\underline{q} = \begin{pmatrix} \cos\left(\frac{s}{2}\right) \\ c_1 \cdot \sin\left(\frac{s}{2}\right) \\ c_2 \cdot \sin\left(\frac{s}{2}\right) \\ c_3 \cdot \sin\left(\frac{s}{2}\right) \end{pmatrix} \quad \text{Formel 2.13}$$

## 2.4.2 Lösungen zur Orientierungsdatenfusion (Orientierungskompensation)

Das Problem der Orientierungsbestimmung bzw. der Datenfusion von Orientierungsinformationen findet sich in sehr vielen Anwendungsfällen und entsprechend umfangreich ist die wissenschaftliche Literatur zu dem Thema (vgl. Kapitel 11.8). In den hier beschriebenen Fällen werden Daten von MEMS-Sensoren, also Beschleunigungs- und Drehratensensoren, miteinander fusioniert. In einigen Fällen kommen auch Magnetsensoren hinzu. Darüber hinaus gibt es noch andere Messmethoden,

z.B. die optische Messung von Rotationen bzw. der Orientierung im Raum durch optisches Tracking oder Bildverarbeitung. Alle diese Daten lassen sich miteinander fusionieren.

Das zugrundeliegende Problem ist in allen Fällen, dass Gyroskope kurzfristig sehr gute Daten liefern (Hochpassfilter-Verhalten), aber sich langfristig ein Driftfehler akkumuliert. Dieses Problem ist systeminhärent und liegt an der Tatsache, dass Gyroskope Winkelgeschwindigkeiten messen. Zur Ermittlung der Orientierung ist die Winkelgeschwindigkeit zu integrieren. Aufgrund von Temperaturschwankungen, Vibrationen und anderen Störquellen entsteht ein Fehler, der sich fortpflanzt. Da die Sensoren nur relative Bewegungen detektieren, kann ein Fehler, der einmal im System ist, nicht mehr kompensiert werden. Deshalb werden Gyroskope mit weiteren Sensoren wie Accelerometer und Magnetsensoren fusioniert. Diese Fusion leistet die Orientierungskompensation und nennt sich Orientierungsdatenfusion. Vorteil der Accelerometer und Magnetsensoren ist die Verwendung einer absoluten Referenz wie des Erdgravitationsfeldes und des Erdmagnetfeldes zur Bestimmung der Orientierung. Die Messung der Orientierung auf Basis dieser Messgrößen ist im Allgemeinen driftfrei. Jedoch unterliegen diese Sensoren weiteren Störungen und sind schon aufgrund der fehlenden Integration verrauschter als die Winkel des Gyroskops. Entsprechend ist es erforderlich, die Daten gemittelt über einen längeren Zeitraum eingehen zu lassen (Tiefpass-Verhalten), um einen Driftfehler zu kompensieren.

### **EKF (Extended Kalman Filter):**

Der Erweiterte Kalman-Filter ist eine Erweiterung des Kalman Filters auf nicht-lineare Systeme. Die Orientierungsbestimmung ist ein solches nicht-lineares System. Beim EKF wird das System durch eine Taylorreihenentwicklung linearisiert, wobei es unterschiedliche Ansätze bei der Modellierung gibt. Eine Implementierung des EKFs von Sabatini [35] wurde von Lukas Iffländer für einen Quadrocopter durchgeführt (vgl. Tabelle 11.1 A06). Aufgrund des hohen Rechenaufwandes und der Tatsache, dass das Filterkonzept nicht zur Anwendung im Flug ausgelegt ist, eignet sich dieser weniger für den Einsatz im AQuoter18. Darüber hinaus gibt es diverse weitere Implementierungen des EKFs zur Orientierungsbestimmung [36] [37] [38].

Verglichen mit einfacheren Lösungen liegen die Vorteile beim EKF darin, dass durch das EKF nicht-lineare Effekte besser modelliert werden können. Als Nachteil sind die erhöhte Komplexität und die damit verbundene Fehleranfälligkeit, der Rechenaufwand sowie Aufwand zur Modellierung und Parametrisierung zu nennen.

### **UKF (Unscented Kalman Filter):**

Eine Alternative zur Anwendung des Kalman Filters auf nicht-lineare Systeme bietet der UKF. Im Vergleich zum EKF besteht der wesentliche Unterschied in der Art der Verarbeitung der Zufallsvariablen. Beim EKF werden die Zufallsvariablen als gaußverteilt angenommen und mit Hilfe einer Linearisierung des Systems geschätzt. Beim UKF werden die Zufallsvariablen durch eine Menge signifikanter Punkte, sogenannter Sigmapunkte, modelliert.

Die Schwierigkeiten des UKFs liegen in der Wahl der Filterparameter sowie der Modellierung des Systems, womit der UKF im Vergleich zum EKF nochmals eine erhöhte Komplexität mit sich bringt. Eine Implementierung dieses Filters zur Orientierungsbestimmung findet sich bei Klingebeil [39].

### **Weitere Ansätze zur Orientierungskompensation:**

Zu den weiteren Lösungsansätzen gehören der komplementäre Kalman Filter (engl.: CKF,

Complementary Kalman Filter, [40]), der MKF (Multisensor Kalman Filter, [41]) der MMKF (Multi Modell Kalman Filter, [42]) sowie der Mahony Filter [43], um nur einige zu nennen.

### 2.4.3 Zusammenfassung: Wahl der Orientierungskompensation

Bei der Wahl und Implementierung des Systems zur Orientierungskompensation ist abzuwägen zwischen Komplexität der Modellierung, Implementierungs- und Parametrisierungsaufwand, Erweiterbarkeit, Rechenlast und Genauigkeit. Die Genauigkeit, der Aufwand und die Rechenlast steigen im Allgemeinen mit der Komplexität der Ansätze. Da die Orientierungskompensation bei einem Quadropter auf einem Mikrocontroller mit wenig Rechenkapazität und gemeinsam mit anderen Funktionen mit möglichst hoher Wiederholrate laufen soll, verbieten sich meist komplexe Ansätze wie der EKF und der UKF.

Da jedoch gleichzeitig die Orientierung das Kernstück eines jeden Quadropters darstellt und eine korrekte Orientierung gerade für eine autonome Drohne von besonderer Bedeutung ist, bieten sich basierend auf dem Stand der Technik und den mathematischen Grundlagen Untersuchungen zur Orientierungskompensation an. Dazu sind ein eigenes Konzept und eine eigene Implementierung zur Orientierungskompensation notwendig. Eine vollfunktionsfähige Lösung aus bestehenden fremden Systemen zu verwenden, erwies sich daher nicht als sonderlich praktikables Vorgehen. Eine Blackbox-Lösung eines derart substantiellen Aspekts bürgt zudem den Nachteil der Limitation, des fehlenden Eingriffs sowie des Mangels an Wissen. Die gefundene Lösung ist zudem hardwareunabhängig.

### 2.4.4 Quellen

[33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43]

## 2.5 Regelungstechnik für Quadropter

In diesem Abschnitt soll kurz auf ein paar regelungstechnische Hintergründe und Grundlagen dieser Arbeit eingegangen werden. Dabei ist zu beachten, dass sowohl das Fach Regelungstechnik als auch eine umfassende regelungstechnische Betrachtung von Quadroptern überaus umfangreich sind, welche als ein Teilgebiet dieser Arbeit behandelt werden, aber nicht im Fokus dieser liegen. Folglich existiert bei der Wahl des Reglers und dessen Auslegung eine Vielzahl an Alternativen. Neben dem klassischen PID-Regler, der bei der Regelung von Quadroptern weit verbreitet ist, existieren Ansätze basierend auf der Zustandsregelung, der H-unendlich-Regelung (H-Infinity) und dem Fuzzy-Regler. Darüber hinaus existieren auch viele Mischformen und Modifikationen der genannten Methoden.

Die für das weitere Verständnis notwendige Basis der Regelungstechnik für Quadropter soll in diesem Abschnitt geschaffen werden, wobei für eine detailliertere Betrachtung der Thematik auf einschlägige Literatur verwiesen wird (vgl. Kapitel 2.5.4 und 11.9).

### 2.5.1 Instabiles System

Der Quadropter gehört zu den sogenannten instabilen Systemen. Wird der Quadropter nicht permanent geregelt, so stürzt er unweigerlich ab. Eine einfache Steuerung der Lage eines Quadropters über die Motoren ohne Feedback über eine Sensorik, z.B. indem an allen Motoren der gleiche Stellwert eingestellt wird, ist nicht praktikabel. Dieser Umstand ist der Tatsache geschuldet, dass auch bei möglichst optimaler, d.h. symmetrischer Auslegung aller Komponenten wie Motoren, Propeller und Mechanik (Schwerpunktverteilung) eine vollständige Balance dennoch



nie erreicht wird. Im Gegensatz zur Steuerung sorgt die Regelung permanent dafür, dass eine Abweichung vom Sollwert durch eine Änderung der Motorstellwerte ausgeglichen wird.

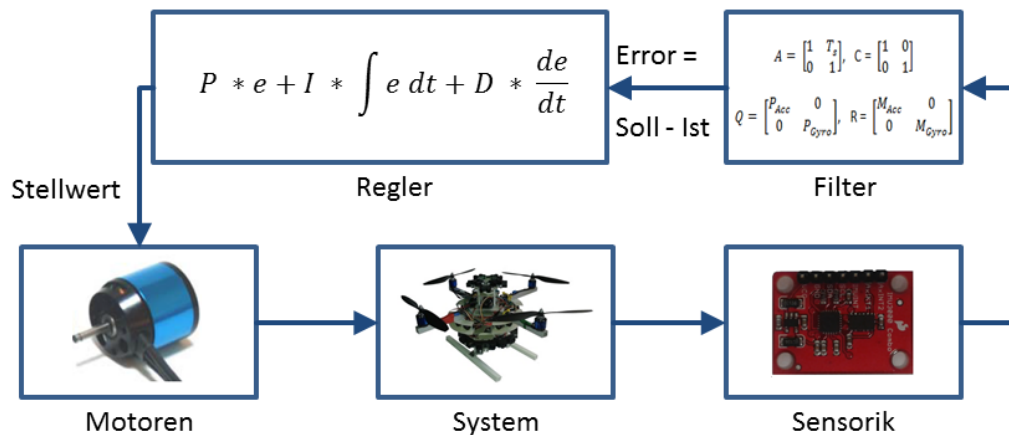


Abbildung 2.18: Regelung eines Quadropters

Abbildung 2.18 veranschaulicht das vereinfachte Prinzip der Regelung eines Quadropters. Mit Hilfe der Sensorik und eines Filters bzw. durch Datenfusion wird der Ist-Wert, die aktuelle Lage, bestimmt. Im Falle eines PID-Reglers wird dann der Fehler  $e$  als Differenz zwischen Soll- und Ist-Wert ermittelt und geht als Eingang in den Regler ein. Der Ausgang des Reglers bestimmt den Stellwert des Motors, der wiederum das System manipuliert. Diese Manipulation wird wiederum von den Sensoren gemessen und somit über den Regler zurückgekoppelt, auch Feedback genannt. Mit Hilfe des Reglers ist es zudem möglich, die gewünschte Lage beliebig zu steuern. Dies ist eine Voraussetzung für den autonomen Flug.

### 2.5.2 Mehrgrößensystem und Superposition

Das Regelungssystem des Quadropters gehört zu den Klassen der Mehrgrößensysteme, auch MIMO (Multiple Input Multiple Output) genannt. Die je 3 translatorischen und 3 rotatorischen Freiheitsgrade (6 DOF, degree of freedom), d.h. die Lage des Quadropters im Raum in Bezug auf Position XYZ und Orientierung RPY, sublimieren sich zu 6 Freiheitsgraden, die durch die vier Motoren manipuliert werden.

Der Tatsache, dass das System 4 Aktuatoren und 6 Freiheitsgrade besitzt, ist geschuldet, dass nicht alle Freiheitsgrade unabhängig voneinander manipuliert werden können. Diese Abhängigkeit und die Tatsache, dass es sich um ein MIMO-System handelt, erhöhen die Komplexität bei einer ganzheitlichen Betrachtung des Regelungsproblems verglichen mit einfachen SISO-Systemen (Single Input Single Output) erheblich. Dazu kommt, dass das Gesamtsystem, da bereits die Orientierungsdarstellung nicht-linear ist, ebenfalls nicht-linear ist. Demzufolge existieren keine Matrizen  $\mathbf{M}$  und  $\mathbf{N}$ , für die Formel 2.14 (vereinfachte Zustandsgleichung) gilt.

Um dieses Problem zu lösen, gibt es im Grunde nur das Mittel der Vereinfachung. Um dennoch eine analytische Lösung zu erhalten, kann das System durch eine Taylorreihenentwicklung linearisiert werden. Ein anderes, einfacheres und weit verbreitetes Verfahren ist die Superposition. Das Problem der 6 Freiheitsgrade wird heruntergebrochen auf 6 Probleme mit einem Freiheitsgrad. Aus dem MIMO 4I6O (4 Inputs, 6 Outputs) werden 6 überlagerte SISO-Systeme (Kaskadierung).

$$\begin{bmatrix} X(t+1) \\ Y(t+1) \\ Z(t+1) \\ R(t+1) \\ P(t+1) \\ Y(t+1) \end{bmatrix} = \mathbf{M} \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \\ R(t) \\ P(t) \\ Y(t) \end{bmatrix} + \mathbf{N} \quad \text{Formel 2.14}$$

Zunächst aber lässt sich statt der unlösbaren Formel 2.14 durch Trennung von Translation und Rotation und ohne Einschränkung der Allgemeinheit Formel 2.15 für die Translation bzw. die Formel 2.16 für die Rotation aufstellen.

$$\begin{bmatrix} X(t+1) \\ Y(t+1) \\ Z(t+1) \end{bmatrix} = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} + \mathbf{R}(t) * \begin{bmatrix} \Delta X(t) \\ \Delta Y(t) \\ \Delta Z(t) \end{bmatrix} \quad \text{Formel 2.15: Translation}$$

$$\mathbf{R}(R(t+1), P(t+1), Y(t+1)) = \mathbf{R}(t+1) = \mathbf{R}(t) * \Delta \mathbf{R}(t) \quad \text{Formel 2.16: Rotation}$$

Das Verfahren der Superposition lässt sich fortsetzen, bis letztendlich die 6 SISO-Systeme (vgl. Formel 2.17) scheinbar unabhängig übrig bleiben:

$$\begin{bmatrix} X(t+1) \\ Y(t+1) \\ Z(t+1) \\ R(t+1) \\ P(t+1) \\ Y(t+1) \end{bmatrix} = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \\ R(t) \\ P(t) \\ Y(t) \end{bmatrix} + \begin{bmatrix} \Delta X(t) \\ \Delta Y(t) \\ \Delta Z(t) \\ \Delta R(t) \\ \Delta P(t) \\ \Delta Y(t) \end{bmatrix} \quad \text{Formel 2.17: 6 SISO-Systeme}$$

Diese Darstellung ist die einfachste und zugleich die ungenaueste Modellierung des Systems. Dennoch ist es möglich, die Regelung eines Quadropters damit auszulegen. Bei solch einem System würde der Fehler nach kurzer Zeit jedoch derart ansteigen, dass der Quadropterschief in der Luft steht. Die Gleichung dient daher mehr dem Verständnis, als dass eine Implementierung sinnvoll erscheint. Sie veranschaulicht aber sehr gut die Grundidee der Superposition.

Dieselbe Unterteilung des Regelungsproblems in 6 Teilprobleme lässt sich auch bei Betrachtung der Dynamik des Systems herleiten. Aufgrund des symmetrischen Aufbaus des Quadropters sowie der damit verbundenen gleichmäßigen Verteilung der Aktuatoren sowie derer wirkender Kräfte lassen sich einige Vereinfachungen machen. Die Kippachsen, d.h. die x- und y- Achse (Roll, Pitch), werden im Wesentlichen durch die beiden entlang der jeweiligen Achse am Ende der beiden Ausleger angebrachten Motoren manipuliert. Dreht der rechte Motor schneller als der linke Motor, so „rollt“ sich der Quadropters nach links (vgl. Abbildung 2.19 links). Die z-Achse (Gierachse) wird durch alle vier Motoren gemeinsam manipuliert, indem die Motoren der x-Achse linksdrehend und die Motoren der y-Achse rechtsdrehend sind. Drehen somit die beiden Motoren einer Achse stärker als die beiden Motoren der anderen Achse, so lässt sich damit der Quadropters nach dem Satz der Drehimpulserhaltung um die z-Achse gieren (vgl. vgl. Abbildung 2.19 rechts).

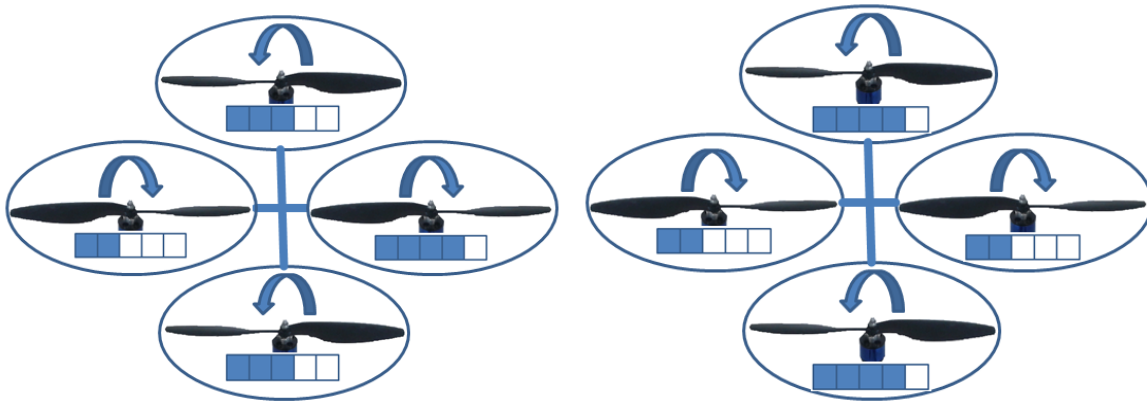


Abbildung 2.19: Prinzip des Rollens (links) und des Gierens (rechts)

Damit wurden aus dem Problem der dreidimensionalen Lageregelung (vgl. Formel 2.16) drei eindimensionale Regelungsprobleme. Ebenso lässt sich bei der Position verfahren. Die Höhenregelung lässt sich realisieren, indem die Summe aller vier Motoren geregelt wird. Mehr Gas in der Summe lässt den Quadropters steigen, weniger lässt ihn aufgrund der Schwerkraft sinken. Die Position in x- und y-Richtung lässt sich regeln, indem der Quadropters sich neigt. Ein Rollen um die x-Achse ermöglicht eine Positionsänderung in der y-Achse und umgekehrt. Somit lässt sich das sechsdimensionale Regelungsproblem herunterbrechen auf ein Regelungsproblem von 6 Reglern in einer Dimension.

Ausgehend von diesen Überlegungen lässt sich ein Regler auslegen, wobei zur Bestimmung der Zustände Formel 2.15 und Formel 2.16 herangezogen werden.

### 2.5.3 Reglerwahl und Auslegung

Übliche Regler für Quadropters sind PID-Regler und Zustandsregler (LQ-Regler, linear-quadratischer Regler), wobei der PID-Regler sehr verbreitet ist. Der PID-Regler ist ein bewährter Regler, der in vielen Anwendungsbereichen zum Einsatz kommt. Entsprechend umfangreich ist die verfügbare Literatur [44] [45] [46]. Dabei steht PID für Proportional, Integral und Differential, womit die einzelnen Bestandteile des Reglers beschrieben werden. Der Proportionalteil greift proportional zum Fehler in den Stellwert ein. Der Integralteil integriert den Fehler und greift proportional zum Integral in den Stellwert ein, während der Differentialteil den Fehler ableitet und proportional zur Änderung des Fehlers (Ableitung) in das System eingreift.

Mit etwas Erfahrung kann der Regler relativ gut empirisch parametrisiert werden. Die Schwierigkeit bei der Auslegung des Reglers ist nicht die Implementierung des Regelalgorithmus, sondern dessen Parametrisierung. Ein falsch parametrisierter Regler führt zum Absturz des Systems, weshalb ein Trial-And-Error Ansatz nicht einfach möglich ist. Für den Fall des Quadropters bietet sich der eigens dazu entwickelte 3DOF QCS (engl.: quadrotor control system, QCS) an, mit dem der Regler eingestellt werden kann (siehe S. 63). Mit dem QCS kann der Lageregler parametrisiert und das System realistisch getestet werden.

Neben dem empirischen Tunen der PID-Parameter gibt es noch weitere Ansätze wie z.B. das mathematische und physikalische Modellieren des Systems, um damit die optimalen Reglerparameter unter Eingabe des gewünschten Regelverhaltens zu errechnen. Üblicherweise geschieht dies unter Zuhilfenahme von Matlab. In der lesenswerten Masterarbeit von Lebedev wurde

dies mit bescheidenem Erfolg probiert (vgl. Tabelle 11.1 A15). Größtes Problem bei diesem Vorgehen ist neben der bereits diskutierten mathematischen Komplexität (nicht-lineares MIMO 4I6O System) die Modellierung und Parametrisierung. Dies gilt insbesondere, wenn keine langjährigen persönlichen Erfahrungswerte vorliegen. Dann ist es schwierig, zu beurteilen, welche Effekte dominant und welche vernachlässigbar sind. Der empirisch eingestellte Regler hat sich hier als das bessere Vorgehen erwiesen.

Ein System wie ein Quadropter, bei dem der Winkel durch einen PID-Regler geregelt wird, ist der Regelung mit Hilfe eines Zustandsreglers (Zustandsrückführung) und den Zuständen Winkel, Winkelgeschwindigkeit und Winkelfehlersumme sehr ähnlich.

## 2.5.4 Quellen

[9] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57]

# 2.6 Algorithmen und Methoden

## 2.6.1 Optischer Fluss

Unter optischem Fluss wird ein Verschiebungs- bzw. Geschwindigkeitsvektor oder auch Vektorfeld verstanden, das jeweils der Transformation zwei aufeinander folgender Bilder entspricht. Somit ist der optische Fluss ein Verfahren der Bildverarbeitung, das der Bestimmung von Bewegungen aus Bildsequenzen dient. Üblicherweise werden Schwarz-Weiß bzw. Graustufenbilder herangezogen, wodurch die Bewegung von Helligkeitsmustern untersucht wird. Die Methoden zum optischen Fluss gliedern sich wiederum in drei Kategorien: Differenzielle Methoden, Frequenzbasierte Methoden und Korrelationsmethoden. Eine Differenzielle Methode sowie das Fourier Tracking, eine frequenzbasierte Methode unter Verwendung der Fourier Transformation (FT), finden in dieser Arbeit Anwendung und werden daher näher erläutert. Die Korrelationsmethoden versuchen durch eine Vergleichssuche basierend auf einem Ähnlichkeitsmaß die Transformation höchster Übereinstimmung zu ermitteln, weshalb sie in der Regel sehr rechenintensiv sind [A04] [58] [59] [60].

### Differenzielle Methode

Die differentiellen Methoden untersuchen Helligkeitsänderungen bzw. das Helligkeitsdifferential der Bilder, um eine Aussage zur Transformation aufeinanderfolgender Bilder zu treffen. Mathematisch basieren die Methoden auf der Kontinuitätsgleichung des optischen Flusses (Formel 2.18).

$$\frac{\partial H}{\partial t} + \frac{\partial H}{\partial x} \frac{dx}{dt} + \frac{\partial H}{\partial y} \frac{dy}{dt} = 0$$

**Formel 2.18: Kontinuitätsgleichung  
des optischen Flusses  
H steht für Helligkeit**

Die Methode nach Lucas und Kanade ist ein verbreiteter Ansatz zur Bestimmung des optischen Flusses [61]. Eine Realisierung liefert der Algorithmus von Srinivasan [62]. Hierbei wird zur Vereinfachung angenommen, dass die Verschiebung im Bild maximal 1 Pixel beträgt und einer Translation in der z-Ebene entspricht. Unter diesen Nebenbedingungen kann nun mit Formel 2.19 der optische Fluss  $u$  und  $v$  in  $x$ - und  $y$ -Richtung bestimmt werden.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_x \sum_y (P_t - P_{t-1}) * (P_2 - P_1) \\ \sum_x \sum_y (P_t - P_{t-1}) * (P_4 - P_3) \end{bmatrix}$$

Formel  
2.19

$$* \begin{bmatrix} \sum_x \sum_y (P_2 - P_1)^2 & \sum_x \sum_y (P_2 - P_1) * (P_4 - P_3) \\ \sum_x \sum_y (P_2 - P_1) * (P_4 - P_3) & \sum_x \sum_y (P_4 - P_3)^2 \end{bmatrix}^{-1}$$

$P_x(i,j)$ ,  $P_y(i,j)$  und  $P_t(i,j)$  sind die partiellen Helligkeitsdifferenziale nach der Richtung x und y bzw. der Zeit t. Diese lassen sich aus einem Bild (vgl. Abbildung 2.20) nach Formel 2.19 ermitteln, wobei die Indizes als Zahl den Nachbarpixeln entsprechen.

$$\begin{aligned} P_x(i,j) &= P_2 - P_1 \\ P_y(i,j) &= P_4 - P_3 \\ P_t(i,j) &= P_t - P_{t-1} \end{aligned}$$

Formel  
2.20

In Abbildung 2.20 entspricht das gelb umrandete Pixel  $P_t$  bzw.  $P_{t-1}$  und die rot umrandeten Pixel sind die dazu benachbarten Pixel  $P_1, P_2, P_3$  und  $P_4$  [P4][P5][A04].

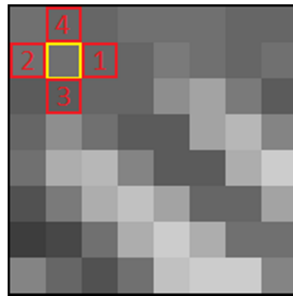


Abbildung 2.20: Prinzip der differentiellen Methode nach Srinivasan [A04]

### Fourier Tracking (FT)

Die FT-basierte Bildregistrierung, auch Fourier Tracking genannt, ist ein frequenzbasiertes Verfahren zur Bestimmung des optischen Flusses zweier Bilder  $H_1$  and  $H_2$ . Dabei werden die Bilder mit Hilfe der Fourier Transformation in den Frequenz- bzw. komplexen Zahlenbereich überführt, wodurch die Bilder sich wie mathematische Funktionen bzw. Signale verarbeiten lassen. Dazu wird das Bild bestehend aus Helligkeitsintensitäten in zwei Dimensionen als Messung eines zweidimensionalen Signals bzw. einer zweidimensionalen Funktion interpretiert. Somit wird die Faltung zweier Bilder im Frequenzbereich, Phasenkorrelation genannt, möglich und es lässt sich somit das Cross-Power-

Spektrum (CPS, Formel 2.21) bestimmen [63].

$$CPS = \frac{H_1[s_1, s_2] * H_2[s_1, s_2]}{|H_1[s_1, s_2]H_2[s_1, s_2]|} \quad \text{Formel 2.21}$$

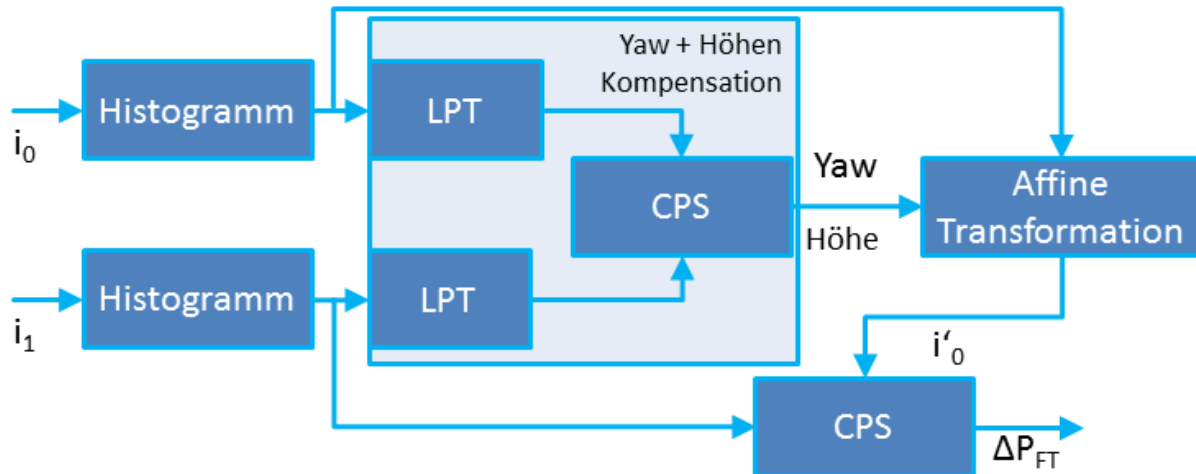


Abbildung 2.21: Fourier Tracking Algorithmus [A23]

Das Ergebnis der Faltung bzw. der Punkt des Maximums des CPS entspricht der Translation beider Bilder. Dies gilt jedoch ebenso wie bei der erwähnten differentiellen Methode nur, wenn die Bilder sich lediglich durch eine Translation in der z-Ebene unterscheiden. Daher sind die Bilder vorher mit Hilfe des Wissens um die Höhe und Drehung entsprechend zu transformieren. Diese Informationen können durch andere Sensoren bereitgestellt werden. Eine andere Möglichkeit bietet die vorherige Bestimmung von Drehung und Skalierung mit Hilfe der Fourier Transformation [64]. Es hat sich gezeigt, dass dies zu besseren Ergebnissen führt. Eine Lösung zum Fourier Tracking basierend auf der Library OpenCV wurde von Eric Reinthal implementiert und in das AQopterI8-Projekt integriert (Abbildung 2.21). Das System erlaubt die iterative Positionserfassung mit Hilfe der auf den Boden gerichteten Kameras und operiert auch unter widrigen Lichtverhältnissen. Auf dem verwendeten Prozessor (LP-180) erreicht es eine Samplezeit von 100ms. Somit kann der Quadrocopter um die z-Achse rotieren und weiterhin seine Position mit Hilfe des optischen Flusses erfassen [A23][P8].

In Abbildung 2.21 ist der Algorithmus zur relativen Positionsbestimmung mittels Fourier Tracking vereinfacht zusammengefasst. Um Robustheit gegenüber Lichtverhältnisänderungen zu erlangen, werden die Bilder zunächst mit Hilfe eines Histogramm-Ausgleichs vorverarbeitet. Anschließend wird basierend auf der Log-Polar-Transformation der vorab Fourier-transformierten Bilder und dem durch Phasenkorrelation bestimmten CPS die Drehung um die z-Achse (yaw) sowie die Höhenänderung ermittelt. Diese beiden Informationen werden herangezogen, um das erste Bild derart zu transformieren, dass sich beide Bilder nur noch durch die affine Transformation einer Translation in der z-Ebene unterscheiden. Somit kann durch erneute Phasenkorrelation und Bestimmung des CPS die Positionsänderung ermittelt werden.

## 2.6.2 Kollisionsvermeidung

In der Robotik ist die Kollisionsvermeidung stark mit den Themen Pfadplanung, Navigation, Objekterkennung und Mapping verknüpft (Abbildung 2.22). Dies kommt daher, dass ein Roboter

(jederzeit) stehen bleiben kann und im Gegensatz zu einer fliegenden Drohne wie einem Quadrocopter die Kollisionsvermeidung erstmal nur während seiner eigenen Bewegung, d.h. während der Pfadplanung bzw. dessen Ausführung, verwendet. Besser ist es dagegen, wenn der Roboter oder die Drohne auch unverschuldete Kollisionen vermeiden kann, d.h. auch in der Lage ist, anderen beweglichen Objekten auszuweichen, die sich auf Kollisionskurs befinden.

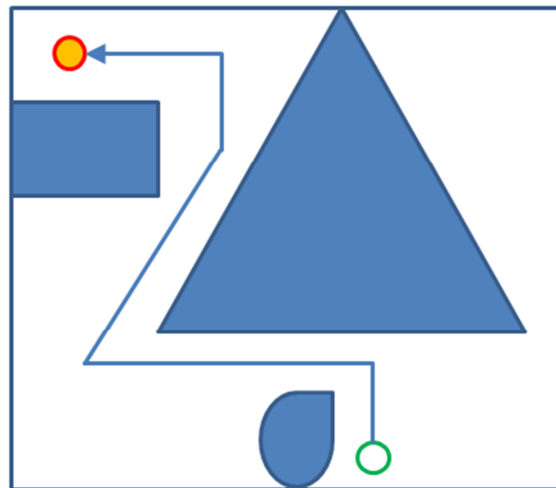


Abbildung 2.22: Prinzip Kollisionsvermeidung basierend auf Pfadplanung und Navigation,  
Grüner Kreis = Start, Roter Kreis = Ziel

Die aktuellen Ansätze, auch solche für Quadrocopter, beruhen häufig auf einer Karte, der Lokalisation des Agenten auf dieser Karte und einer Pfadplanung, die Kollisionen bereits vorher oder dynamisch im Betrieb ausschließt. Eine klassische Methode dazu ist die Potentialfeldmethode, bei der ein Potentialfeld vom Start bis zum Ziel auf der Karte generiert wird. Der Roboter folgt dann stets dem höheren Potential. Dazu ist das Ziel mit dem maximalen Potential ausgestattet und benachbarte Felder erhalten sukzessive immer weniger Potential. Hindernisse werden mit Nullpotential versehen, so dass der Algorithmus automatisch diesen ausweicht. Dieser Ansatz ist denkbar einfach, leidet jedoch unter Stabilitätsproblemen, insbesondere bei engen Passagen. Daher gibt es eine Vielzahl an Weiterentwicklungen wie z.B. das Vektorfeld-Histogramm und die Bubble-Band Methode [65].

Im Unterschied zu einem fahrenden Roboter besteht eine der größten Herausforderung bei einem autonomen Quadrocopter in der Realisierung einer Abstandsregelung zu Hindernissen, die auch ohne Pfadplanung, Navigation und Karte auskommt. Der Quadrocopter kann nicht einfach die Aktuatoren anhalten, denn dann würde er abstürzen. Demzufolge ist er immer in Bewegung. D.h. auch, dass der Quadrocopter womöglich in eine Richtung driftet, ohne dies aktiv zu steuern oder zu wünschen. Auch bei perfekt eingestellten Systemen ist dies spätestens bei Windbewegungen der Fall, wobei der Quadrocopter im Flug selbst permanent Luftverwirbelungen erzeugt. Um Kollisionen zu vermeiden, muss somit permanent der Abstand zu Hindernissen mit Hilfe eines Reglers geregelt werden, wenn die Position nicht anderweitig mit Hilfe eines Sensors und durch Positionsregelung gehalten wird. Selbst in diesem Fall ist das Problem nur vertagt, da eine solche Positionssensorik keine Hindernisse erkennt und im autonomen Flug wieder Abstände zu Hindernissen geregelt werden müssen.

Dabei sind alle Ansätze, die eine Karte benutzen bzw. voraussetzen, rechenintensiv und benötigen viel Speicherplatz, weshalb sie sich nicht für den Einsatz auf einem Mikrocontroller eignen. Die

Kollisionsvermeidung (Abstandsregelung) ist aber wie die Lageregelung eine substantielle Aufgabe eines autonomen Quadropters. Daher ist ein Konzept von Vorteil, dass diese Anforderung unabhängig von einer schweren, teuren und aufwändigen Hardware erfüllt. In dem Kontext dieser Arbeit wird die Kollisionsvermeidung somit als grundsätzlich unabhängig von den Teilen Mapping, Pfadplanung und Navigation betrachtet. Dies hat den Vorteil, dass das System auch dann nicht mit Hindernissen kollidiert, wenn Mapping, Pfadplanung oder Navigation ausfallen oder versagen.

Für dieses Konzept gibt es grundsätzlich zwei Methoden:

- Kollisionsvermeidung durch Ausweichen
- Kollisionsvermeidung durch Abstandsregelung.

Bei ersterer Methode wird bei Detektion eines Hindernisses immer in die entgegengesetzte Richtung bzw. den freien Raum ausgewichen, wozu unterschiedliche Algorithmen und Implementierungen existieren [66] [67] [68]. Dieser Ansatz ist prinzipiell sehr einfach und es gibt bereits käuflich erwerbbar Systeme, die so etwas realisieren. Der Walkera QR Infra X [69] besitzt vier Infrarotsensoren und weicht damit Hindernissen aus, indem er sich in die entgegengesetzte Richtung fortbewegt. Somit kann der Walkera QR Hindernissen ausweichen, nicht jedoch den Abstand regeln. Zudem weicht das System oftmals so dynamisch aus, dass es bei der Ausweichbewegung eine Kollision mit einem Hindernis in der Gegenrichtung verursacht.

Der Nachteil hierbei ist, dass das simple Ausweichen schlecht mit der höherlagigen Navigation und Pfadplanung vereinbar ist. Wenn der Quadropters die Mission verfolgt, in Gebäuden nach Objekten zu suchen, so wäre es hinderlich, wenn er jedes Mal, wenn er ein Hindernis sieht, in eine andere Richtung fliegt, obwohl die Pfadplanung vorsieht, einen bestimmten Bereich nun abzusuchen oder in eine vorgegebene Richtung zu fliegen. Das Suchverhalten wäre unkontrolliert und chaotisch. Bestimmte Bereiche würden oft, andere gar nicht abgesucht. Auch muss der Quadropters längere Zeit an einer Wand entlang fliegen können. Daher ist so ein einfacher Ansatz für einen Quadropters, der eine komplexe Mission erfüllen soll, ungeeignet. Letztere Methode, die aktive Abstandsregelung unabhängig von einer Karte und der Pfadplanung, ist ein neuer Ansatz, der hier verfolgt wird. Dieser lässt sich gut mit existierenden Methoden der Pfadplanung und Navigation aus der Robotik kombinieren. Gleichzeitig ist aber sichergestellt, dass jederzeit Kollisionen vermieden werden können, auch wenn das Mapping oder die Pfadplanung versagt, weil das Hindernis z.B. vorher nicht da war, so wie ein Roboter in diesem Fall stehen bleibt, wenn er zu nah an ein Hindernis herankommt. Ohne funktionierende Kollisionsvermeidung stürzt das System vermutlich oder unter Umständen nach kurzer Zeit ab. Mit funktionierender Kollisionsvermeidung und bei Ausfall der Navigation und des Mapping, mag die Drohne solange „überstehen“, bis diese Funktionen neugestartet sind, um dann die Mission ggf. fortzusetzen, abubrechen oder neu zu starten.

### 2.6.3 Stereo-optische Abstandsmessung

Die stereo-optische Abstandsmessung (SV-System) macht sich grundsätzlich dieselben Prinzipien zu Nutze wie die Tiefenwahrnehmung von Mensch und Tier: Die Bestimmung der Abstände geschieht durch Triangulation und das Verfahren bedient sich dabei Methoden der Bildverarbeitung. Ein und dieselbe Szene wird von zwei unterschiedlichen Perspektiven aus betrachtet und die jeweils korrespondierenden Punkte in beiden Bildern durch sogenannte Matching Algorithmen ermittelt. Durch Triangulation wird dann die Entfernung zum Messobjekt bestimmt. Dazu sind zunächst durch



Kalibrierung die notwendigen Konstanten des stereo-optischen Systems, intrinsische Parameter der Kameras wie ihre Brennweite  $f$ , extrinsische Parameter des Systems wie der Kameraabstand  $T$  sowie Korrekturparameter zur Distorsionskompensation und Skalierung zu bestimmen. Mit Hilfe der Disparität  $d$ , der Messung des stereo-optischen Systems bzw. des Ergebnisses des Matching Algorithmus, kann dann die Entfernung des Punktes  $Z$  mit Hilfe der Brennweite  $f$  und des Skalierungsfaktors  $s$  nach Formel 2.22 berechnet werden.

$$Z = s \cdot \frac{f \cdot T}{d}$$

Formel 2.22: Tiefenformel

Dabei ist  $T$  die Basislinie, d.h. der Abstand zwischen den Kameras bzw. ihren Projektionszentren. Die Disparität gibt die Anzahl an Pixeln an, mit der der jeweilige Pixel zwischen den beiden Bildern verschoben ist und liegt nach dem Matching der Bilder als Disparitätenkarte mit den Disparitäten der einzelnen Pixel vor.

Abbildung 2.23 fasst den von Helge Lauterbach für den AQopterI8 implementierten und auf OpenCV basierenden Algorithmus zur stereo-optischen Abstandsmessung grob vereinfacht zusammen [A16]. Die Kalibrierung wird sowohl einmalig vorab zur Bestimmung der benötigten Parameter durchgeführt als auch in jedem Durchlauf auf die aktuellen Kamerabilder angewendet. Es folgt das Matching zur Bestimmung der Disparitätenkarte sowie die Tiefenformel zur Ermittlung der Tiefenkarte [70] [71] [72] [73] [74] [53] [75].



Abbildung 2.23: SV Algorithmus (grob vereinfacht)

#### 2.6.4 Objekterkennung

Unter Objekterkennung wird die Identifikation und Lokalisation von Objekten bzw. Gegenständen in einem Bild verstanden. Somit ist die Objekterkennung Teil der Bildverarbeitung, wobei mit der Library Open CV bereits fertige, frei zugängliche Lösungen existieren. Abbildung 2.24 fasst die von Christian Reul für den AQopterI8 implementierte und gemeinsam optimierte Objekterkennung zusammen [A09]. Der Algorithmus kann basierend auf Kamerabildern einen roten Ball detektieren.

Zunächst wird in der Zielfarbenkongruenz ein Graustufenbild erzeugt, das die Kongruenz des Bildes zur Zielfarbe abbildet. Beim Summentest, bei dem die Anzahl der Pixel gezählt wird, die die Zielfarbe enthalten, wird vorab überprüft, ob überhaupt genug Pixel zum Erkennen eines roten Kreises vorliegen. Andernfalls kann an dieser Stelle bereits abgebrochen werden, was Rechenaufwand einspart. Falls genug Zielpixel vorliegen, erfolgt eine Kantendetektion basierend auf Sobel-Operatoren. Das Ergebnis der Kantendetektion dient als Eingabe zur Hough-Kreis-Transformation, dessen Resultat bereits die gesuchten Kreise enthält. Da die Suche nach einem passenden Kreis sehr aufwändig ist, wird der erwartete Radius bzw. Radiusbereich vorgegeben. Durch den Kreisfilter werden unzulässige Kreise ausgefiltert, die andere Kreise überschneiden oder nicht aus ausreichend Zielpixeln bestehen. Der Algorithmus kann auch mehrere Kreise bzw. Objekte, d.h. in diesem Fall rote Bälle, gleichzeitig in einem Bild erkennen. Er arbeitet mit ca. 35ms Samplezeit [A09].



Abbildung 2.24: Algorithmus Objekterkennung

## 2.6.5 Lichtlaufzeitmessung mittels 4 Phasenalgorithmus

Der 4 Phasenalgorithmus ist ein Verfahren zur Lichtlaufzeitmessung und wird verwendet in PMD Kameras. Die Vorteile dieses Verfahrens sind die Reduktion von Störeinflüssen anderer Lichtquellen durch eine differentielle Betrachtung und die theoretische Unabhängigkeit der Entfernungsmessung von der Amplitude des Signals.

$$g(t) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin((2k-1)\omega t)}{2k-1} \quad \text{Formel 2.23}$$

$$s(t) = P_{opt} (t - T_L) = a (A + \sin(\omega t - \omega T_L)) \quad \text{Formel 2.24}$$

Zunächst wird das modulierte, ausgesendete Signal  $g(t)$  (Formel 2.23) mit der empfangenen Lichtreflexion  $s(t)$  (Formel 2.24) gefaltet. Diese Faltung ergibt die Korrelationsfunktion  $\varphi_{sg}(\tau)$  (Formel 2.25).

$$\varphi_{sg}(\tau) = s(t) \otimes g(t) = \frac{1}{T} \int_{-T/2}^{T/2} s(t)g(\tau - t) dt \quad \text{Formel 2.25}$$

Einsetzen von Formel 2.23 und Formel 2.24 in Formel 2.25 ergibt Formel 2.26.

$$\varphi_{sg}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} [a (A + \sin(\omega t - \omega T_L))] \left[ \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin((2k-1)\omega(\tau-t))}{2k-1} \right] dt \quad \text{Formel 2.26}$$

Da nur das ausgesendete und das empfangene Signal dieselbe Frequenz besitzen, bleibt nach Integration (Faltung) und Vereinfachung nur noch die gemeinsame, sogenannte DC-Komponente (Formel 2.27) übrig, wobei  $\varphi_0$  dem Phasenunterschied zwischen ausgesendetem und empfangenem Signal entspricht.

$$\varphi_{sg}(\tau) = \frac{2a}{\pi} \cos(\omega T_L + \tau) = \frac{2a}{\pi} \cos(\varphi_0 + \tau) \quad \text{Formel 2.27}$$

Nun kommt der Schritt, dem das Verfahren, der 4 Phasenalgorithmus, seinen Namen verdankt. Zur Bestimmung des Phasenunterschieds, der später zur Laufzeit- und Abstandsbestimmung genutzt wird, werden nun vier Signale (Formel 2.28) mit den Phasenverschiebungen  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  und  $270^\circ$  miteinander verrechnet (Formel 2.29).

$$\begin{aligned}
\varphi_{sg}(0) &= \frac{2a}{\pi} \cos(\varphi_0) \\
\varphi_{sg}(90) &= -\frac{2a}{\pi} \sin(\varphi_0) \\
\varphi_{sg}(180) &= -\frac{2a}{\pi} \cos(\varphi_0) \\
\varphi_{sg}(270) &= \frac{2a}{\pi} \sin(\varphi_0)
\end{aligned}$$

Formel 2.28

$$\frac{\varphi_{sg}(270) - \varphi_{sg}(90)}{\varphi_{sg}(0) - \varphi_{sg}(180)} = \frac{\frac{2a}{\pi} \sin(\varphi_0) - [-\frac{2a}{\pi} \sin(\varphi_0)]}{\frac{2a}{\pi} \cos(\varphi_0) - [-\frac{2a}{\pi} \cos(\varphi_0)]} = \frac{\sin(\varphi_0)}{\cos(\varphi_0)}$$

Formel 2.29

Nun kann unabhängig von der Amplitude a des empfangenen Signals die Phasendifferenz über den Arkustangens ermittelt werden (Formel 2.30) [32]:

$$\varphi_0 = \arctan\left(\frac{\sin(\varphi_0)}{\cos(\varphi_0)}\right) = \arctan\left(\frac{\varphi_{sg}(270) - \varphi_{sg}(90)}{\varphi_{sg}(0) - \varphi_{sg}(180)}\right)$$

Formel 2.30

Der Abstand d kann dann mit Hilfe von Formel 2.31 bestimmt werden [76].

$$d = \left\{ \frac{c * (\pi + \varphi_0)}{80 \text{ MHz} * \pi} + o \right\} \text{ mod } \left\{ \frac{c}{40 \text{ MHz}} \right\}$$

Formel 2.31

Die Modulo-Operation mod wird verwendet, um einen Überlauf bei Mehrdeutigkeit auszuschließen. C ist die Lichtgeschwindigkeit. Der Offset o ist eine pixelabhängige Signallaufzeitverzögerung und die Modulationsfrequenz von 20MHz ist bereits verrechnet.

### 2.6.6 Mapping

Mit Mapping ist das Erstellen einer Karte aus den Daten der OD Sensoren gemeint. In vielen Systemen ist das Mapping bzw. das SLAM-Verfahren der Kern des Systems, da eine Karte und die gleichzeitige Bestimmung der Position in der Karte die Grundlagen der Navigation sind. Entsprechend umfangreich sind die bereits bestehenden Lösungen, die weitestgehend der Robotik entnommen sind. In dieser Arbeit wird ein anderer Ansatz gewählt, womit das Mapping kein wesentlicher Aspekt des Systems darstellt. Deshalb wird an dieser Stelle auf das Thema auch nicht näher eingegangen [77] [78] [79].

Das verfolgte Konzept hat die beiden Vorteile, dass das System auch ohne Mapping autonom agieren kann und dass alle wesentlichen Funktionen neben der Lageregelung, wie die Kollisionsvermeidung und die Positionsregelung, auf einem Mikrocontroller ausgeführt werden können. Dieser eher leistungsschwache Prozessor kann unabhängig operieren, womit das System nicht nur modular ist, sondern auch die Zuverlässigkeit durch eine geringere Abhängigkeit steigt.

Das Mapping des AQopter18 basiert auf dem von Norbert Schmitt implementierten Verfahren. Dieses kann aus den Messungen der US, IR und OF Sensoren eine Karte [vgl. Abbildung 2.25] seiner Umgebung erstellen [A05]. Obwohl die Abbildungen fehlerhafte Artefakte aufweisen, ist festzuhalten, dass die Ausmaße des Raumes, trotz der Verwendung preisgünstiger und relativ

ungenauer Sensorik, grundsätzlich korrekt wiedergegeben werden.

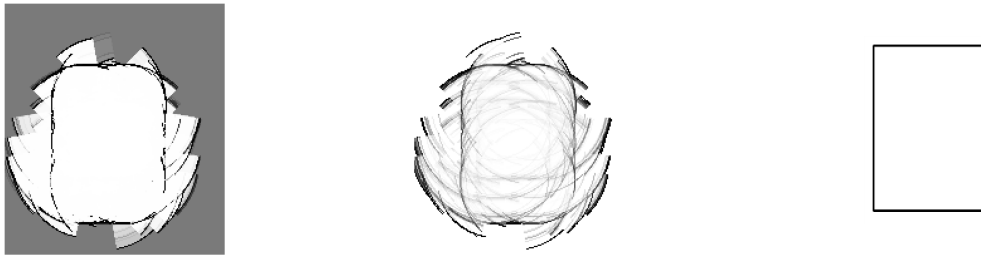


Abbildung 2.25: Karten (links mit Bayes Filter, mittig mit Zählverfahren, rechts Idealfall)

Das implementierte Verfahren basiert auf dem sogenannten Occupancy Grid (engl. für Belegungsgitter). Dies bedeutet, dass die Karte in Zellen aufgeteilt ist. Jeder Zelle ist ein Wert zugewiesen, der angibt, wie wahrscheinlich die Zelle blockiert ist, d.h. ein Hindernis beinhaltet.

Abbildung 2.26 fasst das implementierte Konzept des Mappings zusammen. Dabei werden zunächst die Messdaten der Abstandssensoren bestehend aus Ultraschall- und Infrarotsensoren mit Hilfe der bekannten Position in Weltkoordinaten transformiert. Die Positionsbestimmung nutzt den optischen Flusssensor. Anschließend werden die beobachteten Zellen bestimmt. Bei den Ultraschallsensoren kommt hierzu ein Kegelmodell zum Einsatz. Die Infrarotmessungen werden durch Strahlen modelliert. Es erfolgt eine Gewichtung über alle beobachteten Zellen einer Messung, um z.B. Artefakte, wie die durch den Öffnungswinkel verursachte Ausbreitung blockierter Zellen auf Nachbarn, zu reduzieren. Anschließend werden die Wahrscheinlichkeiten der Zellen aktualisiert. Hierzu stehen die beiden Methoden Bayes Filter und Zählverfahren zur Verfügung. Schließlich wird die Karte (engl. Map) um die Ergebnisse aktualisiert. Die Wiederholrate beträgt 100ms.

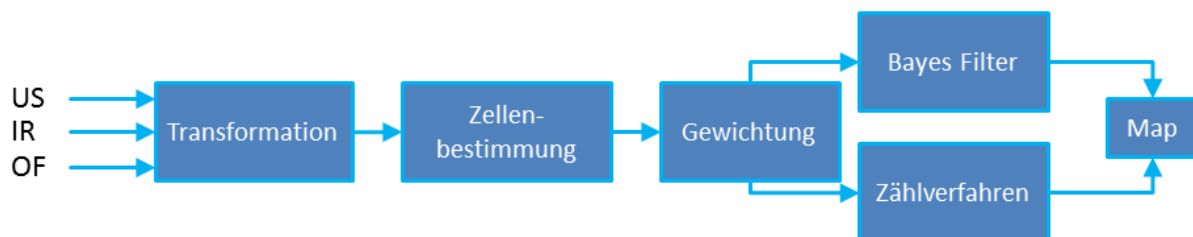


Abbildung 2.26: Konzept Mapping des AQuadropter18 [A05]

### 2.6.7 Quellen

[53] [58] [59] [60] [61] [62] [63] [64] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [77] [78] [79] [P8] [A04] [A05] [A09] [A16] [A23]

## 2.7 Verwandte Systeme (Related Works)

Quadropten und andere Multikopten sind heutzutage nichts Außergewöhnliches mehr. Der Onlinehändler Conrad bietet heute (13.06.2014) vier „Einsteiger“-Systeme für weniger als 40€ an. Das sah in den Anfängen, vor 5-10 Jahren, noch ganz anders aus, wo man für das Geld nicht mal einen geeigneten Sensor bekam. Aus dem Ausland (z.B. China) sind die Systeme noch günstiger zu beziehen. Umgekehrt gibt es heute auch professionelle Systeme zu kaufen, die mehrere Tausend Euro kosten. Diese große Bandbreite und die gewaltigen Sprünge der Preise und

Systemeigenschaften verdeutlichen die rasante Entwicklung, die diese Technologie in den letzten Jahren gemacht hat. Dabei ist die Entwicklung längst noch nicht abgeschlossen ist.

Der folgende Abschnitt gibt einen Einblick in eine Auswahl bestehender Quadrocopter-Systeme, seien es kommerzielle Systeme, d.h. käuflich erwerbliche Komplettsysteme, wissenschaftliche Prototypen, d.h. Entwicklungs- und Forschungsarbeiten sowie Hobbyprojekte. Damit wird eine Einordnung des AQopterI8 ermöglicht und es werden Gemeinsamkeiten und Unterschiede behandelt, woraus sich das Alleinstellungsmerkmal ergibt.

Da eine große Vielzahl an Systemen existiert, kann nur eine begrenzte Auswahl vorgestellt werden, die sich nach Bekanntheit und wissenschaftlicher Bedeutung richtet. Nach dem Grad der Autonomie lassen sich die Systeme in drei Kategorien einteilen: Manuelle Systeme, Semi-Autonome System und Autonome Systeme, im Englischen meist *fully autonomous* genannt.

### 2.7.1 Manuelle Systeme

Zu den manuellen Systemen (ferngesteuert, engl.: remotely piloted) gehören alle Quadrocopter, die einen Piloten benötigen und somit permanent per Fernsteuerung bedient werden. Dabei steuert der Pilot entweder die Winkeländerung oder den Winkel des Quadrocopters. Die Liste an manuellen Systemen unterschiedlicher Hersteller ist gewaltig. Hinzu kommen die Eigenbau-Systeme von Hobbyisten und Bastlern. Entsprechend gibt es in diesem Bereich heutzutage quasi keine wissenschaftliche Forschung mehr.

Weit verbreitet sind diese Systeme im Hobbybereich zum selber Bauen oder aber auch als gekauftes Spielzeug. Entsprechend dienen viele Systeme dem Freizeitvergnügen. Oftmals verfügen die Systeme über Kameras zur Luftbildaufnahme. Darüber hinaus etabliert sich zunehmend eine kommerzielle oder berufliche Verwendung, z.B. bei Reportern für Zeitung und Fernsehen zur Luftbildaufnahme, bei Biologen zur Tierbeobachtung (z.B. Pinguine zählen), bei Such- und Rettungsaktionen wie zum Auffinden von Rehkitz in Feldern (Rettung vor Mähreschern) oder der Suche nach vermissten Personen durch die Bergwacht. Die Liste der möglichen Anwendungen ist ebenso lang wie die Liste der heutigen Systeme [80] [48] [2] [3] [81].

Unterscheiden lassen sich die Systeme z.B. an Hand der Größe. So gibt es sogenannte Nano Quadrocopter, die so klein sind, dass man sie in einer Hand halten kann (Abbildung 2.27). Aber es existieren auch Systeme, die einen Durchmesser von mehreren Metern haben und eine Person transportieren können.

Weitere Unterscheidungsmerkmale sind Preis, Flugdauer, Nutzlast, Verlässlichkeit, Garantie und Zusatzfunktionen wie die Möglichkeit und Qualität, Bilder und Videos per Kamera aufzunehmen und in Echtzeit zu übertragen.



Abbildung 2.27: QR Ladybird von Walkera [Foto: Ulrich Heither]

Aufgrund sicherheitsrelevanter und juristischer Hintergründe verfügt im Grunde jedes Quadrocopter-System mindestens über alle Fähigkeiten eines manuellen Systems. Dies ist vergleichbar mit einem Pilot in modernen Linienflugzeugen: Der Pilot kann jederzeit die Kontrolle übernehmen und das Flugzeug manuell steuern, auch wenn das Flugzeug theoretisch ganz oder meist ohne ihn fliegen könnte.

## 2.7.2 Semi-Autonome Systeme

Mit semi-autonomen Systemen sind solche gemeint, die teilweise bzw. scheinbar autonom agieren können, dabei aber permanent auf ein externes System zurückgreifen, das sie nicht mitführen (vgl. Kapitel 2.2.3). Somit ist die Autonomie an die Verbindung zu dem externen System geknüpft und endet, sobald diese Verbindung unterbrochen wird. Die semi-autonomen Quadrokopter-Systeme lassen sich an Hand des erforderlichen externen Systems in drei Kategorien einteilen, die im Weiteren näher erläutert werden. Zudem werden einige Beispiele semi-autonomer Systeme genannt. Insgesamt sind die Übergänge jedoch fließend, da viele Universitäten mit Firmen kooperieren und für ihre Forschung kommerzielle Systeme verwenden.

### OTS-basierte Systeme

Mit Optischem Tracking (OTS: Optisches Tracking System) sind externe Kamerasysteme, häufig Infrarotkameras, gemeint, die im Raum aufgestellt sind und der Positionsbestimmung des Quadrokopters dienen. Dabei werden in aller Regel gut sichtbare Marker am Quadrokopter befestigt, die Infrarotlicht reflektieren (passiv) oder emittieren (aktiv). Größter Vorteil dieser Systeme ist die hohe Genauigkeit - Auflösungen bis in den Submillimeterbereich sind möglich - bei gleichzeitig hoher Wiederholrate (360 FPS [82]) und geringer Fehleranfälligkeit unter Laborbedingungen. Nachteil der Systeme ist neben dem vergleichsweise hohen Preis vor allem ihre Immobilität und die Beschränkung auf Innenräume, da Sonnenlicht ein großer Störfaktor darstellt. Somit lassen sich Quadrokopter, die OTS benutzen, nicht in unbekanntem Räumen und schlecht im Außenbereich einsetzen.

Die vermutlich bekanntesten Arbeiten über Quadrokopter basierend auf OTS stammen von der Universität in Pennsylvania (GRASP Lab: General Robotics, Automation, Sensing and Perception) und der Eidgenössischen Technischen Hochschule Zürich (ETH). Einige ihrer Arbeiten werden hier kurz beschrieben:

- GRASP

Das GRASP Lab ist mit seinen Arbeiten federführend in der Erforschung und Entwicklung von OTS-basierter Steuerung und Regelung von Quadrokoptern. Mit Hilfe von Prinzipien der Zustandsregelung und Trajektorien-Planung werden Quadrokopter in die Lage versetzt anspruchsvolle Flugmanöver durchzuführen [9]. Dabei fliegen die Quadrokopter z.B. durch enge Deckenfenster, wobei sich die Systeme hochkant oder quer stellen müssen, um überhaupt durch das Fenster zu passen. Die Systeme können zudem mit Hilfe von Klettissen an Wänden und Decken landen. Darüber hinaus können die Quadrokopter im Schwarm fliegen und dabei Musik auf einfachen Instrumenten wie Schlagzeug und Keyboard erzeugen [83] [84] [85].

- ETH

Die ETH hat ein vergleichbares System und zeigt „*erstaunliche athletische*“ Flugmanöver. So können zwei ihrer Quadrokopter miteinander Ping Pong spielen und drei Quadrokopter sind in der Lage, mit Hilfe eines Netzes, das sie gemeinsam tragen, einen Ball zu werfen und wieder zu fangen [86].

Bei all diesen Systemen findet sowohl die Positionsbestimmung als auch die Steuerung auf externer Hardware statt und den Quadrokoptern selbst wird „nur noch“ per Funk gesagt, was sie zu tun

haben.

## GNSS-/GPS-basierte Systeme

GNSS (engl.: Global Navigation Satellite System) ist der Sammelbegriff für globale satellitengestützte Navigationssysteme. GPS, (engl.: Global Positioning System), das satellitengestützte globale Positionssystem des amerikanischen Militärs, ist die älteste, bekannteste und bedeutendste Fassung eines GNSS-Systems. Aus historischen Gründen ist daher meist von GPS die Rede.

Mit Hilfe eines GNSS-/GPS-Receiver, der für wenig Geld zu kaufen ist, erlaubt das System quasi weltweit eine Positionsbestimmung von Objekten und somit auch von Quadrokokptern. Die in hohen Stückzahlen produzierten und somit kostengünstigen Receiver haben üblicherweise eine Genauigkeit von ca. 2m bei einer Wiederholrate im einstelligen Hz-Bereich. Vorteile hierbei sind vor allem der hohe Verbreitungsgrad und der Preis, jedoch sind die Systeme praktisch nur in Außenräumen, d.h. draußen, zu verwenden, da die Receiver in Gebäuden meist nur noch am Fenster ein ausreichend starkes Signale von mindestens vier benötigten Satelliten empfangen. Im Allgemeinen lässt sich sagen, dass die Empfänger auf freiem Feld besser funktionieren als in der Nähe oder innerhalb von Gebäuden, während sie in Tunneln komplett ausfallen. Somit ist GPS nicht hinreichend für einen Quadrokokpter, der autonom innerhalb von Gebäuden operieren soll, während sich solche Systeme im Außenbereich heute weitestgehend durchgesetzt haben.

Heute existiert eine Vielzahl verfügbarer GPS-basierter Systeme, weshalb dies quasi die Standardlösung für den automatischen Flug im Außenbereich darstellt. Die folgende Auflistung ist eine Auswahl bekannter Namen auf diesem Gebiet, wobei hinter jedem Namen mehr oder weniger direkt ein Unternehmen steht. Dies verdeutlicht auch die wirtschaftliche Bedeutung, die diese Technologie schon heute hat. Alle hier aufgelisteten System können Positionen mit Hilfe von GPS halten und anfliegen. :

- Mikrokokpter  
Unter dem Namen HiSystems GmbH, welche hinter der Plattform Mikrokokpter steckt, werden Quadrokokpter, Hexakokpter und Oktokokpter vertrieben, die preislich im Bereich von 1000€ bis 5000€ liegen und eine Kamera zur Luftbilddaufnahme tragen können [3].
- Ardukokpter  
Ardukokpter ist ein auf Arduino basierter Quadrokokpter, der dem Open Source Hobbyprojekt DIY Drones entstammt. Es werden Quadrokokpter, Trikokpter, Hexakokpter und Oktokokpter unterstützt [80] [87]. Fertige System, die eine Kamera tragen können, lassen sich bei der Firma 3D Robotics für ungefähr 1000€ bis 2000€ beziehen [88].
- Aibotix  
Die Firma Aibotix vertreibt einen Hexakokpter, der mit einer Kamera (Spiegelreflex, Wärmebild, Multispektral) ausgerüstet werden kann. Das System ist somit für Inspektionsaufgaben, wie z.B. die Überprüfung von Hochspannungsleitungen, konzipiert. Der Preis für das System liegt bei ca. 30.000€. Das System besitzt vier Ultraschallsensoren zur Höhenregelung über Grund [89] [90].
- Microdrones  
Das Unternehmen Microdrones bietet Quadrokokpter unterschiedlicher Größe für Luftbilddaufnahmen an und adressiert somit Anwendungen aus den Bereichen Überwachung, Sicherheit, Inspektion, Forschung und Landschaftsfotografie (vgl.

Abbildung 2.28). Der md4-200, der kleinste verfügbare Quadrokoopter von microdrones, kostet schätzungsweise 8.000€ bis 20.000€ [91] [7].



Abbildung 2.28: md4-200 (links) und md4-3000 (rechts) von Microdrones auf der UAVveek 2015

### Off-Board Systeme

Einige Systeme, üblicherweise solche, die sehr rechenintensive Methoden der Bildverarbeitung zur Positionsbestimmung nutzen, führen diese Berechnungen auf leistungsstarken externen Computern, sogenannte Off-Board Prozessoren, durch. Damit ist eine ständige Funkverbindung zwischen Quadrokoopter und Off-Board Prozessor notwendig. Fällt der Off-Board Prozessor aus, entfernt sich der Quadrokoopter räumlich bzw. wird die Funkverbindung unterbrochen, kann der Quadrokoopter seine Aufgaben nicht mehr durchführen. Da aber die begrenzte Rechenleistung on-Board eine entscheidende Nebenbedingung bei der Realisierung ist, sind Off-Board Lösungen für ein autonomes System womöglich nicht nur praktisch sondern auch konzeptionell nicht gangbar.

Optische Systeme zur Positionsbestimmung mit Hilfe von Bildverarbeitung sowie Systeme, die dreidimensionale SLAM-Verfahren realisieren, sind weit verbreitet, aber oftmals auch sehr rechenintensiv. Michael Blösch präsentiert ein SLAM-Verfahren (SLAM: Simultaneous Localization and Mapping, gleichzeitiges Lokalisieren und Kartographieren) unter Verwendung eines Laser Scanners, welches externe Hardware benötigt [92]. Slawomir Gronzka beschreibt in seiner Arbeit ein weiteres SLAM-Verfahren, das ebenfalls einen Laser Scanner verwendet und aufgrund der Rechenlast auf einem externen Laptop ausgeführt wird [93]. Ein drittes Verfahren unter Verwendung externer Hardware wird von Sven Lange beschrieben. Dieses ist ein kamerabasiertes Verfahren zur automatischen Landung auf einem vorgegebenen Landeplatz. Mit Hilfe des fest vorgegebenen Musters des Landeplatzes lokalisiert sich das System [94]. Jakob Engel stellt ein weiteres kamerabasiertes SLAM-Verfahren vor, das auf einem EKF basiert. Es erkennt Features in seiner Umgebung und kann sich somit in einem Raum lokalisieren. Als Quadrokoopter kommt die AR Drohne von Parrot zum Einsatz. Das System ist auf einen externen Laptop angewiesen, worauf die Berechnungen stattfinden und wovon es automatisch gesteuert wird [95].

### 2.7.3 Autonome Systeme

Der weit überwiegende Teil heutiger Quadrokoopter-Systeme ist manuell oder semi-autonom, wobei der wissenschaftliche Fortschritt auch hier nicht Halt macht. In den letzten Jahren bis einschließlich heute wurde und wird im Bereich Autonomie von Quadrokooptern sehr viel geforscht und entwickelt. Solche Arbeiten kommen bisher beinahe ausschließlich aus dem universitären Umfeld. Gleichzeitig lässt sich aber sagen, dass gerade aus diesen Arbeiten im Grunde heute noch keine kommerziellen Lösungen auf dem Markt verfügbar sind. Dieser Umstand ist auf der einen Seite dem hohen



technischen Komplexitätsgrad und der noch fehlenden Reife dieser Lösungen im Kontrast zu den beliebig unkontrollierbaren Umweltbedingungen geschuldet. Die Entwicklung autonomer Systeme wird zudem gehemmt durch den schlechten Ruf in der öffentlichen Wahrnehmung (Drohne als Tabuwort), insbesondere bei politischen Entscheidungsträgern, sowie den Risiken und Unsicherheiten solcher Systeme in Bezug auf Haftung und gesetzlicher Regelung, die autonome Systeme ohne Operator bisher untersagt.

### **Laser-Scanner-basierte Systeme**

Den Erfahrungen aus der Robotik folgend, ist die Verwendung von Laser Scannern für autonome Quadrocopter eine verbreitete, wenn nicht die verbreiteste Lösung. Vorteile von Laser Scannern sind die hohe Genauigkeit und Auflösung in Bezug auf Winkel und Abstand. Somit eignen sich Laser Scanner ideal für SLAM-basierte Verfahren zur Positionsbestimmung und Navigation, die schon seit längerem in fahrenden Robotern Anwendung finden. Nachteile von Laser Scannern sind der hohe Preis, ihr Gewicht sowie der Umstand, dass die bisher verwendeten Scanner nur eine Ebene abdecken. Entsprechend ist es hier ein gängiges Verfahren, dass der Quadrocopter auf der Stelle unterschiedliche Ebenen seiner Umgebung scannt, also in der Höhe variiert, bevor er im Anschluss mit Hilfe der dabei erstellen Karte Positionen anfliegt. Kommen nun bewegliche Objekte, die sich nicht auf der aktuellen Ebene befinden, dem Quadrocopter in die Quere, so können diese Objekte nicht erkannt und Kollisionen mit ihnen nicht vermieden werden. Zudem leidet ein Laser Scanner unter vergleichbaren Nachteilen wie andere optische Sensoren und versagt z.B. in verrauchten Räumen und bei lichtdurchlässigen Hindernissen.

Die Firma AscTec vertreibt Quadrocopter, Hexakopter und Oktokopter zugeschnitten für die beiden Anwendungsfelder Luftbild und Forschung. Die Quadro- und Hexakopter kosten ohne Extras zwischen 3.000€ und 5.000€. Ein dazu passender, auf diesem System montierbarer Laser Scanner kostet je nach Reichweite entweder 3.000€ (4m) oder 5.000€ (30m) [96] [97]. Damit bietet AscTec eine Plattform autonomer Systeme für universitäre Forschungsarbeiten an. Shaojie Shen (GRASP Lab) präsentiert ein SLAM-basiertes Verfahren, welches auf einem AscTec Quadrocopter aufsetzt und einen 1.6GHz Atom Rechner verwendet [98]. Damit kann das System ohne externe Hardware operieren. Es kommt der Hokuyo UTM-30LX Laser Scanner zum Einsatz, welcher 5.000€ kostet. Das Prozessorboard kostet ca. 1.000€, womit das System insgesamt bei ca. 11.000€ liegt.

### **Kinect-basierte Systeme**

Die Kinect ist ein von Microsoft und PrimeSense entwickeltes Sensorsystem für die Spielekonsole Xbox 360. Sie besteht aus einem Infrarot-Tiefensensor und einer Farbkamera. Auf Basis dieser Sensoren bestimmt das Sensorsystem ein farbiges 3D-Tiefenbild und kann somit Personen und Gesten erkennen. Der Tiefensensor besteht aus einem Infrarotprojektor (Sender) und einem CMOS-Sensor (Receiver). Der Projektor sendet ein Infrarotgitter aus, welches von den Objekten in der Umgebung reflektiert wird. Der Tiefensensor ist eine mehrdimensionale Weiterentwicklung der bereits vorgestellten Infrarotabstandssensoren. Es existiert ein SDK (Software Development Kit) für die Kinect und eine von Hector Martin entwickelte freie Library OpenKinect. Patrick Bouffard hat die OpenKinect Library unter Verwendung des Betriebssystems ROS genutzt, um einen Quadrocopter (AscTec Pelican) mit einer Kinect auszurüsten. Mit Hilfe der Kinect kann der Quadrocopter seine Höhe bestimmen und Hindernisse vor sich erkennen. Damit kann der Quadrocopter autonom im Raum fliegen. Vorteile dieser Lösung ist die Tatsache, dass die Kinect ein bereits sehr mächtiger Sensor mit einem großen Öffnungswinkel ist, aber gleichzeitig wenig kostet (100€). Das von Patrick

Bouffard verwendete System kostet somit ca. 7.000€ und ist damit deutlich günstiger als eine auf Laser Scanner basierte Lösung. Größter Nachteil ist jedoch der Umstand, dass nur eine Seite mit der Kinect abgedeckt ist und die Kinect recht groß und mit 500g auch sehr schwer ist. Um eine volle Abdeckung zu erreichen, wären mehrere Kinects nötig. Diese Menge an Informationen müsste dann aber auch von dem on-Board Prozessor verarbeitet werden können, was eine weitere Schwierigkeit darstellt. In Bezug auf Größe und Gewicht könnte man sicherlich einiges einsparen, würde man die Kinect aufschrauben und das Gehäuse soweit es geht entfernen. Eine andere Alternative wäre es in Kooperation mit dem Hersteller eine optimierte Lösung für einen Quadrocopter zu entwickeln. Gestützt nur auf optische Sensoren versagt diese Lösung ebenfalls in verrauchten Räumen. Zudem ist zu erwarten, dass starke Sonneneinstrahlung und andere Störungen im infraroten Spektrum den Infrarot-Tiefensensor stören [99] [100] [101] [102] [103].

### Kamera-basierte Systeme

Mit Hilfe von Kameras lässt sich ebenfalls ein autonomer Quadrocopter realisieren. Die Verwendung eines optischen Flusssensors zur Positionsbestimmung in Verbindung mit Abstandssensoren zur Höhenbestimmung ist die einfachste Möglichkeit der Realisierung [P4][P5]. Stephan Weiss stellt ein SLAM-Verfahren vor, das einem Pelican Quadrocopter der Firma AscTec unter Zuhilfenahme einer Kamera erlaubt autonom zu fliegen. Die Berechnungen finden auf einem 1.6GHz Intel Atom Rechner statt. Das System kostet ca. 7.000€. Damit das SLAM-Verfahren funktioniert, müssen die Texturen in der Umgebung ausreichend kontrastreich sein. Bei unifarbene Umgebungen, schlechten Licht- oder Sichtverhältnissen versagt das System [79].

### Weitere Arbeiten

Es existieren noch ein paar weitere Arbeiten, die darüber hinaus andere Sensoren nutzen, wie z.B. die Arbeiten von Samir Bouabdallah und Marcel Beck [48] [104]. In diesem System kommen 4 SRF10 Ultraschallsensoren zur Hinderniserkennung zum Einsatz, während gleichzeitig die Positionsregelung mit einer Kamera erfolgt. Das Verfahren der Kollisionsvermeidung ist im Grunde ein Hindernisausweichen und keine Abstandsregelung. Die vier Ultraschallsensoren reichen nicht aus, die komplette Umgebung abzudecken. Zudem versagen Ultraschallsensoren bei schallschluckenden Oberflächen wie z.B. Stoffen. Eine Abstandsregelung zu Personen anstatt eines simplen Ausweichens wäre mit dieser Sensorik nicht zuverlässig realisierbar.

Eine andere Lösung, bei der 4 Infrarotsensoren zur Kollisionsvermeidung verwendet werden, wird von James Roberts beschrieben [67]. Auch hier wird keine Abstandsregelung realisiert, sondern der Quadrocopter weicht Hindernissen lediglich aus, indem er in die entgegengesetzte Richtung fliegt. Eine Lösung allein mit vier Infrarotsensoren hat zudem den Nachteil, dass große Bereiche um den Quadrocopter herum überhaupt nicht abgedeckt werden. Insbesondere Hindernisse, die sich diagonal zum Quadrocopter befinden, werden übersehen.

#### 2.7.4 Quellen

[2] [3] [7] [48] [67] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [P4] [P5]

## 2.8 Zusammenfassung

### 2.8.1 Einordnung des Systems

Es lässt sich zusammenfassend feststellen, dass die bisherigen autonomen Quadrokopter-Systeme meist Mono-Sensortyp-Systeme sind, d.h. es wird nur eine Sensorklasse zur Umweltwahrnehmung verwendet. Das Kapitel 2.3 zur Sensorik hat aber gezeigt, dass jede Sensorklasse intrinsische Nachteile aufweist. Der Mensch steuert seine komplexen Aufgaben und Interaktionen mit der Umwelt im Vergleich dazu auch nicht allein basierend auf einem Sinn, sondern verfügt zumindest über die klassischen fünf bereits von Aristoteles unterschiedenen Sinne: Tasten, Hören, Sehen, Riechen und Schmecken (vgl. Abbildung 2.29). Somit lässt sich folgern, dass ein autonomes System eine ganze Reihe unterschiedlicher Sensoren benötigt. Ein solches umfassendes System



Abbildung 2.29: Hans Makart "Die fünf Sinne" [11]

existiert aber heute nicht. Insbesondere Systeme mit Kollisionsvermeidung verwenden entweder optische Sensorik oder Ultraschall. Beide Systeme auf sich allein gestellt sind nicht verlässlich.

Des Weiteren lässt sich festhalten, dass alle bisherigen autonomen Systeme sehr teuer sind. Sie basieren im Grunde permanent auf einem SLAM-Prozess zur Positionsbestimmung und Kollisionsvermeidung. Dadurch steigen die Kosten zusätzlich in Bezug auf Geldmittel (extra „dedizierter“ Prozessor, dadurch mehr Gewicht und Größe) und Rechenaufwand. Somit benötigt das System für eine im Grunde sehr einfache Aufgabe wie das Vermeiden von Kollisionen permanent einen sehr rechenintensiven Prozess. Im Falle eines Fehlers der Hardware auf dem High-Level Rechner oder während der Kommunikation zwischen PC und Mikrocontroller, wenn der Prozessor des PCs aufgrund weiterer Funktionen wie Objekterkennung etc. ausgelastet ist oder aus anderen Gründen die Software abstürzt, kann die kritische Aufgabe der Kollisionsvermeidung nicht mehr korrekt wahrgenommen werden. Dabei ist zu bedenken, dass ein Verfahren, das einen komplexen Algorithmus und zwei Rechner (PC + Mikrocontroller) verwendet, sehr viel fehleranfälliger ist als ein simples Verfahren, das erst mal nur auf einem Mikrocontroller läuft und optional ein weiteres, komplexes System hinzuzieht. Daher ist eine Lösung wünschenswert, die eine Kollisionsvermeidung allein auf dem Hauptrechner Mikrocontroller realisiert, so dass dieser Teil unabhängig vom Rest, dem leistungsstarken PC, operabel bleibt.

### 2.8.2 Anforderungen (Constraints)

An diese Arbeit, das Projekt und System AQopter18 sind unterschiedliche Anforderungen gestellt, die im Weiteren näher erläutert werden. Die herausragende Anforderung an das System ist die Forderung nach Autonomie. Der Quadrokopter ist in die Lage zu versetzen, seine Aufgaben autonom nur unter Verwendung mitgeführter Ressourcen (Sensoren, Prozessoren, etc.) durchzuführen. Darüber hinaus gibt es eine Reihe weiterer Anforderungen an das System wie Verlässlichkeit, Einfachheit, Erweiterbarkeit und Modularität. Das System soll grundsätzlich auch unter widrigen Umweltbedingungen in der Lage sein zu operieren, z.B. auch in verrauchten Räumen fliegen können und dabei Kollisionen vermeiden. Demzufolge ist eine Vielzahl an unterschiedlichen Sensoren zu verwenden. Die Anforderungen an Einfachheit, Erweiterbarkeit und Modularität stehen auch mit zwei weiteren Anforderungen in Zusammenhang. Neben dem Zweck der wissenschaftlichen

Forschung im Bereich Autonomie ist das System auch direkt involviert in Lehr- und Lernaufträge der Universität. Dazu ist ein nach Möglichkeit einfaches, erweiterbares und modulares System von Vorteil, welches es mit wenig Aufwand ermöglicht, beliebige Teilaspekte zu analysieren und weiterzuentwickeln. Dabei sind Entwicklungen simultan und unabhängig vom restlichen System erfolgt, aber jederzeit war der volle Zugriff bzw. Eingriff auf alle Systemaspekte möglich. So konnte und kann das System sehr leicht und effizient mit neuen Funktionen ausgestattet werden, auch um neue Konzepte, Theorien und Methoden zu erproben. Darüber hinaus ist das System involviert in weitere Forschungsprojekte am Lehrstuhl und dient als Demonstrator für neuste Technologien des visionären Betriebssystem Rodos.

Mit Einfachheit und Verlässlichkeit ist auch die in Kapitel 2.8.1 beschriebene konzeptionelle Trennung der Aufgaben der Prozessoren gemeint, die alle wesentlichen Funktionen, darunter insbesondere die Kollisionsvermeidung, dem Mikrocontroller zuordnet. Unter Modularität und Verlässlichkeit ist auch gemeint, dass selbst bei Ausfall bestimmter Sensoren und Teilsysteme das Gesamtsystem seine prinzipielle Aufgabe noch erfüllen kann. So ist z.B. eine Lösung zur Kollisionsvermeidung mit Hilfe von Ultraschallsensoren, die dazu auf optische Sensoren angewiesen ist, schon vom Konzept her wesentlich schlechter als eine hier präferierte Lösung, die allein mit Ultraschallsensoren operieren kann aber optional zusätzliche bzw. beliebige Sensoren verwenden kann. Mit Modularität ist nicht zuletzt gemeint, dass das System jederzeit manuell, semi-autonom oder autonom operieren kann und dabei je nach Modus und Konfiguration auf sehr unterschiedliche Hardware zurückgreift und gleichzeitig im Wesentlichen dieselbe Software verwendet.

Neben diesen primär konzeptionellen Anforderungen an das System, ist die kostengünstige Realisierung des Systems eine weitere bedeutsame Bedingung. Im System sind fast ausschließlich kostengünstige Komponenten verbaut. Neben den Kosten für Hardware sind auch die Kosten für Software und Entwicklungszeit zu berücksichtigen, weshalb einer einfachen Lösung einer womöglich weiter optimierten, aber unter Umständen sehr komplexen und aufwändigen Lösung an einigen Stellen Vorrang gegeben wurde bzw. wird.

Aus den postulierten, primären Anforderungen leiten sich ein paar weitere Anforderungen ab, die bei der Arbeit im Auge zu behalten sind. So ist das System nach Möglichkeit mit einer ausreichenden Zahl kostengünstiger, unterschiedlicher Sensoren auszustatten, die gleichzeitig sowohl in Bezug auf Größe und Gewicht als auch in Bezug auf Komplexität der Ansteuerung und Auswertung, insbesondere der benötigten Rechenleistung, den Anforderungen gerecht werden müssen. Gerade die Rechenleistung ist ein begrenztes, kostbares Gut und so wird eine weitere Anforderung an die Algorithmen und Programmierung nach Laufzeitoptimierung gestellt. Dies gilt ebenso für Speicher. Die Anforderungen an das System lassen sich wie folgt zusammenfassen:

- Autonomie
- Low-Cost (Niedrige Kosten)
- Einfachheit
- Verlässlichkeit
- Modularität
- Erweiterbarkeit
- Flexibilität
- Know-How (Dokumentation)

Um die Ergebnis der Arbeit später beurteilen zu können, sind ein paar Anforderungen noch zu quantifizieren:

- **Autonomie und Funktionalität:** Die Anforderung an die autonome Positioniergenauigkeit (Position, Abstand, Höhe) beträgt 1m. Bei der Lokalisierung von Objekten wird eine Genauigkeit von 50cm gefordert. Diese Anforderungen sind deutlich stärker als die Genauigkeit von kostengünstigen GNSS-Receivern, die üblicherweise Genauigkeiten von ca. 2m aufweisen.
- **Kosten:** Die Anforderungen an die Kosten des Gesamtsystems werden auf 3000€ beziffert. Zu Beginn der Arbeiten war bereits jeder Laser-Scanner teurer. Aktuell gibt es kein vergleichbares System zu diesem Preis.

### 2.8.3 Quintessenz

Zusammenfassend kann gesagt werden, dass heute kein System existiert, welches die benannten Anforderungen erfüllt (Kapitel 2.8.2). Aktueller Stand in der Brandbekämpfung ist der Einsatz von Feuerwehrleuten, die zu Fuß, im Stehen oder Liegen, verrauchte Areale absuchen. Quadrocopter, die autonom in brennenden Häusern operieren könnten, existieren nicht.

Verfügbare Quadrocopter-Systeme sind entweder Hobby-Projekte oder kommerzielle Systeme. Die kommerziellen Systeme sind stark eingeschränkt in Bezug auf Erweiterbarkeit, teuer und auch nur semi-autonom. Alle übrigen bekannten Systeme, darunter Hobby-Projekte und Forschungsarbeiten, erreichen nicht den geforderten Grad an Autonomie und Verlässlichkeit, da es sich um Semi-Autonome Systeme oder Mono-Sensorsysteme handelt. Zudem basieren viele der autonomen Forschungsarbeiten auf kommerziellen Systemen, wodurch die Gesamtsysteme sehr teuer und wenige flexibel im Design sind.





## 3.1 Gesamtkonzept AQopterI8

Um die Anforderungen aus Kapitel 2.8.2 zu erfüllen, ist die Entwicklung eines neuartigen Quadropters mit einem eigenen Gesamtkonzept erforderlich. Abbildung 3.1 beschreibt dieses Gesamtkonzept grob.

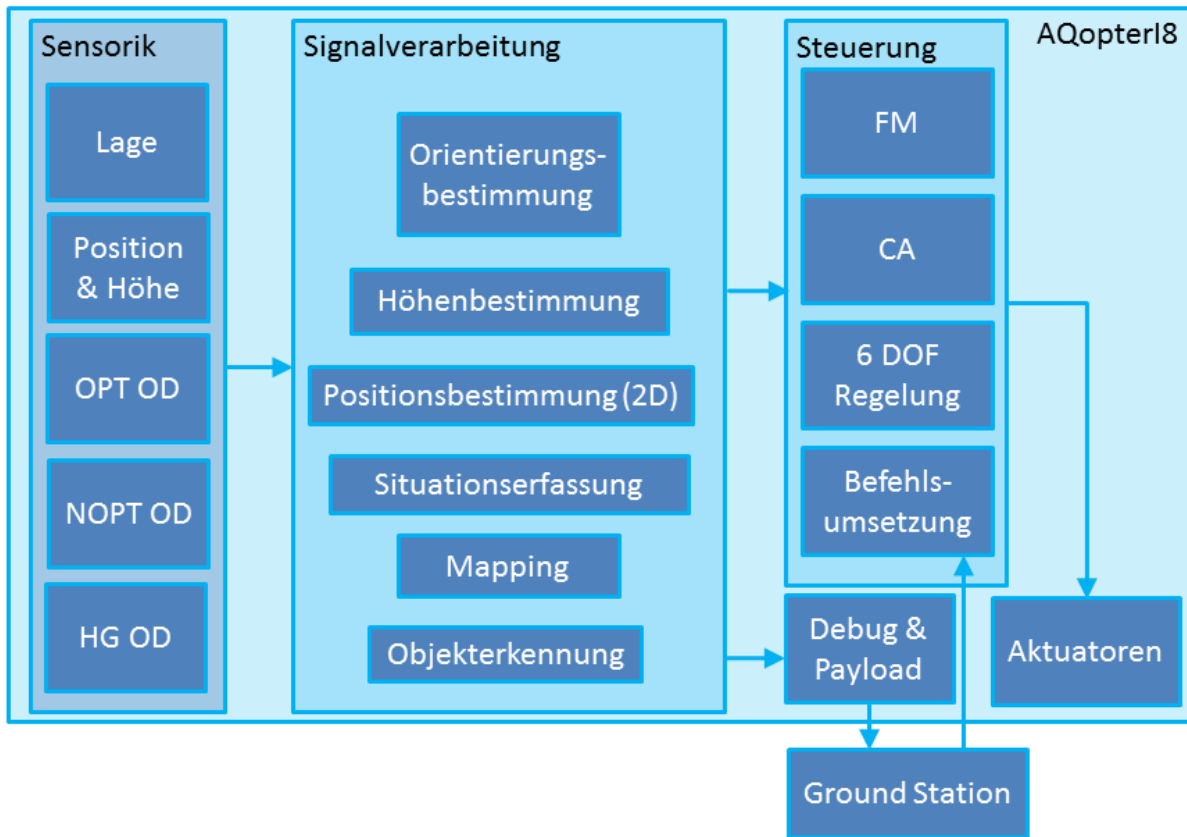


Abbildung 3.1: Gesamtkonzept

Das Gesamtsystem wird wie jeder Roboter in die drei Teilsysteme Sensorik, Signalverarbeitung und Steuerung eingeteilt. Hinzu kommen Blöcke für die Kommunikation mit der Kommandostation (Ground Station) für TM und TC sowie die Aktuatoren.

Eine Besonderheit des Systems liegt in der gleichzeitigen Verwendung von fünf verschiedenen on-Board Sensortypen, die im Weiteren näher erläutert werden:

- **Lage**  
Mit Lage-Sensorik ist die 6 DOF IMU bestehend aus Accelerometer und Gyrometer in Kombination mit dem 3 DOF Magnetometer zur Bestimmung der Orientierung im Raum gemeint: Auch 9 DOF IMU genannt.
- **Position & Höhe**  
Hierunter werden alle Sensoren zusammengefasst, die zur unmittelbaren Bestimmung der 2D Position und Höhe herangezogen werden können. Namentlich sind dies OF-Sensoren für die Position sowie Druck- (Luftdruck) und Abstandssensoren (IR + US) für die Höhe.
- **OPT OD**  
Mit OPT OD, optischer Hinderniserkennung (OD, obstacle detection), sind optische



Sensoren gemeint, die Hindernisse in der Umgebung erkennen: Hier sind dies Infrarot-Abstandssensoren und ein SV-System. Da SV-Systeme und IR-Sensoren sich komplementär gut ergänzen (vgl. Kapitel 2.3), werden beide Sensoren verwendet. Nachteil dieser Sensorklasse ist die Notwendigkeit einer Sichtverbindung zu den Objekten.

- NOPT OD

Mit NOPT OD, nicht-Optischer Hinderniserkennung, sind alle übrigen Technologien, namentlich Ultraschall und Radar, gemeint, um Abstände zu Hindernissen zu bestimmen. Diese nicht-optischen Systeme benötigen keine klare Sicht, können also auch noch durch Störungen, wie z.B. Rauch und Nebel, messen. Somit ergänzen sich diese Sensoren optimal zu den OPT OD Sensoren. Das Konzept sieht die Verwendung von Sonar (Ultraschall) und Radar vor.

- HG OD

Mit HG OD, hochgenauer Hinderniserkennung, sind Sensoren gemeint, die Hindernisse, d.h. Abstände und Größen von Objekten, sehr genau detektieren können. Ein solcher Sensor ist notwendig, wenn der Quadrokopter Engstellen wie z.B. Türen und Fenster passieren soll. Auch die gezielte Benutzung eines Manipulators erfordert eine hohe Genauigkeit. Für weitere intelligente Aufgaben wie Gesten-, Personen- und Objekterkennung sind solche Sensoren ebenfalls erforderlich bzw. nützlich.

Zu diesem Zweck wird hier eine PMD-Kamera verwendet, die ein 3D-Bild im Millimeterbereich liefert. Als günstige Alternative dient ein Laser Scanner, der jedoch nur Daten in einer Ebene liefert. Meist sind Laser Scanner sehr viel teurer als die hier verwendete PMD Kamera und liefern nur Daten in einer Ebene. Andernfalls werden die Systeme nochmals erheblich teurer und schwerer. Dafür ist die Messweite von Laser gegenüber PMD meist deutlich größer.

Ausgewählte Sensoren dieser fünf Sensortypen werden mit Methoden der Signalverarbeitung derart fusioniert, dass ein zuverlässiges Abbild der Umwelt sowie der eigenen Pose in dieser entsteht. Dieser Teil des Konzepts gliedert sich in die folgenden sechs Funktionen:

- Orientierungsbestimmung

Aus den Daten der Lagesensoren wird die Orientierung bestimmt. Diese ist zur korrekten Lage-Regelung permanent erforderlich. Mit Hilfe der Daten weiterer Sensoren kann die Lage korrigiert werden.

- Höhenbestimmung

Das Wissen um die Flughöhe ist zur Höhenregelung erforderlich. Unter Flughöhe wird die relative Höhe verstanden, womit der Abstand über Grund (Abstand zum Boden) gemeint ist. Darüber hinaus ist die absolute Höhe, d.h. die relative Höhe zum Startpunkt, für die Navigation in Gebäuden mit mehreren Etagen erforderlich. Die Flughöhe reicht dazu nicht mehr aus, da sie bei mehreren Etagen mehrdeutig ist. Mit Hilfe eines Luftdrucksensors kann die Mehrdeutigkeit aufgelöst werden. Beide Informationen werden im Konzept berücksichtigt. Zur Datenfusion, der Bestimmung der Flughöhe, kommt der eigens dazu entwickelte Weighted Filter zum Einsatz.

- Positionsbestimmung

Für den autonomen Flug ist das Wissen um die eigene Position permanent erforderlich.

Dies erfolgt mit Hilfe der Positionssensorik basierend auf Prinzipien des optischen Flusses. Es gilt hier die Nebenbedingung, dass dies auch ohne Verwendung einer Karte möglich sein muss. Darüber hinaus sieht das Konzept eine Positionsbestimmung mit Hilfe der Karte (SLAM) vor, wie es schon in anderen Arbeiten erfolgt.

- **Situationserfassung**  
Unter Situationserfassung ist das Ausmachen von Hindernissen in der Umgebung aus den gewonnenen OD Sensordaten gemeint. Dazu fusioniert die Situationserfassung die verfügbaren Informationen. Die Situationserfassung bildet gemeinsam mit der OD Sensorik die Hinderniserkennung. Dieser Vorgang hat jederzeit auch unabhängig vom Wissen um die eigene Position und dem Vorhandensein einer Karte zu erfolgen, damit Kollisionen zuverlässig vermieden werden können. Es kommen die drei Sensortypen OPT OD, NOPT OD und HG OD zum Einsatz: Namentlich sieht das Konzept die Verwendung von IR, US, Radar, SV, Lidar und PMD vor. Die Fusion erfolgt wiederum mittels Weighted Filter.
- **Mapping**  
Mit Mapping ist das Erstellen einer Karte der Umgebung aus den Daten der Situationserfassung gemeint. Es kommt das in Kapitel 2.6.6 beschriebene und von Norbert Schmitt implementierte Verfahren zum Einsatz.
- **Objekterkennung**  
Die Objekterkennung ermöglicht es dem Quadropter Gegenstände zu identifizieren. Es kommt das in Kapitel 2.6.4 beschriebene und von Christian Reul implementierte Verfahren zum Einsatz.

Die Steuerung des Systems lässt sich durch vier Blöcke abbilden:

- **Flight Management (FM)**  
Unter Flight Management werden alle Funktionen und Methoden verstanden, die der höchsten Steuerungsschicht entsprechen. Dies betrifft auf der einen Seite die Verwaltung und Umsetzung des Betriebsmodus, welcher besagt, ob sich das System im manuellen, semi-autonomen oder autonomen Modus befindet. Hinzu kommen weitere Funktionen aus dem Bereich Navigation, dazu zählen die wegpunktbasierte Steuerung und die Planung und administrative Umsetzung von Suchen und anderen komplexen Befehlen.
- **Kollisionsvermeidung (CA)**  
Unter Kollisionsvermeidung wird die Abstandsregelung in alle Raumrichtungen zur Verhinderung von Kollisionen mit Objekten aus der Umwelt verstanden. Die Kollisionsvermeidung greift für ihre Aufgaben auf die Situationserfassung zurück.
- **6 DOF Regelung**  
Mit 6 DOF Regelung ist der 6I4O MIMO Regler gemeint, der alle sechs Freiheitsgrade des Quadropter im Raum regelt. Dazu gehören die Lageregelung, Höhenregelung und Positionsregelung.
- **Befehlsumsetzung**  
Die Befehlsumsetzung verarbeitet die über Funk empfangenen Kommandos und reicht diese je nach Bedarf z.B. an das FM weiter.

## 3.2 Konzeptvergleich: AQopterI8 vs. Stand der Technik

Um die konzeptionellen Unterschiede zu anderen Systemen zu verdeutlichen, beschreibt dieses Unterkapitel zwei semi-autonome Systeme, d.h. je ein OTS-basiertes und ein GPS-basiertes sowie das autonome, rein optische System von Shaojie Shen [98], welches einen Laserscanner und eine Kamera verwendet. An Hand dieser Vergleichssysteme wird dann exemplarisch der Unterschied zum hier aufgestellten Konzept verdeutlicht.

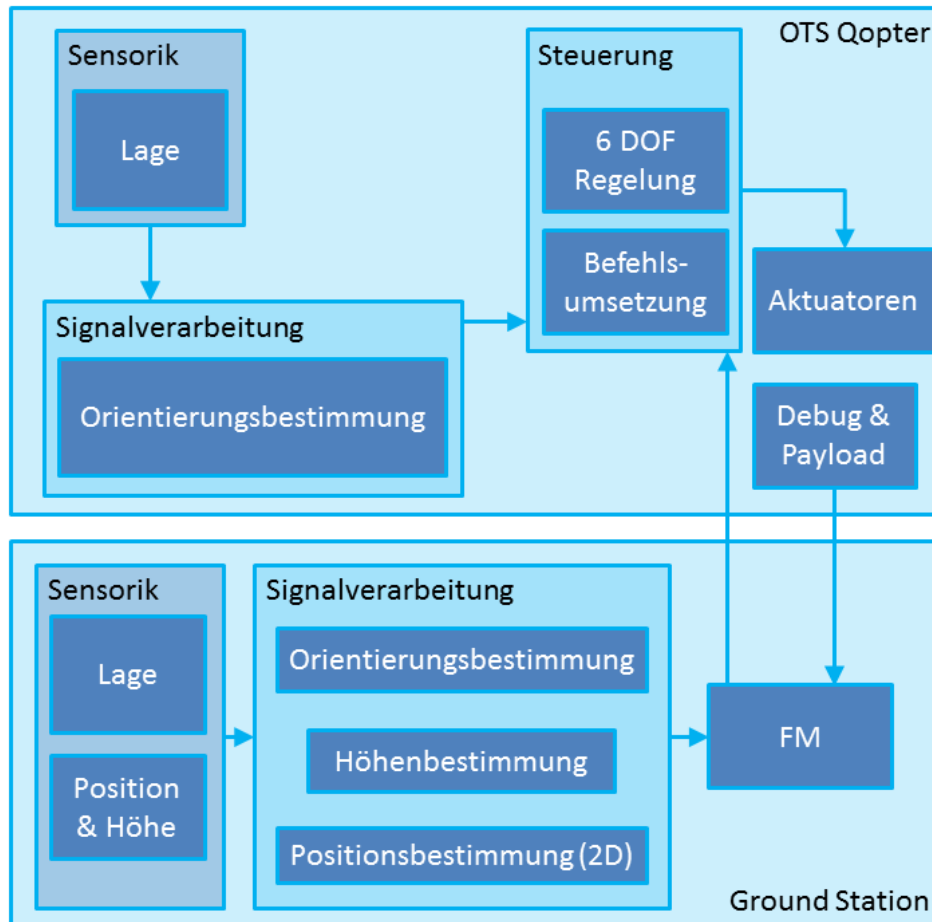


Abbildung 3.2: Konzept OTS-basiertes System

In Analogie zum aufgestellten Konzept verdeutlicht Abbildung 3.2 das Konzept eines Systems basierend auf optischem Tracking. Auf dem ersten Blick wird bereits deutlich, dass dieses System wenig mit dem aufgestellten Konzept gemein hat, da wesentliche Blöcke des Gesamtkonzepts (Abbildung 3.1) wie die OD Sensorik und dazugehörige Signalverarbeitung komplett fehlen. Hinzu kommt, dass viele Teile nicht on-Board, sondern auf einen externen Rechner, der Bodenstation (Ground Station), ausgelagert sind. Auf diesem externen System findet auch die Positionsbestimmung statt. Die Orientierung wird meist redundant bestimmt, einmal on-Board mit Hilfe von IMUs, aber zusätzlich noch durch das OTS, also extern. Da der Quadrocopter bei diesem Konzept im Grunde beinahe nur noch Aktuator ist, kann wahlweise die Steuerung und Regelung komplett ausgelagert sein oder aber on-Board implementiert werden, wobei es sich anbietet, zeitkritische Operationen on-Board und rechenintensive Berechnungen auf der Bodenstation durchzuführen. Die Abbildung macht auch den Unterschied zwischen semi-autonom und autonom deutlich, da hier der Quadrocopter im Grunde automatisch gesteuert wird, d.h. über die

Bodenstation automatisch ferngesteuert wird, und ohne die Bodenstation und externe Sensorik nicht operabel ist.

Abbildung 3.3 illustriert das Konzept eines typischen Systems basierend auf GPS. Auch in diesem System fehlen die OD-Sensorik und die dazugehörige Signalverarbeitung sowie Kollisionsvermeidung. Die Sensorik ist in diesem Fall zwar on-Board, ist aber auf das externe GPS-Signal angewiesen.

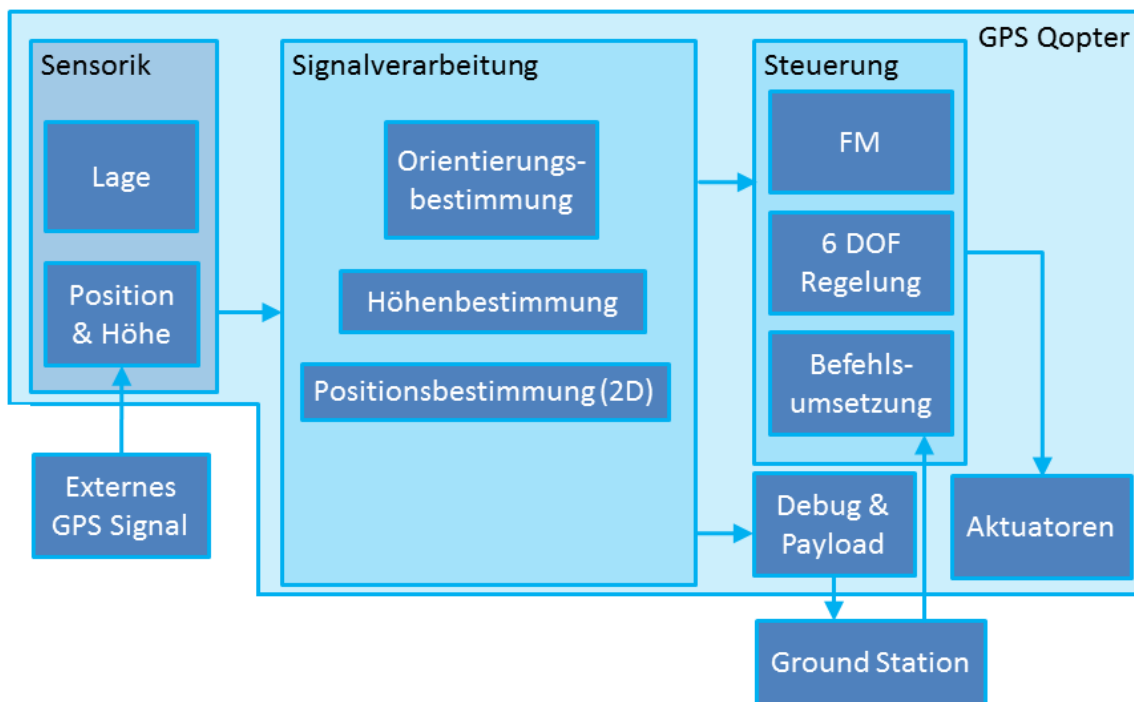


Abbildung 3.3: Konzept GPS-basiertes System

Im Gegensatz zu den beiden erstgenannten Vergleichskonzepten (Abbildung 3.2, Abbildung 3.3) ähnelt das autonome System von Shen (Abbildung 3.4) dem aufgestellten Gesamtkonzept (Abbildung 3.1) konzeptionell in vielen Punkten. Wesentlicher konzeptioneller Unterschied ist die verwendete Sensorik und davon abgeleitet das System der Signalverarbeitung. Shen verwendet nur optische Sensoren, weshalb das System auf gute Sichtverhältnisse angewiesen ist. Die Signalverarbeitung basiert auf einer komplexen SLAM-basierten Lösung unter Verwendung des ICP Algorithmus (Iterative Closest Point) zur Positionsschätzung und eines EKF zur Positionsfusion, was aufgrund der gegenseitigen Abhängigkeiten hier in einem Block dargestellt wird. Dadurch erscheint das System zwar einfacher, ist aber wesentlich komplexer und entsprechend rechenintensiv. Dieser Teil muss daher auf einem leistungsstarken Rechner, bei Shen ist das ein 1.6Ghz Atom, ausgeführt werden, während bei dem weiter oben vorgestellten Gesamtkonzept einige vergleichbare Funktionen wie die Positionsbestimmung und Situationserfassung auch auf einem Mikrocontroller erfolgen kann. Aus den Abbildungen lässt sich vielmehr sagen, dass das selbst aufgestellte Gesamtkonzept modularer und redundanter ausgelegt ist. Zu beachten sei darüber hinaus, dass die genannte Analogie nur konzeptioneller Natur ist. In Bezug auf die jeweilige Implementierung haben beide Systeme kaum

noch etwas gemeinsam.

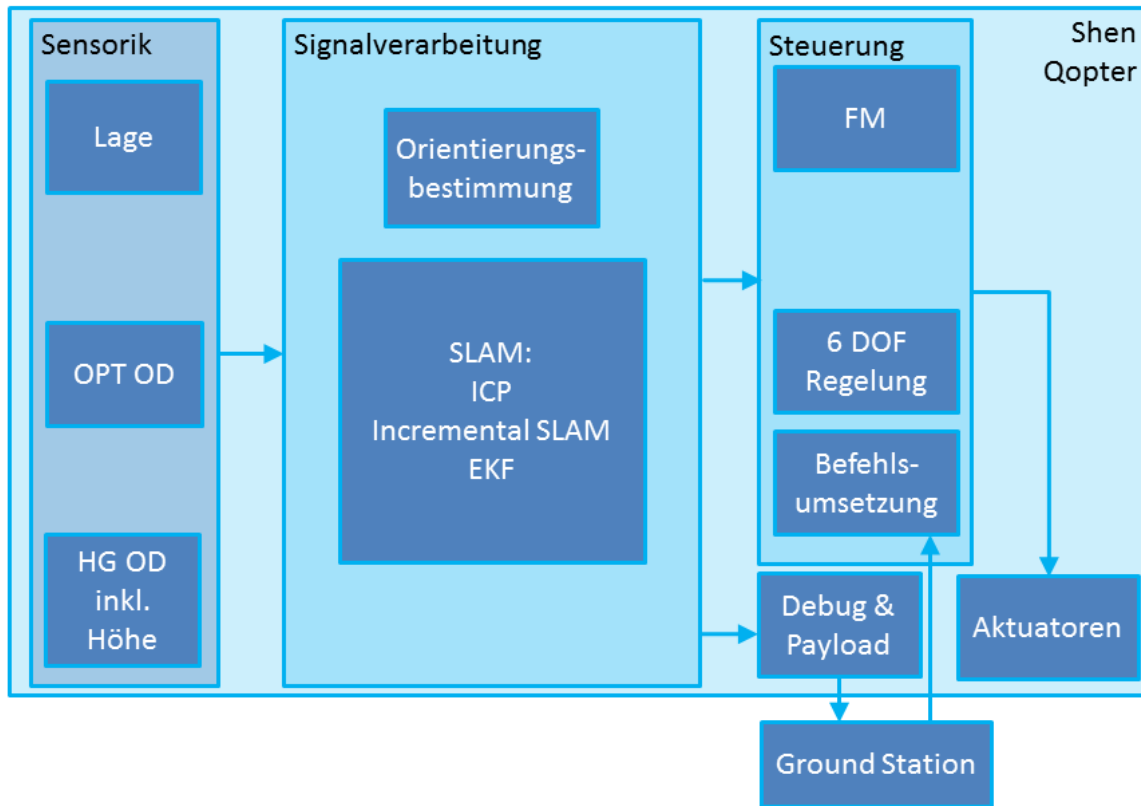


Abbildung 3.4: Konzept Kamera + Laserscanner basiertes System von Shen [98]

### 3.3 Innovative Aspekte

Das folgende Kapitel behandelt Neuheiten der Arbeit und daraus resultierende innovative Anwendungen. Dabei gliedert sich dieses Kapitel in die drei Unterkapitel „Gesamtsystem“, „Lehre“ und „Anwendung“. Das Unterkapitel „Gesamtsystem“ behandelt innovative Aspekte des technischen Gesamtsystems während im Kapitel „Lehre“ neuartige Entwicklungen für die Lehre Thema sind. Das Kapitel „Anwendung“ behandelt innovative, potentielle Anwendungsmöglichkeiten.

#### 3.3.1 Gesamtsystem

Das Gesamtsystem selbst, ein neuartiger autonomer Quadropter, genannt AQopterI8, ist bereits innovativ (vgl. Abbildung 3.5). Der AQopterI8 verwendet eine einzigartige Komposition von Low-Cost Sensorik und ist damit in der Lage, autonom zu fliegen. Dabei kann das System Hindernisse erkennen und Kollisionen vermeiden, Objekte suchen, zählen und lokalisieren, eine Karte seiner Umgebung anfertigen sowie gerichtet fliegen. Dies alles geschieht ausschließlich unter Verwendung bordeigener Ressourcen. Ein solches System existierte bis heute in eben dieser Form nicht.

Innovativ ist bereits das Konzept (vgl. Abbildung 3.1) und die Komposition der Sensorik, ein Zusammenspiel aus 6 DOF IMU sowie Magnet-, Luftdruck-, Infrarot-, Ultraschall-, Radar- und PMD-Sensoren mit Kameras und optischen Flusssensoren, welche zusammen mit eigens entwickelten

Datenfusionssystemen wie dem Weighted Filter (WF, S. 65), dem Triple Awareness Filter (TAF, S. 70) und dem komplementären Quaternionenfilter (S. 92) sowie mit Hilfe bereits bestehender Ansätze, wie dem Kalman Filter, in einem einzigartigen Zusammenspiel aus Sensorik und Signalverarbeitung münden. Der Weighted Filter ist ein Algorithmus zur Gewichtung beliebiger Sensoren und zur einfachen Datenfusion. Der komplementäre Quaternionenfilter ist ein spezieller komplementärer Filter, der Quaternionen fusioniert.



Abbildung 3.5: AQopter18 Gesamtsystem

Ein besonderer Aspekt ist die Tatsache, dass nicht immer alle Sensoren erforderlich sind, d.h. das System ist modular und flexibel, aber auch, dass somit Ausfälle einiger Sensoren automatisch kompensiert werden können. Dazu verwendet der AQopter18 die Vielzahl an Techniken zur Hinderniserkennung. Zudem entfernt sich das System nicht schlicht von Hindernissen (simples Ausweichen) oder ist auf eine Karte angewiesen, sondern regelt aktiv seinen Abstand zu Hindernissen mit Hilfe der aktuell gewonnenen Messdaten. Dadurch kann es sich sehr schnell an Veränderungen in seiner Umgebung anpassen und gleichzeitig seine Mission weiter verfolgen.

Schließlich verfügt der AQopter18 über verschiedene Lösungen zur Positionsbestimmung. Darunter sind mit dem OF-Sensor und Fourier Tracking zwei optische Verfahren, die eine autonome Positionsbestimmung ermöglichen. Des Weiteren kann der AQopter18 autonom Objekte erkennen, suchen, zählen und lokalisieren. Überdies sind ein Kippgelenk, ein Manipulator und eine HMD-Anzeige sowie diverse Steuerungsoptionen in das System integriert.

### 3.3.2 Lehre

Im deutschlandweit einzigartigen Studiengang Luft- und Raumfahrtinformatik der Universität Würzburg entwickeln Studenten im Fach Luft- und Raumfahrtlabor ihren eigenen Quadrocopter. In einem vorbereitenden Kurs lernen die Studenten die Grundlagen der Mikrocontroller-Programmierung, welche sie in diesem Fach benötigen. Unter Anleitung, bestehend aus Konzept, Theorie und Hilfestellung bei der Umsetzung, programmieren die Studenten in 2er-Gruppen über ein

halbes bis ganzes Semester die Software für einen Quadrocopter, wobei Themen wie Sensorik, Lageregelung (Attitude, Yaw), Quaternionen, Kalman Filter, Superposition und mehrdimensionale Regelung am praktischen und anwendungsnahen Beispiel Quadrocopter behandelt werden.

Zum Experimentieren stehen den Studenten zwei eigens dazu entwickelte Quadrocopter-Regelungssysteme (engl.: quadrotor control system, QCS) zur Verfügung. Der erste Aufbau (Abbildung 3.6) ist im Grunde eine inverse Wippe bzw. ein inverses Pendel. An diesem Aufbau können sowohl der Attitude Regler als auch der Yaw Regler nach Wunsch zunächst unabhängig und später gleichzeitig implementiert und eingestellt werden. Der Aufbau kann also um eine Achse kippen (pitch) als auch um eine Achse gieren (yaw). Da er der Einstellung der Regler für zwei Freiheitsgrade dient und somit auch zwei Freiheitsgrade manipuliert, wird er 2 DOF QCS genannt.

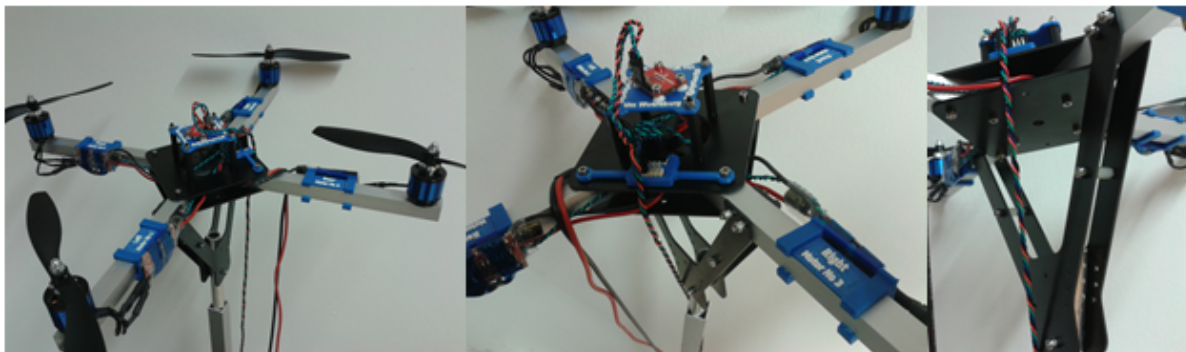


Abbildung 3.6: 2 DOF Quadrocopter-Regelungssystem (2 DOF QCS)

Der zweite Aufbau (Abbildung 3.7) verwendet ein Axialgelenk, mit dem der Quadrocopter auf einer Stange montiert wird [105]. Somit kann sich der Quadrocopter in alle drei rotatorischen Raumachsen frei bewegen und wird daher 3 DOF QCS genannt. Der 3 DOF QCS ermöglicht das Einstellen der vollständigen Lageregelung, womit das Zusammenspiel der Regler für die drei Freiheitsgrade optimiert werden kann. Dies ist erforderlich, da die verschiedenen Regler, insbesondere der Yaw- und die beiden Attituderegler, miteinander konkurrieren. Das System lässt sich somit unter flugähnlichen Bedingungen testen. Experimente haben ergeben, dass das Regelungssystem im freien Flug vergleichbar zum 3 DOF QCS ist, womit derselbe Regler verwendet werden kann. Dagegen unterscheiden sich die optimalen Reglerparameter für das 2 DOF System und das 3 DOF System aufgrund unterschiedlicher physikalischer Zusammenhänge beider Systeme erheblich.



Abbildung 3.7: 3 DOF Quadrocopter-Regelungssystem (3 DOF QCS)

Beide Systeme lassen sich zudem in der Höhe variieren, womit ein Freiheitsgrad hinzukommt. Theoretisch handelt es sich also um ein 3 DOF und ein 4 DOF System. Somit lässt sich dann auch eine Höhenregelung realisieren bzw. dessen Korrektheit analysieren, wobei die Höhenregelung im freien

Flug völlig anders zu parametrisieren ist. Aus Sicherheitsgründen wird diese Funktion nicht in der Lehre verwendet, sondern wurde nur bei der Entwicklung der Höhenregelung genutzt.

Die FH Brandenburg hat zwei Exemplare unseres Systems (Hardware + Software) bereits von uns erworben und plant eine vergleichbare Lehrveranstaltung sowie die Entwicklung eines ähnlichen Systems, basierend auf unseren Vorarbeiten. Die FH Würzburg-Schweinfurt hat ebenfalls 4 Exemplare erworben und verwendet diese in der Lehre.

In einer eigenen Online-Umfrage zu eben diesem Thema wurde untersucht, wie die Studenten diese Veranstaltung in Vergleich zu sonstigen Veranstaltungen ihres Studiums sehen. Die Umfrage und ihre Ergebnisse werden in Kapitel 9 behandelt.

### 3.3.3 Anwendung

Heutige Anwendungsfelder für kommerzielle Quadropter sind beinahe ausschließlich im Außenbereich zu finden. Die Mehrzahl kommerzieller Systeme ist mit einer Kamera und einem GPS-Receiver ausgestattet und wird zum Erstellen von Fotos und Videoaufnahmen genutzt (Luftbildaufnahmen). Diese Aufnahmen dienen z.B. der Landwirtschaft, Archäologie und Geologie, der Dokumentation und Darstellung von Gebäuden und Immobilien (3D Rekonstruktion) sowie der Wartung und Kontrolle von Anlagen. Darüber hinaus werden derart gewonnene Luftbildaufnahmen in Reportagen und Filmen verwendet. Somit haben sich Quadropter heute zu einer günstigen Alternative für Helikopter entwickelt.

In all den oben genannten Anwendungsszenarien und weiteren kann auch ein GPS-basiertes System nicht völlig autonom eingesetzt werden, da es nicht über ein System zur Hinderniserkennung und Kollisionsvermeidung verfügt. Zudem beschränkt die alleinige Positionsbestimmung mittels GPS die Einsatzgebiete und Zuverlässigkeit des Systems, da die Position z.B. in Gebäuden, Tunneln oder bei anderen Signalstörungen wie durch GPS-Jammer nicht korrekt bestimmt wird (vgl. Kapitel 2.2.3).

Ein autonomes System bietet Vorteile wie das Erleichtern der Arbeit des Piloten und die Reduktion der Anforderungen an seine Ausbildung bis hin zur Einsparung von Arbeitskräften, womit letztlich eine Kostenreduktion verbunden ist.

Neben den genannten, bestehenden Anwendungsfeldern eröffnen sich für ein autonomes System, wie es im Konzept dieser Arbeit beschrieben wird, noch weitere Anwendungsmöglichkeiten. Dazu gehören Feuerwehreinsätze, bei denen Quadropter dort eingesetzt werden könnten, wo der Einsatz für Menschen (lebens-)gefährlich ist. Somit werden nicht mehr Kosten, sondern unbezahlbare Menschenleben geschont. Ein Szenario stellt der Brand einer Tiefgarage dar. Die Tiefgarage ist verraucht, einsturzgefährdet und muss nach Verletzten durchsucht werden. Anstatt Feuerwehrleute in Gefahr zu bringen, könnte der AQopter18 die Aufgabe übernehmen. In einem solchen Szenario ist die alleinige Verwendung optischer Sensoren nicht hinreichend, sondern es bedarf zusätzlicher Sensoren wie Ultraschall und Radar.

Auch in sonstigen Katastrophenfällen, bei denen entzündliches Gas, Radioaktivität oder giftige Chemikalien austreten, können zukünftig Quadropter zum Einsatz kommen. Ein konkretes Beispiel ist ein Chemieunfall an der Universität Würzburg vom Jahr 2008, bei dem die Würzburger Feuerwehr im Einsatz war [106]. Zukünftig könnte auch ein solcher Einsatz mit Hilfe eines autonomen Quadropters vereinfacht werden.



Weitere mögliche Anwendungsfelder für den AQopter18 sind die Überwachung von Museen, Industrie- und Chemieanlagen sowie die Erkundung von Bergwerken, Höhlen, Tunneln, Katakomben und Kanalisationen zu Forschungs- und Wartungsarbeiten. Im Jahr 2014 verunglückte ein Höhlenforscher im Riesending-Schacht bei Berchtesgaden, zu dessen Bergung 200 Helfer eingesetzt wurden [107]. Solche Szenarien wären zukünftig vermeidbar, wenn die Erkundung mit einem geeigneten Quadrokopter durchgeführt würde.

## 3.4 Datenfusion mit Weighted Filter

### 3.4.1 Überblick und Einordnung

Mit Weighted Filter (WF) ist der eigens zum Bewerten widersprüchlicher Datenquellen entwickelte Algorithmus gemeint, der in dieser Arbeit zur Höhenbestimmung und Hinderniserkennung Anwendung findet [P3, P10]. Hintergrund ist die Verwendung verschiedener unverlässlicher Sensoren, wobei zunächst keine expliziten Informationen über die Richtigkeit der Sensordaten zur Verfügung stehen. Die Korrektheit der Messungen muss somit mit Hilfe der Messgröße selbst und im Kontext redundanter Messungen überprüft werden, wobei das Ergebnis der Datenfusion die so ermittelte beste Messung ist.

Eben dieses Problem ist typisch bei der Verwendung von Ultraschall und Infrarotsensoren zur Entfernungsmessung. Beide Sensortypen liefern zuweilen aufgrund vielfältiger Fehlerquellen stark verrauschte oder falsche Messwerte. Bei näherer Betrachtung des Problems lassen sich jedoch im Kontext mehrere Phänomene nutzbar machen, die es dem Weighted Filter im Normalfall erlauben, aus widersprüchlichen Daten den korrekten Wert zu bestimmen.

Das erste genutzte Phänomen ist die Tatsache, dass schlechte oder falsche Messungen oftmals stark verrauscht sind. Dies lässt sich durch die Varianz der Messungen bzw. an Hand der Differenz aufeinanderfolgender Messungen feststellen. Des Weiteren kann bei einem dynamischen System wie einem Quadrokopter davon ausgegangen werden, dass Messungen unmöglich über einen längeren Zeitraum konstant sind. Ändert sich ein Wert gar nicht, so ist der Sensor aller Wahrscheinlichkeit nach ausgefallen. Außerdem gilt, ist der Wert außerhalb des gültigen Messbereichs, so wird er ebenfalls verworfen.

Kerngedanke des Weighted Filters ist jedoch der Vergleich aller zur Verfügung stehenden Sensoren bzw. Sensormessungen miteinander. Der Sensor, dessen Messungen am besten mit allen gültigen Messungen übereinstimmen, ist am wahrscheinlichsten korrekt. Dabei ist zu beachten, dass bei der Bewertung nicht allein die aktuelle Messung aber auch nicht alle Messungen berücksichtigt werden sollten. Lange zurückliegende Messungen liefern nur bedingt Informationen über die aktuelle Gültigkeit. Eine vereinzelte Messung mag aber zufällig mal schlecht sein, das System sollte dann aber nicht sofort dem Sensor misstrauen, da dies erfahrungsgemäß zu mehr Schwankungen, einer geringeren Stabilität und somit schlechteren Resultaten führt. Die kurzfristige Historie fließt also beim Weighted Filter in einem geeigneten Maße ebenfalls mit ein.

In Abgrenzung zum weitverbreiteten Kalman Filter behandelt der Weighted Filter primär das Problem der Gewichtung von Messungen. Bei einem Kalman Filter erfolgt die Gewichtung über die Kovarianzmatrizen von Prozessrauschen und Messrauschen. Dabei geht der Kalman Filter davon aus, dass das Rauschen normalverteilt ist, d.h. es ist mittelwertfrei und durch eine Varianz vollständig

beschrieben. Jedoch gibt es eine Reihe von Messfehlern, bei denen diese Voraussetzungen nicht erfüllt sind. Das Problem lässt sich zwar durch eine dynamische Parametrisierung der Kovarianzen umgehen, dennoch stellt dies die größte Schwierigkeit bei der Implementierung eines Kalman Filters dar. Eine Lösung des Problems liefert der Weighted Filter, der jedem Sensor dynamisch Gewichte zuweist. Mit diesen Gewichten können die Kovarianzmatrizen des Kalman Filters aktualisiert werden. Dabei ist insbesondere zu beachten, dass beim Kalman Filter fehlerhafte Messungen das Ergebnis verfälschen, wenn es nicht gelingt, dies bei Messung und Datenfusion zu erkennen und zu verhindern.

Um dieses Problem zu umgehen, verwendet der Weighted Filter nur die beste Messung. In abgeschwächter Form können auch mehrere gute Messungen eingehen. Somit kann der Weighted Filter auch ohne Kalman Filter zur Datenfusion zum Einsatz kommen. Dies ist auch dann sinnvoll, wenn ein Kalman Filter zu aufwändig erscheint, da ein weiterer Nachteil des Kalman Filters in seiner Komplexität und dem damit verbundenen Rechenaufwand besteht.

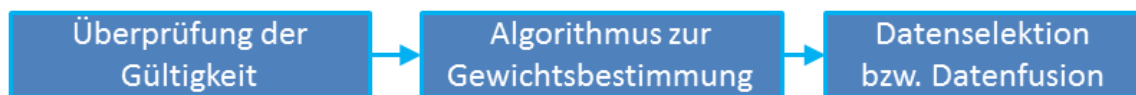


Abbildung 3.8: Überblick Weighted Filter

Abbildung 3.8. zeigt das Konzept des Weighted Filter bestehend aus drei Blöcken. Wie bereits beschrieben, wird zunächst geprüft, ob eine Messung ungültig ist. Ungültige Messungen fließen nicht ein. Im nächsten Schritt werden mit Hilfe des in Abschnitt 3.4.2 beschriebenen Algorithmus Gewichte berechnet. Schließlich erfolgt basierend auf den Gewichten die Datenselektion bzw. Datenfusion.

### 3.4.2 Algorithmus (WF)

Der Weighted Filter teilt zunächst alle relevanten Sensoren in Haupt- und Referenzsensoren ein. Hauptsensoren sind solche, die die Messgröße, z.B. einen Abstand, direkt erfassen. Referenzsensoren dagegen sind solche Sensoren, die zwar bei der Wahl der besten Messung hilfreich sind, aber die Messgröße nicht direkt messen oder aufgrund anderer Überlegungen als Hauptsensor nicht in Frage kommen. Beim Beispiel Abstandsmessung wäre ein Ultraschallsensor ein Hauptsensor, der direkt den Abstand misst, während z.B. eine IMU als Referenzsensor dienen kann, womit die Änderung (Ableitung) des Abstandes bestimmt werden kann. Die Information um die Änderung des Abstandes kann dann bei der Wahl der richtigen Abstandsmessung hinzugezogen werden.

Der Weighted Filter teilt dann jedem Hauptsensor einen Gewichtungsfaktor (Gesamtgewicht) zu. Jedes Gesamtgewicht besteht aus 4 Teilgewichten. Die Berechnung der Gewichtungsfaktoren gliedert sich somit in 5 Teilschritte:

- 1.) Bestimmung des Hauptgewichtes
- 2.) Bestimmung des Referenzgewichtes
- 3.) Bestimmung des Eigengewichtes
- 4.) Bestimmung des Verlaufgewichtes
- 5.) Bestimmung des Gesamtgewichtes

### 3.4.2.1 Wahl der Norm

Zur Bestimmung der Gewichte ist zunächst eine dem Problem angemessene Norm  $N$  auszuwählen. Mögliche Normen sind die Differenznorm  $N_d$  und die quadratische Norm  $N_q$  (Formel 3.1).

$$N_d(m_i, m_j) = |m_i - m_j| \quad \text{Formel 3.1}$$

$$N_q(m_i, m_j) = (m_i - m_j)^2$$

Durch die Wahl der Norm kann bereits auf die Gewichtung von Abweichungen Einfluss genommen werden. Bei der Differenznorm gehen Fehler stets linear und gleichmäßig ein, während bei der quadratischen Norm größere Fehler stärker in die Gewichtung eingehen als kleinere Abweichungen. Grundsätzlich ist eine beliebige monoton steigende Funktion als Norm möglich.

### 3.4.2.2 Bestimmung des Hauptgewichtes

Das Hauptgewicht eines Hauptsensors beschreibt, wie sehr die Sensormessung mit den Messungen der anderen Hauptsensoren übereinstimmt. Bei  $h$  Hauptsensoren beträgt die Anzahl zu bestimmender Hauptgewichte  $\sum_{i=1}^h (h - i)$ , mit  $h > 2$ . Bei weniger als drei Hauptsensoren sind keine Hauptgewichte zu bestimmen. Das Hauptgewicht  $M_i$  des Sensors  $i$  wird durch gewichtete Addition der Normen ermittelt. Die Norm  $N$  wird dabei auf die Kombination aus Hauptsensor  $m_i$  und allen übrigen Hauptsensoren  $m_j$  angewendet (Formel 3.5).

$$M_i = \sum_j \{\alpha_j \cdot N(m_i, m_j)\} \quad \text{Formel 3.2}$$

Hierbei sind  $\alpha_j$  sensorabhängige Parameter, mit denen der Einfluss der Hauptsensoren dimensioniert werden kann.

### 3.4.2.3 Bestimmung des Referenzgewichtes

Das Referenzgewicht eines Hauptsensors beschreibt, wie sehr die Sensormessung  $m_i$  mit den Messungen der Referenzsensoren  $r_j$  übereinstimmt. Das Referenzgewicht  $R_i$  des Sensors  $i$  wird analog zum Hauptgewicht bestimmt (Formel 3.3).

$$R_i = \sum_j \{\beta_j \cdot N(m_i, r_j)\} \quad \text{Formel 3.3}$$

Mit den Faktoren  $\beta_j$  wird der Einfluss der Referenzsensoren parametrisiert. Auf Messungen von Referenzsensoren, die lediglich Änderungen erfassen, wird vor der Normbildung die letzte gültige Messung, d.h. die letzte Ausgabe des WF, addiert.

### 3.4.2.4 Bestimmung des Eigengewichts

Das Eigengewicht eines Hauptsensors beschreibt, wie verrauscht und zueinander plausibel die Messungen eines Hauptsensors sind. Fehlerhafte und stark verrauschte Sensoren, deren Messungen eine hohe Varianz aufweisen, haben ein hohes, d.h. schlechtes Eigengewicht. Zur Bestimmung des Eigengewichts des Hauptsensors  $i$  wird zunächst aus der vorherigen Messung  $m_{i,k-1}$  der nächste Messwert  $E_i$  mit Hilfe der geschätzten Änderung  $S_i$  geschätzt. Mit dem Index  $k-1$  ist der vorherige

Zeitstempel gemeint.

$$E_i = m_{i,k-1} + S_i \quad \text{Formel 3.4}$$

$S_i$  ist die Änderung des Messwertes, welche mit Hilfe der Referenzsensoren gemessen oder aus vorangegangenen Messungen approximiert werden kann. Mit Formel 3.5 kann dann das Eigengewicht  $O_i$  bestimmt werden.

$$O_i = \gamma_o \cdot N(m_i, E_i) \quad \text{Formel 3.5}$$

Mit dem Faktor  $\gamma_o$  wird der Einfluss des Eigengewichts dimensioniert. Hierbei ist zu beachten, dass über die Schätzung  $S_i$  als eine Komponente des Eigengewichts ggf. ein weiterer Referenzsensor mit einfließt. Dementsprechend ist der Faktor  $\gamma_o$  zu wählen.

#### 3.4.2.5 Bestimmung des Verlaufgewichtes

Das Verlaufgewicht beschreibt, wie sehr die aktuelle Messung mit dem letzten Ergebnis  $w_{k-1}$  des Weighted Filters übereinstimmt. Mit Formel 3.6 wird der nächste Messwert  $F_i$  in Analogie zu Formel 3.4 geschätzt.

$$F_i = w_{k-1} + S_i \quad \text{Formel 3.6}$$

Im Anschluss kann mit dem Faktor  $\gamma_v$  zur Parametrisierung das Verlaufgewicht  $V_i$  entsprechend dem Eigengewicht bestimmt werden (vgl. Formel 3.5 und Formel 3.7).

$$V_i = \gamma_v \cdot N(m_i, E_i) \quad \text{Formel 3.7}$$

#### 3.4.2.6 Bestimmung des Gesamtgewichtes

Schließlich wird das Gesamtgewicht  $T_{i,k}$  durch Addition der Teilgewichte und exponentielle Glättung (engl.: exponentially weighted moving average, kurz: EWMA) bestimmt (Formel 3.8). Die Historie fließt über die exponentiell geglättete Mittelwertbildung ein. Aufgrund des EWMA-Anteils schwächt sich der Einfluss vergangener Messwerte mit der Zeit ab. Gleichzeitig gehen sowohl das aktuelle Ergebnis als auch vergangene Resultate in den Gewichtungsfaktor ein.

$$T_{i,k} = \varepsilon * T_{i,k-1} + (1 - \varepsilon) * (M_{i,k} + R_{i,k} + O_{i,k} + V_{i,k}) \quad \text{Formel 3.8}$$

Mit dem Glättungsfaktor  $\varepsilon \in [0,1]$  wird der Einfluss der Historie eingestellt. Je größer der Wert gewählt wird, desto konservativer verhält sich das Filter, d.h. einem einmal vertrauten Sensor wird wieder vertraut.

### 3.4.3 Datenselektion / Datenfusion

Das Konzept des Weighted Filters sieht neben der Verwendung in Kombination mit einem Kalman Filter zwei Ansätze zur Bestimmung des Ergebnisses vor: Datenselektion und Datenfusion. Mit

Datenselektion ist gemeint, dass die Sensormessung mit dem besten (niedrigsten) Gewicht weiterverwendet wird. Dadurch verbessert sich die Messung des jeweiligen Sensors zwar in keiner Weise, jedoch fließt auch keine falsche Messung ein. Die grundlegende Überlegung dabei lautet: Unter mehreren Messwerten kann nur einer Wert wahr sein.

Eine andere Möglichkeit ist die Datenfusion. Bedenkt man, dass Messungen stets auch zufällig verrauscht sind, kann das Rauschen durch Mittelung reduziert werden. Dazu werden alle Messungen der Hauptsensoren entsprechend ihres Gewichtes miteinander verrechnet (Formel 3.9).

$$w_k = \sum_i \frac{\frac{\sum_h T_{h,k}}{T_{i,k}}}{\sum_j \frac{\sum_h T_{h,k}}{T_{j,k}}} \cdot m_{i,k} \quad \text{Formel 3.9}$$

Die Auswahl zwischen Datenselektion und Datenfusion kann a prio festgelegt werden. Es bietet sich aber an, dies dynamisch und abhängig von der Relation der Gewichte vorzunehmen. Liegen die Gewichte nahe beieinander, d.h. ihre Abweichung ist unterhalb einer bestimmten Schwelle, so werden die Daten fusioniert. In dem Fall weichen die Sensoren nur gering voneinander ab und alle Sensoren können als zuverlässig angesehen werden. Andernfalls macht eine Datenselektion mehr Sinn, da die Sensoren größere Abweichungen haben und somit davon ausgegangen werden muss, dass falsche Werte gemessen wurden.

Formel 3.9 ist zudem zu entnehmen, dass stets nur die aktuellen Messungen in das Endergebnis  $w_k$  des Filters einfließen. Das WF beeinflusst das Zeitverhalten des Signals somit gar nicht und verschuldet theoretisch auch keine Verzögerung. Gleichzeitig wird aber auch das Rauschen kaum verringert. Zweck des WF ist es aber, falsche Messungen aufzuspüren und zu verwerfen, statt wie ein Kalman Filter Rauschen zu beseitigen.

### 3.4.4 Parametrisierung

Bis auf die wenigen Parameter  $\alpha_j$ ,  $\beta_j$ ,  $\gamma_o$ ,  $\gamma_v$ ,  $\varepsilon$  zur Gewichtung der Sensoren, des Eigengewichts, des Verlaufs und der Historie, liefert der Weighted Filter bereits alle nötigen Vorgaben, d.h. Rechenvorschriften zur Implementierung. Die Wahl der Parameter ist vergleichsweise einfach. Es genügen konstante Werte, um dennoch eine dynamische Gewichtung der Sensoren zu erreichen und um fehlerhafte Messungen ausschließen zu können. Die Parameter sind so zu wählen, dass sie dem a priori Wissen um die Genauigkeit der Sensoren folgt. Die Gewichtung sollte dabei jedoch ausgewogen bleiben, wenn bestimmte Sensormessungen nicht von vornherein nur marginal eingehen sollen.

Hierbei ist zu berücksichtigen, dass der WF schon aufgrund seiner Struktur den Sensor bevorzugt, der im letzten Durchlauf das beste Ergebnis lieferte. Der Glättungsfaktor  $\varepsilon$  sollte daher mit Bedacht gewählt werden. Seine Optimierung stellt ein Trade-Off Problem zwischen schneller Reaktion und Stabilität dar. Eine schnelle Reaktion ist dann wünschenswert, wenn Änderungen, z.B. gültige Messgrößen sprünge, schnell in die Messung eingehen sollen. Ändern sich die Gewichte dagegen zu sprunghaft und wird zu oft zwischen den Sensoren gewechselt, so kann dies spätestens bei einer darauf basierenden Regelung zu Problemen führen.

### 3.5 Datenfusion mit Triple Awareness Filter (TAF)

Der Triple Awareness Filter (TAF, dt.: dreifache Wahrnehmung) ist ein selbstentwickelter, einfacher Algorithmus zur Datenfusion bei zweidimensionaler Hinderniserkennung und Kollisionsvermeidung, ohne Manipulation der Höhe. Er geht von dem Konzept aus, die Umgebung des Quadropters über 8 Sektoren als die verfügbaren Abstände in 8 Richtungen wahrzunehmen (Abbildung 3.9 links). Die Aufgabe des Filters besteht nun darin, aus drei 45° versetzt gemessenen Abständen die kürzeste Entfernung auszuwählen und dabei Fehler und Störungen zu unterdrücken. Dieser kürzeste Abstand entspricht dem verfügbaren Platz in die jeweilige Richtung. Wie in Abbildung 3.9 (mitte) zu sehen ist, wird aus dem Abstand nach vorne (rot markiert, engl. main) und den Abständen der direkten Nachbarn – vorne links und vorne rechts (gelb markiert, engl.: direct neighbor) - ein gemeinsamer Abstand nach vorne bestimmt. Diese Reduktion ist notwendig und korrekt, da das Problem der Kollisionsvermeidung hinreichend beschrieben ist durch den verfügbaren Abstand in die vier Raumrichtungen vorne, links, rechts und hinten. Alle weiteren Abstände darüber hinaus sind Kompositionen dieser Raumrichtungen und lassen sich somit durch selbige ausdrücken.

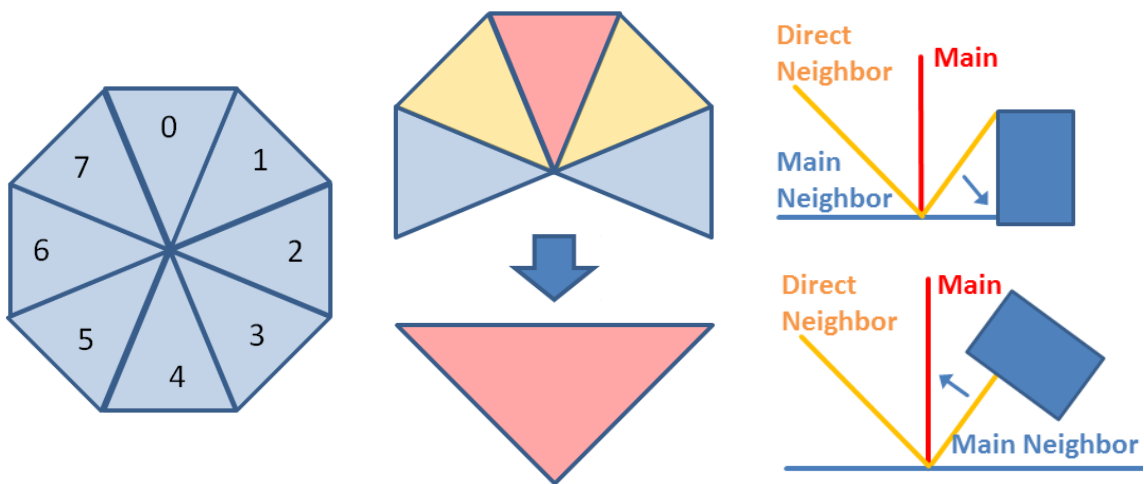


Abbildung 3.9: TAF Konzept

Damit besteht das zugrundeliegende Problem in der Frage, wie multiple Abstände unterschiedlicher Richtungen miteinander zu fusionieren sind. Dazu liefert der TAF Algorithmus eine Lösung. Das Ziel der Fusion ist die Bestimmung des minimalen Abstandes zur Vermeidung von Kollisionen. Die Verwendung der Min-Funktion, d.h. die schlichte Bildung des Minimums, ist nicht ausreichend, da hierbei die nachfolgend beschriebenen Schwierigkeiten auftreten.

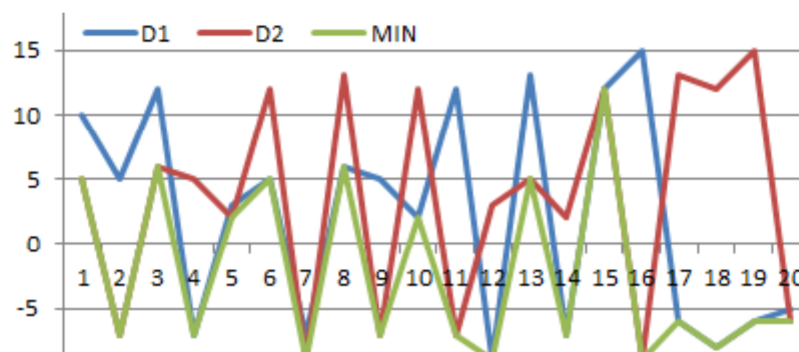


Abbildung 3.10: Minimum (grün) zweier verrauschter Signal D<sub>1</sub> und D<sub>2</sub>

Bei mehreren stark verrauschten Signalen ist deren Minimum deutlich niedriger als der Mittelwert jedes einzelnen Signals (vgl. Abbildung 3.10). Die Mittelwerte von  $D_1$  und  $D_2$  betragen 2,25 und 3,6. Der Mittelwert der Min-Funktion  $\min(D_1, D_2)$  beträgt -2,25 und entspricht kaum mehr den beiden ursprünglichen Signalen. Somit wird die Messung durch Bildung des Minimums stark verfälscht. Dieses Problem hängt sehr stark mit dem sprunghaften Verhalten der Min-Funktion zusammen. Es führt zu einem sehr schnellen Wechsel zwischen den verschiedenen Signalen bzw. Sensoren, was einer stabilen Regelung zuwider läuft. Deshalb verwendet der TAF nicht lediglich die aktuelle Messung, sondern die Ergebnisse der letzten drei Messungen.

Ein weiteres Problem besteht darin, dass die Min-Funktion den Kontext, in dem die Messungen stehen, nicht berücksichtigt. Der TAF hat zum Ziel, den kürzesten relevanten Abstand in eine Raumrichtung zu ermitteln. Wenn der Quadrokopter sich z.B. in einem Gang befindet, so messen die 45° nach vorn versetzten Sensoren auch die Wand an der Seite. Über die Min-Funktion würde die Wand an der Seite vorne miteinfließen. Um dies zu vermeiden, verwendet der TAF auch die 90° versetzten Sensoren (Abbildung 3.9 rechts: blau, main neighbor). Ist deren Abstand niedriger als der Abstand der direkten Nachbarn, so fließt der entsprechende direkte Nachbar nicht ein. Abbildung 3.11 beschreibt den TAF Algorithmus in Pseudo-Code.

Diesen Ausführungen mag entgegen gehalten werden, dass im beschriebenen Beispiel mit dem Gang die Wand nicht nach vorne wandert, sofern die korrekte Richtung miteinfließt. In dem Fall würde eine Vereinfachung wie beim TAF nicht postuliert. Es stellt sich dann aber die Frage, wie die Messungen in die Regelung einfließen sollen und wie dabei die unterschiedlichen Richtungen Berücksichtigung finden. Spätestens bei der Regelung muss eine Vereinfachung oder Fusion stattfinden. Experimente mit zusätzlichen überlagerten Reglern haben gezeigt, dass das System stabiler ist, wenn nur ein Regler pro unabhängige Raumachse verwendet wird. Zudem ist das Problem durch die vier Abstände vorne, hinten, links und rechts minimalistisch beschrieben.

---

#### Algorithmus TAF

---

**Input:** distance main neighbor left:  $m_l$

**Input:** distance direct neighbor left:  $n_l$

**Input:** distance main:  $m$

**Input:** distance direct neighbor right:  $n_r$

**Input:** distance main neighbor right:  $m_r$

**Input:** tolerance:  $\tau$

**Savings:** last weights:  $w_0, w_1, w_2$

**Output:** fused minimum distance:  $d$

---

**if** ( $m_l < n_l \cdot \tau$ ) **then**  $n_l = \infty$  **end if**

**if** ( $m_r < n_r \cdot \tau$ ) **then**  $n_r = \infty$  **end if**

**if** ( $m < n_l$ ) **then**

**if** ( $m < n_r$ ) **then**  $w_0 = 0$

**else**  $w_0 = 1$

**end if**

**else**

**if** ( $n_l < n_r$ ) **then**  $w_0 = -1$

**else**  $w_0 = 1$

**end if**

**end if**

$$S = \sum_i w_i$$

$$w_2 = w_1$$

$$w_1 = w_0$$

$$d = m$$

**if** ( $S < -1$ ) **then**  $d = n_l$  **end if**

**if** ( $S > 1$ ) **then**  $d = n_r$  **end if**

**return**  $d$

---

Abbildung 3.11: TAF Algorithmus







## 4.1 Überblick Hardware

Das folgende Kapitel widmet sich der Hardware des AQopterI8 bestehend aus Elektronik und Mechanik. Da während der Entwicklungszeit eine ganze Reihe unterschiedlicher Systeme aufgebaut wurde und deren ausführliche Beschreibung den Rahmen dieser Arbeit sprengen würde, wird nur auf die finale Version eingegangen. Eine Auswahl unterschiedlicher Versionen und deren Entwicklungsgeschichte, zu denen die Software stets kompatibel gehalten wurde, finden sich im Anhang (11.4).

## 4.2 Elektronik

Zur Elektronik gehören einmal das elektronische Design, d.h. die Komposition und Ansteuerung der elektronischen Komponenten (4.2.1) sowie deren elektrische Verschaltung zur Spannungsversorgung (4.2.2). Da das elektronische Design des Systems regelmäßig modifiziert und erweitert werden musste, um neue Ansätze zu erproben, wurde Wert auf eine flexible Lösung gelegt. Eine hohe Flexibilität und Modularität im Design stellt auch eine Anforderung an das Gesamtsystem dar. Deshalb wurde bisher auch keine eigene Platine gefertigt, da damit das Design des Systems und die Wahl der Komponenten zementiert werden.

### 4.2.1 Elektronisches Design

Abbildung 4.1 zeigt das elektronische Hardware-Design des AQopterI8. Es zeigt die Schnittstellen und Verknüpfungen aller funktionalen Komponenten des Systems wie Prozessoren, Sensoren, Aktuatoren und Funkmodule, die in dieser Arbeit verbaut und verwendet wurden. Dazu gehören zentral angeordnet die beiden Prozessoren: Der 32-bit AVR Mikrocontroller UC3A1512 mit 60MHz Prozessortakt, 512kB Flash und 64 kB SRAM sowie der PICO-ITX PC LP-180. Der LP-180 ist ein T56N Zweikernrechner der Firma AMD und verfügt über 2x 1,6 GHz Takt und 4GB RAM. Der Mikrocontroller stellt die nötigen Schnittstellen (I<sup>2</sup>C, PWM, ADC, SPI) zur Verfügung, welche der PC nicht liefert und unterliegt harter Echtzeit. Daher ist er zur Regelung des Systems und zur Ansteuerung der Aktuatoren prädestiniert. Zu seinen Aufgaben gehört ebenfalls die Signalverarbeitung der mit ihm verbundenen Sensorik. Der PC dagegen übernimmt alle Aufgaben, die eine hohe Rechenleistung erfordern, wie das Mapping, das Fourier Tracking, die Objekterkennung, die PMD-basierte 3D Hinderniserkennung und die Lidar-basierte 2D Hinderniserkennung.

Die insgesamt 42 Sensoren werden entsprechend ihrer Verwendung in Sensoren zur Höhenbestimmung, Orientierungsbestimmung, Hinderniserkennung, Positionsbestimmung und Objekterkennung gegliedert. Zur Höhenbestimmung werden die zwei Infrarotsensoren GP2Y0A710K0F (kurz A71) und GP2Y0A02YK (kurz A02) der Firma Sharp [15], der Luftdrucksensor BMP085 von Bosch [108] und der Ultraschallsensor SRF02 [109] der Firma Devantech [110] genutzt. Zwei Infrarotsensoren sind nötig, damit ein Messbereich von 20cm bis 5m abgedeckt werden kann. Der GP2Y0A02YK, auch kurzer oder kleiner IR-Sensor (IRS) genannt, misst Entfernungen zwischen 20cm und 150cm. Der GP2Y0A710K0F, auch langer IR-Sensor (IRL) genannt, kann Entfernungen zwischen 80cm und ca. 5m messen.

Die Orientierungsbestimmung verwendet die IMU3000 sowie die MinIMU-9, weil sich mit der Kombination aus dem Gyroskop der IMU3000 sowie dem Accelerometer der MinIMU-9 gute Ergebnisse erzielen ließen. Zudem kann bei Ausfall der einen IMU ohne viel Aufwand mit der anderen IMU weitergearbeitet werden. Die MinIMU-9 verfügt zudem über einen Magnetometer, der zum

Einnorden und zur Drift-Kompensation genutzt werden kann.



Abbildung 4.1: Elektronisches Design (Hardware-Design der elektronischen Komponenten)

Dem Konzept folgend nutzt die Hinderniserkennung ebenfalls die bereits beschriebenen Ultraschall- und Infrarotsensoren als NOPT OD bzw. OPT OD Sensoren. Die beiden frontal angebrachten Logitech C920 Kameras [111] des SV-Systems ergänzen die OPT OD Sensoren. Darüber hinaus dienen der RP Lidar [112] und die PMD nano [113] als HG OD Sensoren zur hochgenauen Hinderniserkennung.

Die Position kann sowohl mit dem ADNS-3080 [31] als auch mit der Kamera C920 unter Verwendung des Fourier Trackings bestimmt werden. Beide Sensoren sind auf den Boden gerichtet. Damit funktioniert die Positionsbestimmung auch dort, wo kein GPS- Empfang möglich ist, z.B. in Gebäuden. Für den Einsatz im Freien wurde zur redundanten Positionsbestimmung ein GPS-Receiver

der Firma 3DR in den AQopter18 integriert.

Zur Objekterkennung wird die Kamera C270 der Firma Logitech verwendet. Es existieren drei Möglichkeiten der Funkverbindung zur Kommunikation (TM/TC) mit dem System. Über das Bluetooth Modul BTM222 können Nachrichten zwischen der Bodenstation und dem AVR-Board ausgetauscht werden. Über W-LAN kann eine Verbindung zum LP-180 aufgebaut werden. Darüber hinaus erhält ein DSM RC Receiver (Satellit) Befehle der 7-Kanal DX7 Funkfernsteuerung der Firma Spektrum [114]. Wenngleich die Kommandierung grundsätzlich sowohl über Bluetooth als auch über W-LAN erfolgen kann, so stellt die Fernsteuerung eine sehr verlässliche Steuerungsalternative dar. Sie dient vor allem dem manuellen Flug und dem Eingriff im Fehlerfall.

Herz des Hardware-Designs ist der I<sup>2</sup>C-Bus, über den die beiden IMUs ausgelesen und die Motorcontroller (engl.: Brushless Controller, kurz BICtrler) angesteuert werden, welche die Kommutierung der bürstenlosen Motoren leisten. Ebenfalls am I<sup>2</sup>C-Bus angeschlossen sind die Ultraschallsensoren SRF02, der Luftdrucksensor BMP085, sowie die ADC-I<sup>2</sup>C Konverter, welche die Infrarotsensoren zur Hinderniserkennung auslesen. Somit steuert der UC3A1512 als I<sup>2</sup>C-Master 22 Slaves.

Über PWM werden die Servos für das Kippgelenk (Gimbal) angesteuert. Der ADNS-3080 wird über SPI ausgelesen. Der DSM Receiver, das Bluetooth-Modul und der GPS-Receiver nutzen USART. Die Kommunikation zwischen dem UC3A1512 und LP-180 läuft ebenfalls über USART. Die Kameras C270, C920 und PMD nano sowie der RP Lidar und das W-LAN Modul sind über USB an dem LP-180 angeschlossen.

Zum Design sei abschließend angemerkt, dass es die Maximalkomposition darstellt. Um ein simultanes Arbeiten am System zu ermöglichen, wurden 10 Systeme mit unterschiedlichen Konfigurationen aufgebaut. Es sind also nicht stets alle Komponenten verbaut bzw. erforderlich. Somit können einzelne Komponenten durchaus weggelassen werden. Somit kann abgestimmt auf die konkrete Anwendung eine optimale Komposition zusammengestellt werden.

#### 4.2.2 Elektrische Verschaltung

Abbildung 4.2 beinhaltet die elektrische Verschaltung des AQopter18 auf das Wesentliche reduziert. Der Lithium-Polymer Akku (kurz LiPo) versorgt mit ca. 12V (3 Zellen) bzw. ca. 16V (4 Zellen) direkt die vier BICtrler. Da alle übrigen Komponenten diese hohe Spannung nicht vertragen, ist eine Spannungsumwandlung erforderlich. Es kommen die Spannungsregler JETImodel SBEC der Firma Hacker [115], der UBEC-3A der Firma Hobbywing [116] und der TMR 0510 von Traco Power [117] zum Einsatz. Der SBEC kann bis zu 6A Dauerstrom liefern. Soviel Strom ist auch erforderlich, da der PC beim Booten und unter Last ca. 5A zieht. Andere Spannungsregler wurden getestet und erwiesen sich als ungeeignet. Zwei 5V Spannungsregler werden verwendet, um den PC von den übrigen Komponenten zu entkoppeln und somit Störungen der anderen Komponenten zu reduzieren. Liegen die übrigen Komponenten an derselben 5V-Spannungsversorgung, so führt dies zu Problemen, die sich meist in Form von I<sup>2</sup>C-Fehlern äußern. Am UBEC angebunden sind alle sonstigen 5V Komponenten. Dazu gehören die Infrarotsensoren, die Ultraschallsensoren, der GPS Receiver, der optische Flusssensor sowie die Servo-Motoren. Zu den 3V Komponenten, die vom TMR mit Spannung versorgt werden, gehören die IMUs, der RC-DSM Receiver, der Luftdrucksensor sowie das Bluetooth-Modul. Die ADC-I<sup>2</sup>C-Level-Konverter verwenden ebenfalls 3V als Referenz- und Eingangsspannung.

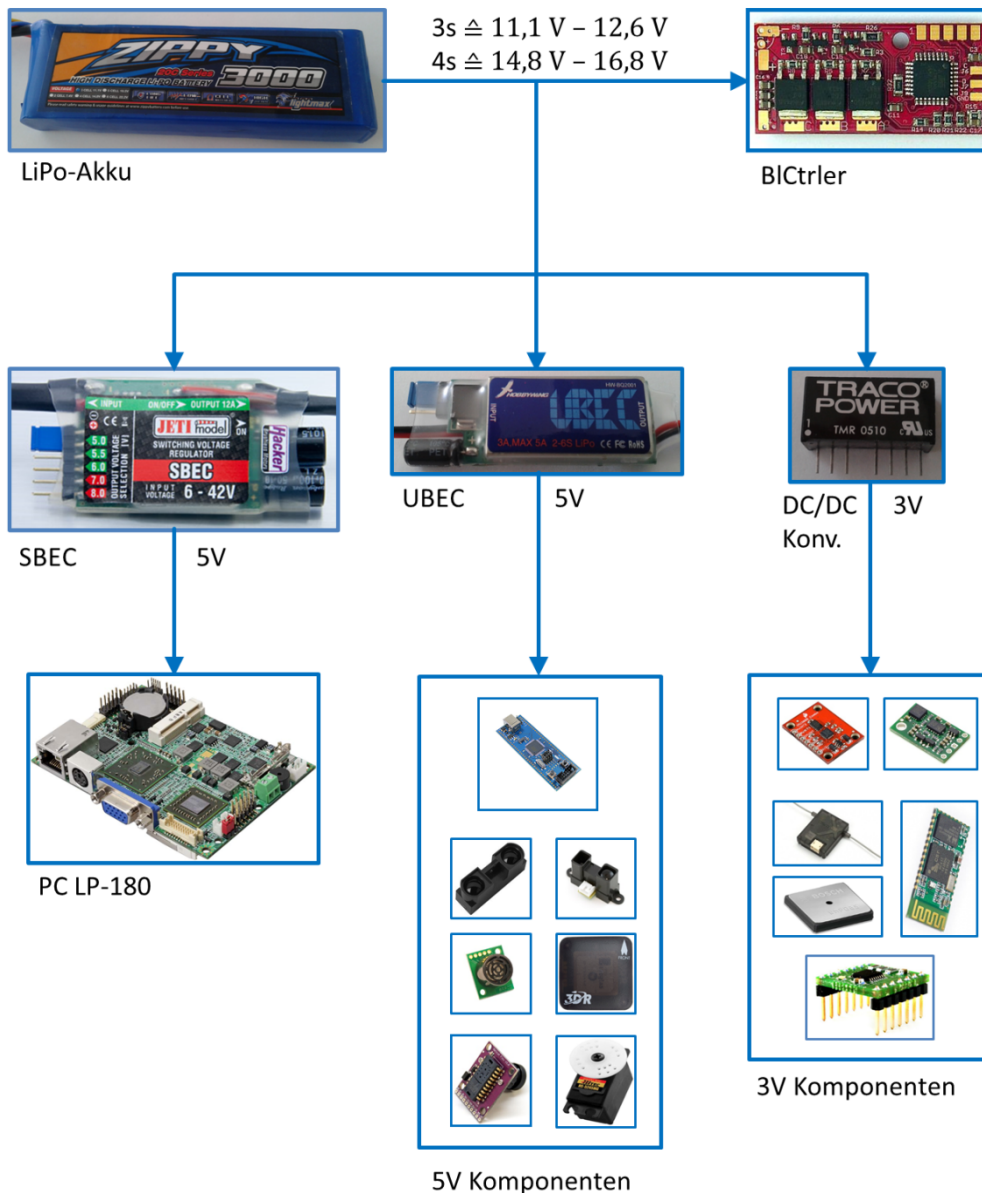


Abbildung 4.2: Elektrische Verschaltung

Da einige I<sup>2</sup>C-Slaves wie die US-Sensoren und die BICtrler auf 5V betrieben werden, wird zwischen diesen Komponenten und den übrigen I<sup>2</sup>C-Slaves sowie dem Master ein Level-Konverter geschaltet. Der Konverter dient hauptsächlich dem Schutz der Komponenten und der Verbesserung der Signalqualität auf dem I<sup>2</sup>C-Bus, womit I<sup>2</sup>C-Fehlern vorgebeugt wird. Das System kann grundsätzlich auch ohne Level-Konverter betrieben werden. Alle USB-Komponenten werden über USB, d.h. über den LP-180, mit Strom versorgt.

## 4.3 Mechanische Konstruktion

### 4.3.1 Mechanisches Design

Das mechanische Design des AQopterI8 (Foto: Abbildung 4.3) gliedert sich in die folgenden 9 Ebenen (vgl. Abbildung 4.4):

- Ebene 1 (oben) mit 8 kurzen IR Sensoren (OD IRS Sensorik)
- Ebene 2 mit dem AVR 32bit Mikrocontroller UC3A0512 und dem LP-180 PICO ITX PC
- Ebene 3 mit der IMU sowie je 4 x BICtrler, Ausleger und Motor
- Ebene 4 mit dem Akku und der Akkuhalterung
- Ebene 5 mit zwei C920 Kameras für SV, der PMD Kamera nano und 2 langen IR Sensoren
- Ebene 6 mit 6 langen IR Sensoren (OD IRL Sensorik)
- Ebene 7 mit 12 US Sensoren (OD US Sensorik)
- Ebene 8 mit der Bodenplatte für die Sensorik nach unten (Höhenbestimmung, OF, OS, FT)
- Ebene 9 (unten) mit dem Landegestell und den Landekufen

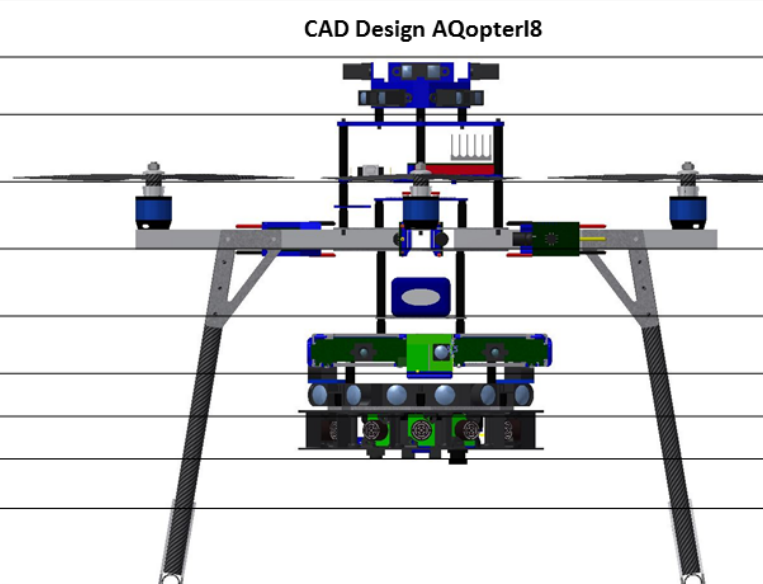
CAD Design AQopterI8	Design Ebenen
	1) OD IRS Sensorik
	2) UC3A, LP 180
	3a) IMU 3b) BICtrl, Ausleger, Motoren
	4) Akku
	5) SV-Kameras, PMD Kamera
	6) OD IRL Sensorik
	7) OD US Sensorik
	8) Bodenplatte
	9) Landegestell

Abbildung 4.4: Frontansicht des AQopterI8 mit 9 Ebenen, CAD Design (CAD Zeichnung Sascha Dechend)

In Ebene 1 befinden sich die 8 kurzen IR Sensoren. Da die OD IR Sensoren einander beeinflussen, sind diese auf Ebene 1 und 6, räumlich voneinander getrennt, angebracht. Mikrocontroller und PC befinden sich auf Ebene 2 relativ weit oben, damit die Anschlüsse gut erreichbar sind. Gleichzeitig ist der LP 180 auf Ebene 2 ausreichend geschützt vor Schäden bei Abstürzen. Zudem erfährt der LP 180 auf Ebene 2 die erforderliche Belüftung für die Kühlung. Die Ebene 3 gliedert sich in zwei Bereiche. Etwas erhöht im Bereich (a) ist die IMU auf Schwingungsdämpfern gelagert. Die IMU befindet sich oberhalb der Hauptstromleitungen, die vom Akku über die BICtrler zu den Motoren führt, um Übersprechen und Störungen auf dem I<sup>2</sup>C-Bus zu reduzieren. Sie befindet sich aber gleichzeitig nahe dem Zentrum, um den Einfluss der Zentrifugalkraft auf die Sensormessungen bei Rotationen des Quadropters zu reduzieren. Im Bereich (b) befinden sich die vier Ausleger mit je einem BICtrler und Motor.

Der Akku inklusive der dazugehörigen Akkuhalterung bildet Ebene 4. Ebene 5 besteht aus den beiden C920 Kameras zur stereo-optischen Abstandsmessung sowie der PMD Kamera nano. Aus Platzgründen befinden sich zwei der 8 OD IRL Sensoren ebenfalls in Ebene 5. Dies ist nicht in der Frontansicht zu sehen, wird aber aus Abbildung 4.5 (links) und Abbildung 4.6 (Seitenansicht) ersichtlich. In Ebene 6 befinden sich die übrigen 6 langen IR Sensoren. Darunter, in Ebene 7, sind die 12 US Sensoren angebracht. Den Abschluss der 7. Ebene, auch Ebene 8 genannt, bildet die Bodenplatte (Abbildung 4.5 rechts). Auf der Bodenplatte sind alle nach unten gerichteten Sensoren montiert. Dazu gehören ein langer und ein kurzer IR Sensor, ein US Sensor, der OF Sensor sowie die beiden Kameras C920 und C270 für das Fourier Tracking (FT) und die Objektsuche (OS). Ebene 9 besteht aus den Landekufen und dem Landegestell, dass hinauf bis Ebene 3 reicht.

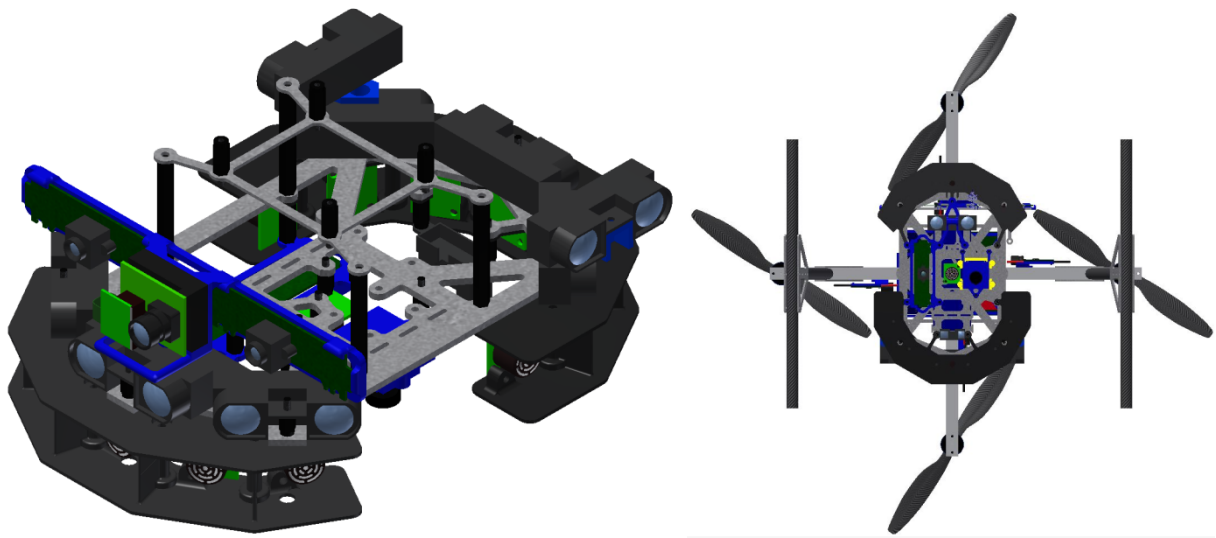


Abbildung 4.5: Ebenen 5-7 (links) und Bodenansicht auf Bodenplatte (rechts) des AQopterI8, CAD Design (CAD Zeichnung Sascha Dechend)

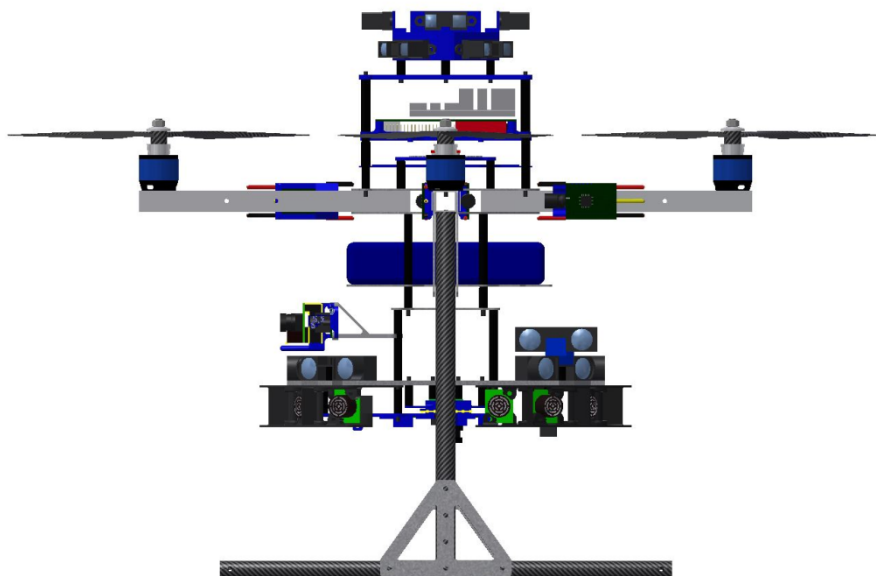


Abbildung 4.6: Seitenansicht des AQopterI8, CAD Design (CAD Zeichnung Sascha Dechend)

Das mechanische Design des AQopterI8 wurde von Sascha Dechend entsprechend der fortlaufend

besprochenen und diskutierten Anforderungen mit einem CAD Programm erstellt und mit Hilfe eines 3D Druckers bzw. einer CNC-Fräse realisiert.

Alternativ zur Verwendung der OD IR Sensoren kann der Laserscanner RP Lidar auf Ebene 1 angebracht werden. In diesem Fall wird auf alle OD IR Sensoren, auch auf die IRL Sensoren in Ebene 5 und 6, verzichtet. Dies ist damit zu erklären, dass beide Sensoren infrarotes Licht zur Abstandsmessung verwenden und durch simultane Verwendung kein komplementärer technologischer Vorteil entsteht. Vielmehr stören sich die Sensoren.

### 4.3.2 Design mit Manipulator

Im Rahmen des vom IHK Würzburg-Schweinfurt mit dem Forschungsförderpreis ausgezeichneten Projektes „Rettungshelfer mit Propellern“ wurde der Quadrocopter um einen Manipulator erweitert, damit dieser in seine Umgebung eingreifen kann (Abbildung 4.7). Die Arbeiten dazu sind noch nicht abgeschlossen. Diese Thematik würde zudem den Rahmen dieser Arbeit sprengen. Dennoch soll hier kurz der Vollständigkeit halber auf die bisherigen Arbeiten zu diesem Thema eingegangen werden.

Zu dem Thema „Erweiterung des Quadrocopters um einen Manipulator“ stellen sich zunächst zwei grundsätzliche Fragen: Wie soll der Manipulator aussehen und wo soll er angebracht werden? Grundsätzlich kann ein Manipulator (Roboterarm) beliebig komplex sein. Je komplexer, größer und schwerer, desto mehr Probleme ergeben sich bei der Umsetzung, denn Gewicht und Größe sind zwei entscheidende Nebenbedingungen. Wird der Roboterarm zu schwer oder groß, so hebt der Quadrocopter nicht mehr ab oder lässt sich schlecht regeln.

Auch soll die Lösung möglichst kompatibel zum bisherigen System sein und seine Größe darf auch nicht stark zunehmen, da das System dann nicht mehr durch eine Tür passen würde

Außerdem lässt sich festhalten, dass sich ein Quadrocopter bereits sehr agil in der Luft bewegen kann. Somit muss der Manipulator nicht von sich aus alle Freiheitsgrade liefern, da diese auch der Quadrocopter erreichen kann. Aufgrund der genannten Überlegungen wurde ein Manipulator mit zwei Gelenken (Ellenbogengelenk, Greifgelenk) als erster Prototyp realisiert, wobei der Aufbau so gehalten ist, dass Änderungen wie das Hinzufügen weiterer Gelenke einfach möglich ist, sollte dies erforderlich sein.

Die zweite Frage nach dem Ort der Anbringung hängt auch stark mit dem bisherigen Design zusammen. Eine Befestigung, die kompatibel dazu ist, ist natürlich von Vorteil. Deshalb wird der Manipulator nicht unterhalb des Quadrocopters befestigt, da sich dies schlecht mit den zum Boden gerichteten Sensoren vereinbaren lässt. Auch ist es sehr unpraktisch, wenn der Quadrotor nur Gegenstände manipulieren kann, über denen er fliegt oder wenn der Arm sehr lang und damit



Abbildung 4.7: AQuadrocopter mit Manipulator



schwer wird. Dabei sei zu bedenken, dass der Propellerkreis ein Hindernis darstellt, quasi ein Totbereich, der den Quadrokoopter mit Manipulator in seinem Arbeitsbereich einschränkt. Ein Manipulator sollte über diesen Bereich hinausreichen. Andernfalls könnte er nicht Mal ohne Weiteres einen Lichtschalter betätigen. Daher wird der Manipulator außen montiert. Es wurden drei Optionen realisiert, um deren Vor- und Nachteile diskutieren zu können.

Das Konzept sieht zudem vor, auf beiden Seiten einen Manipulator anzubringen, damit der Schwerpunkt mittig liegt. Flugtests haben aber bereits gezeigt, dass dies nicht unbedingt notwendig ist, da es genügt, wenn der entsprechende Motor stärker arbeitet. Somit ist das Ungleichgewicht bis zu einem bestimmten Maße kompensierbar. Die notwendige Extrakraft kann entweder fest einprogrammiert sein oder stellt sich dynamisch über den Integralteil des Reglers ein.

Zwei Manipulatoren sind aber auch noch aus einem anderen Grund sinnvoll. Dieser Grund besteht im Bearbeiten und Lösen komplexer und kooperativer Aufgaben. Darunter fällt unter anderem das Bilden einer Kette, womit die Quadrokoopter im Schwarm gemeinsam z.B. eine Person aus dem Feuer ziehen könnten (vgl. Abbildung auf Seite 185). Dazu sind zwei Manipulatoren erforderlich. Dies ist aber nur ein denkbare komplexes Manöver. Auch andere Arbeiten, bei denen z.B. mit dem einen Arm etwas gehalten und mit dem anderen Arm manipuliert, z.B. geschraubt wird, werden mit zwei Armen möglich. Der Mensch verfügt auch über zwei Arme und Hände. Daher ist dies eine sehr naheliegende Lösung, die weitreichendes zusätzliches Potential bietet.

### Befestigung an einem Ausleger

Bei dieser Designalternative wird der Manipulator an einem Ausleger befestigt (vgl. Abbildung 4.8). Zur Realisierung dieser Lösung sind die geringsten Änderungen am bisherigen Design erforderlich. Gleichzeitig reicht der Arm weit aus dem Propellerkreis heraus. Das Design eignet sich besonders für den Flug in Plus-Richtung, wobei der Quadrokoopter mit seinen Auslegern ein Plus bildet und ein Ausleger nach vorne zeigt.



Abbildung 4.8: Manipulator am Ausleger

Nachteil dieser Variante ist die Tatsache, dass das gesamte Gewicht des Manipulators inklusive der Nutzlast von einem Propeller zu tragen ist. Ein weiterer Nachteil sind die starren Beine, die den Quadrokoopter beim Manipulieren behindern könnten.

### Befestigung am beweglichen Bein

Bei der zweiten Designalternative bilden der Manipulator und das Bein bzw. Landegestell eine Einheit. Dies bedeutet der Manipulator ist nicht mehr nur Manipulator, sondern gleichzeitig Manipulator und Landegestell. Der Vorteil besteht darin, dass das Gewicht für das Bein wegfällt und gleichzeitig die Beine beim Manipulieren nicht mehr behindern können. Ein weiterer Vorteil ist, dass die Beine beweglich sind.



Abbildung 4.9: Manipulator am Bein

Zu den Nachteilen gehört wie beim ersten Design der Umstand, dass der Manipulator unterhalb eines Motors angebracht ist. Dabei wird der Abwind des Propellers durch

den Manipulator bzw. dessen Nutzlast gestört, wodurch Auftrieb verloren geht. Zudem kann der Manipulator bei der Landung beschädigt werden und es ist eine Vorrichtung notwendig, die solche Schäden abfängt. Außerdem fehlt für Situationen, bei denen mit voller Last gelandet werden soll, ein von dieser Last getrenntes Landegestell zur Landung.

### Befestigung zwischen Auslegern

Diese Lösung sieht vor, dass der Manipulator zwischen zwei Motoren befestigt wird und eignet sich besonders für den Flug in X-Richtung, wobei die Mitte zwischen zwei Auslegern nach vorne zeigt. Das Design lässt sich noch verfeinern, indem auf die diagonalen Ausleger verzichtet wird und stattdessen ein H-Design Anwendung findet.

Ein Vorteil dieses Designs ist, dass zwei Motoren den Manipulator und dessen Gewicht tragen. Somit steht hier prinzipiell mehr Kraft zur Verfügung. Jedoch setzt die Kraft der Motoren nicht direkt unterhalb des Manipulators an, wodurch sich dieser Effekt auch zum Teil wieder aufhebt. Ein weiterer Vorteil besteht darin, dass sich der Manipulator nicht unterhalb der Propeller befindet und dadurch der

Propellerabwind durch Manipulator und Nutzlast nicht behindert wird. Gleichzeitig benötigt dieses Design mehr Material, wodurch schon von vorneherein das Gesamtgewicht steigt.



Abbildung 4.10: Manipulator zwischen Auslegern

### Realisierung der Manipulatoren

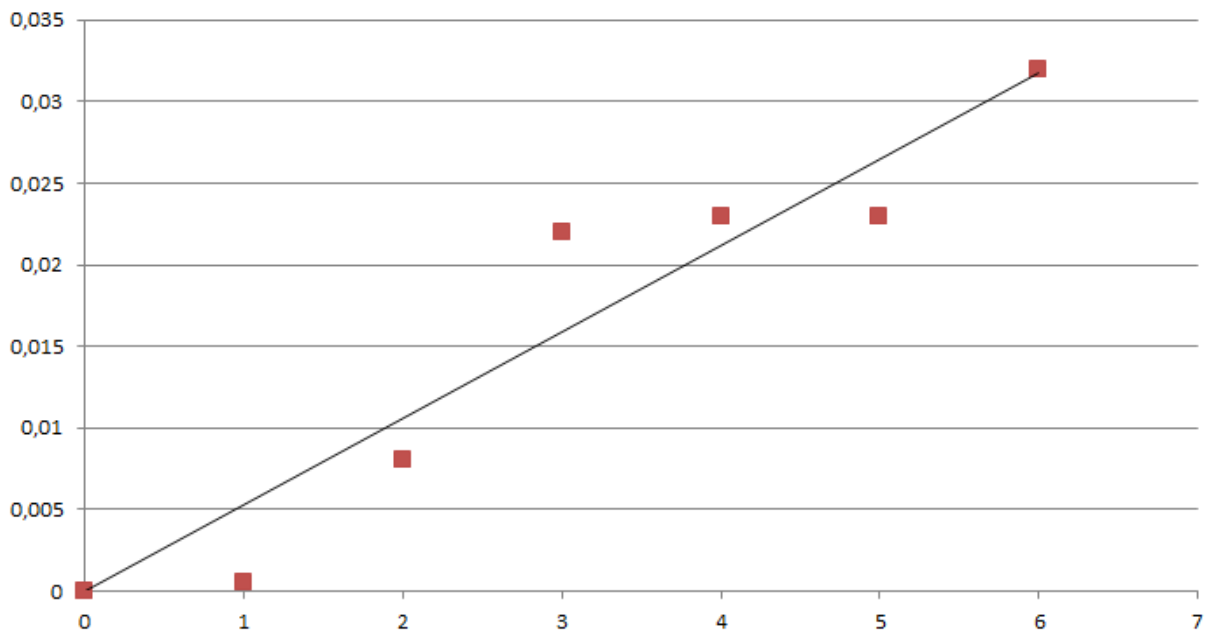
Die Realisierung der Quadrocopter mit Manipulator bestehend aus CAD Zeichnung, Fertigung der mechanischen Teile und Zusammenbau wurde von Studenten durchgeführt, die mit Hilfe der Mittel aus dem Förderpreis bezahlt werden konnten. Für den Quadrocopter aus Abbildung 4.8 war Robin Leblebidci verantwortlich. Für Manipulator und Befestigung aus Abbildung 4.9 war Junaidh Fareedh und für Manipulator und Befestigung aus Abbildung 4.10 war Lekha Sumathy zuständig.





*We are all aware that the senses can be deceived, the eyes fooled.  
But how can we be sure our senses are not being deceived at any particular time,  
or even all the time?  
Might I just be a brain in a tank somewhere,  
tricked all my life into believing in the events of this world  
by some insane computer? And does my life gain or lose meaning  
based on my reaction to such solipsism.  
[unbekannt, Alpha Centauri]*

## 5 Signalverarbeitung (Sensorik)



## 5.1 Überblick Signalverarbeitung

Aufgabe der Signalverarbeitung ist das Auswerten der Sensorik. Die Signalverarbeitung gliedert sich nach dem Zweck der Verarbeitung der Messungen in Orientierungsbestimmung, Höhenbestimmung, Positionsbestimmung sowie Hinderniserkennung und Situationserfassung.

## 5.2 Orientierungsbestimmung

Unter Orientierungsbestimmung ist die Erfassung der drei rotatorischen Freiheitsgrade zu verstehen. Mit der Entscheidung, ein eigenes System aufzubauen, ist auch die Implementierung einer Orientierungsbestimmung notwendig. Als Herz des Systems ist eine Lösung zur Orientierungsbestimmung gleichzeitig die erste Anforderung und Grundvoraussetzung für alle weiteren Arbeiten. Gleichzeitig gibt es zu diesem Thema bereits Lösungen, diese erwiesen sich aber als ungeeignet, da sie für die Anwendung nicht konzipiert, zu ungenau, unflexibel im Design oder zu aufwendig (Implementierung, Rechenaufwand) sind oder aus anderen Gründen nicht zu den gewünschten Ergebnissen führten. Für einen autonomen Quadropter ist die Orientierungsbestimmung von besonderer Bedeutung, da Orientierungsfehler nicht durch einen Piloten korrigiert werden und kleine Fehler in der Orientierung bereits zu Positionsänderungen führen. Gleichzeitig darf die Lösung nicht viel Rechenzeit beanspruchen und sollte nicht auf einen bestimmten Sensor mit Ausnahme des Gyroskops angewiesen sein. Es sei angemerkt, dass zu Beginn der Arbeit keine 9 DOF IMUs kostengünstig verfügbar waren. Aufgrund all dieser Überlegungen wurde eine eigene Lösung entwickelt, implementiert und mit der Zeit in mehreren Schritten verbessert. Obwohl die Problematik entscheidenden Einfluss auf das Gesamtsystem hat und somit bedeutsam ist, lag der wissenschaftliche Fokus nicht auf diesem Thema. Dies hängt auch damit zusammen, dass diese Fragestellung aus wissenschaftlicher Sicht bereits weitestgehend behandelt ist. Es existieren Doktorarbeiten, die sich nur mit dem Thema Orientierungsbestimmung beschäftigen.

Zur Orientierungsbestimmung gehören vier Teile:

- Inertiale Sensorik und Vorverarbeitung
- Kalibrierung
- Konditionierung
- Berechnung der Orientierung (Datenfusion / Filter)

Im Weiteren werden diese vier Teile näher beschrieben. Zur Berechnung der Orientierung sind unterschiedliche Ansätze implementiert, die im darauffolgenden näher erläutert werden (Kapitel 5.2.4 bis 5.2.8).

### 5.2.1 Inertiale Sensorik und Vorverarbeitung

Das Auslesen der Sensorwerte geschieht beim AQopter18 via I<sup>2</sup>C. I<sup>2</sup>C bietet aber keine Fehlererkennung, womit nicht sichergestellt ist, dass die Messwerte bei der Übertragung nicht verfälscht werden. Experimente haben gezeigt, dass dies bei dem verwendeten und vergleichsweise großen I<sup>2</sup>C-Bus durchaus vorkommt. Deshalb werden die IMU Daten zweifach ausgelesen. Der aktuelle Messwert wird dann aus den beiden aktuellen Messwerten und dem letzten Messwert durch Medianbildung ermittelt.

Die Messungen des Gyroskops sind besonders sensibel, da Fehler zunächst im System verweilen und erst durch Kompensation mittels Accelerometer entfernt werden können. Da der Accelerometer aber

systembedingt nur mit geringem Gewicht eingeht, dauert die Kompensation von Fehlern in der Winkelgeschwindigkeit, weshalb große Fehler nach Möglichkeit zu vermeiden sind. Beim Accelerometer und Magnetometer, die quasi absolut messen, ist dies kein so großes Problem. Deshalb wird für das Gyroskop ein eigens entwickelter Filter, der sogenannte Ausreißer-Filter, verwendet, wobei weiterhin jede Messung zweifach ausgelesen wird. Zur Bestimmung des aktuell gültigen Messwerts werden die beiden aktuellen und die beiden vorherigen Messungen herangezogen. Mit Hilfe dieser vier Messungen werden Ausreißer erkannt und ausgefiltert.

Dazu werden zunächst die beiden aktuellen Messwerte  $\omega_{1,t}$  und  $\omega_{2,t}$  zum Zeitpunkt  $t$  miteinander verglichen (Abbildung 5.1). Theoretisch, wenn kein Fehler auftritt, sollten diese Messungen identisch sein, da sie dieselbe Größe zum selben Zeitpunkt beinhalten. Die Messungen können dabei vom selben Sensor, wenn nur ein Sensor zur Verfügung steht, oder von zwei Gyroskopen stammen. Liegt die Differenz beider Messungen unter der geeignet gewählten Schwelle  $S_a$ , so ist das Ergebnis der Mittelwert beider Messungen. Andernfalls muss davon ausgegangen werden, dass einer der beiden Werte ungültig ist. Im Weiteren geht es darum, festzustellen, welcher der beiden neuen Messwerte ungültig ist.

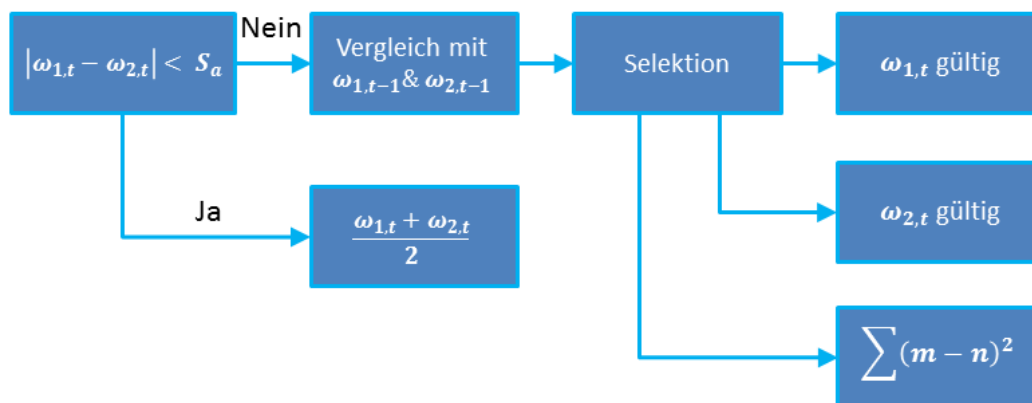


Abbildung 5.1: Konzept Ausreißer-Filter

Dazu werden die beiden aktuellen Messungen mit den beiden alten Messungen  $\omega_{1,t-1}$  und  $\omega_{2,t-1}$  verglichen. Dazu wird gezählt, wie oft die Differenz je einer neuen Messung und einer alten Messung oberhalb der Schranke  $S_a$  liegt. Die Messung  $\omega_{i,t}$  wird genommen, bei der die Schranke seltener überschritten wird, sofern dies nicht öfters als einmal der Fall ist. Der Index  $i$  gibt dabei die ID der Messung an und kann 1 für Messung 1 und 2 für Messung 2 annehmen.  $t$  ist der Zeitpunkt der Messung.

Falls die Schranke öfters als einmal überschritten wird, muss davon ausgegangen werden, dass mehrere Messungen fehlerhaft sind. In diesem Fall wird aus allen vier vorhandenen Messungen die Messung verwendet, die zu allen anderen Messungen in Summe die geringste quadratische Differenz aufweist. Die Summe der quadratischen Differenzen als Auswahlkriterium wird in Abbildung 5.1 mit  $\sum(m - n)^2$  veranschaulicht.

Dieses Verfahren besitzt den Vorteil, dass in den meisten Fällen die aufwändige Berechnung der Summe der quadratischen Differenzen nicht erforderlich ist und dies nur vorgenommen wird, wenn festgestellt wurde, dass wahrscheinlich zwei Messungen fehlerhaft sind. Zudem kann in den meisten

Fällen das Rauschen durch Mittelwertbildung zweier Messungen reduziert werden. Somit passt sich der Ausreißer Filter dynamisch der Situation an.

Darüber hinaus gehört zur Sensorik auch die automatische IMU Auswahl. In Software kann die Priorität aller unterstützter IMUs eingestellt werden und bei Systemstart wird automatisch die IMU mit der höchsten Priorität verwendet. Aktuell werden die 6 in Kapitel 2.3.3 beschriebenen IMUs unterstützt. Dabei kann bei der Einstellung der Prioritäten zwischen Accelerometer und Gyroskop unterschieden werden. Werden die IMU3000 und die MinIMU-9 gleichzeitig angeschlossen, so kann das Gyroskop von der IMU3000 und das Accelerometer der MinIMU-9 genutzt werden. Dies ist auch gleichzeitig die empfohlene Standardeinstellung.

## 5.2.2 Kalibrierung

Unter Kalibrierung wird die Bestimmung von Offset  $o$ , auch Bias genannt, und Skalierungsfaktor  $s$  verstanden, so dass mit Hilfe von Formel 5.1 aus dem Messwert  $m$  der kalibrierte Wert  $k$  bestimmt werden kann. Im Folgenden wird die Kalibrierung der IMU bestehend aus Accelerometer, Gyroskop und Magnetometer behandelt.

$$k = s \cdot m - o$$

Formel 5.1

### Kalibrierung des Gyroskops

Bei Systemstart wird das Offset des Gyroskops bestimmt, indem der Mittelwert aus 2000 Messwerten berechnet wird. Deshalb ist es erforderlich, dass der Quadrocopter bei Systemstart ruhig steht, da sonst die Kalibrierung fehlschlägt. Der Skalierungsfaktor ist fest einprogrammiert und wurde an Hand einer manuellen 360° Drehung bestimmt.

### Kalibrierung des Accelerometers

Zur Kalibrierung des Accelerometers wurden zwei Methoden implementiert: Die dynamische Kalibrierung und die statische Kalibrierung.

Bei der dynamischen Kalibrierung wird das Bias bei Systemstart wie beim Gyroskop aus 2000 Messwerten bestimmt. Voraussetzung dazu ist, dass sich der Quadrocopter in der Horizontalen befindet, so dass angenommen werden kann, dass der Gravitationsvektor vollständig mit der z-Achse des Sensors übereinstimmt. Als Hilfsinstrument kommt dabei eine Libelle zum Einsatz. Der Skalierungsfaktor kann hierbei nicht bestimmt werden.

Alternativ dazu verwendet die statische Kalibrierung das Min-Max-Verfahren (Minimum und Maximum). Dazu wird die Achse, die kalibriert werden soll, mit dem Gravitationsvektor in Deckung gebracht. Durch einmaliges Betätigen des Kalibrierungsknopfs wird die Kalibrierung aktiviert und es werden 2000 Messwerte aufgenommen. Anschließend ist um 180° zu drehen und erneut zu drücken, so dass die Achse nun den invertierten Gravitationsvektor misst. Aus den Messungen werden Offset und Skalierung automatisch berechnet und im Flash gespeichert. Die Werte können beim nächsten Systemstart geladen werden. Zur einfachen Durchführung der Kalibrierung werden per USART entsprechende Anweisungen ausgegeben.

Die kalibrierte Achse wird dabei automatisch erkannt. Dies funktioniert darüber, dass die Achse, die den Gravitationsvektor in Linie und invertiert gemessen hat, die höchste betragliche Differenz aus



beiden Messungen aufweist. Diese Differenz entspricht gleichzeitig der zweifachen Skalierung. Mit Formel 5.2 lässt sich nun aus dem Skalierungsfaktor und den beiden Messungen  $m_1$  und  $m_2$  das Offset bestimmen.

$$o = \min(m_1, m_2) + s \quad \text{Formel 5.2}$$

Vorteil der dynamischen Kalibrierung ist die Einfachheit und der geringe Aufwand. Es muss nicht vorher manuell eine Kalibrierung durchgeführt werden. Zudem fängt sie Biasfehler ab, die dadurch entstehen können, dass sich der Bias der Sensoren mit der Zeit und dem Ein- und Ausschalten ändert. Dies war insbesondere bei älteren Sensoren ein Problem. Nachteil ist, dass das System bei der Kalibrierung horizontal ausgerichtet sein muss. Deshalb ist die statische Kalibrierung meist genauer, sofern diese gewissenhaft durchgeführt wird. Die statische Kalibrierung kann zudem für jede Achse einen eigenen Skalierungsfaktor bestimmen.

### Kalibrierung des Magnetometers

Zur Kalibrierung des Magnetometers wurden wiederum zwei Methoden implementiert: Das Min-Max-Verfahren sowie die 360° Drehung. Das Min-Max-Verfahren läuft simultan zur Kalibrierung des Accelerometers ab. Die Kalibrierung mittels 360° Drehung wird durchgeführt, wenn der Kalibrierungsknopf zweimal schnell hintereinander gedrückt wird.

Die 360° Kalibrierung besteht aus zwei Schritten. Zunächst werden die x-Achse und die y-Achse kalibriert, indem 360° um die z-Achse gedreht wird. Im Anschluss wird 360° um die x- oder um die y-Achse gedreht, um die z-Achse zu kalibrieren. Dabei werden jeweils solange Messungen der kalibrierenden Achse aufgenommen, bis erneut gedrückt wurde. Zur Hilfe werden wiederum Anweisungen per USART ausgegeben.

Aus den vorgenommenen Messungen wird für die jeweilige Achse stets nur das Minimum  $m$  sowie das Maximum  $n$  gespeichert. Mit Formel 5.3 lässt sich dann das Offset bestimmen.

$$o = \frac{m + n}{2} \quad \text{Formel 5.3}$$

Dieses Verfahren ist genauer als das Min-Max-Verfahren, aber gleichzeitig aufwändiger.

### 5.2.3 Konditionierung

Unter Konditionierung werden die folgenden Schritte zum Aufbereiten der aktuellen Messwerte verstanden, bevor diese an die Orientierungsbestimmung weitergereicht werden:

- Das Verrechnen der Kalibrierungsdaten bestehend aus Offset und Skalierungsfaktor
- Das Erstellen und Normalisieren von Gravitations- und Magnetfeldvektor
- Die Bestimmung der infinitesimalen Drehung aus den Winkelgeschwindigkeiten des Gyroskops
- Die Anwendung eines Digitalen Tiefpassfilters (Bessel-Filter) sowie eines Mittelwertfilters auf die Accelerometermessungen
- Die Saturierung der Accelerometermessungen mit anschließender Bestimmung des Winkels

über den Sinus

#### 5.2.4 Eindimensionale Integration und Quaternionen

Die einfachste Lösung zur Orientierungsberechnung ist die eindimensionale Integration, d.h. die durch ein Gyroskop gemessenen Winkelgeschwindigkeiten aller drei Achsen werden einzeln integriert (Formel 5.4, Formel 5.5 und Formel 5.14). Mit dieser Form der Orientierungsbestimmung kann ein Quadrokopter bereits stabil im Flug geregelt werden. Dazu ist nicht mal eine Skalierung der Messung in Grad erforderlich, da dies für den Regler nichts weiter darstellt als ein Faktor auf seine Reglerparameter. Diese Art der Berechnung wird jedoch nach kurzer Zeit sehr ungenau und zwar dadurch, dass sich der Quadrokopter nicht immer nur um eine Achse gleichzeitig dreht und somit die Reihenfolge der Drehungen eine Rolle spielt. In der Praxis bedeutet dies, wenn über mehrere Achsen gleichzeitig gedreht wird, kommt es zu Fehlern. Dies lässt sich im Flug nicht vermeiden und führt nach kurzer Zeit zu einem erheblichen Fehler im Winkel. Wird jedoch nur die Winkelgeschwindigkeit geregelt und der Quadrokopter manuell gesteuert, ist dies eher weniger ein Problem.

$$\alpha = \int_0^t \omega_x dt \quad \text{Formel 5.4}$$

$$\beta = \int_0^t \omega_y dt \quad \text{Formel 5.5}$$

$$\gamma = \int_0^t \omega_z dt \quad \text{Formel 5.6}$$

Eine deutliche Verbesserung liefert die Verwendung von Quaternionen (Formel 2.5). In diesem Fall wird die Reihenfolge der Drehungen berücksichtigt. Die Orientierung driftet dennoch, da einmal gemachte Fehler nicht korrigiert werden können, wenn nur Winkelgeschwindigkeiten, d.h. Änderungen des Winkels, gemessen und eingebracht werden. Daher wird eine Orientierungsbestimmung allein auf Basis eines Gyroskops selten verwendet.

Die alleinige Verwendung des Gyroskops eignet sich jedoch zur Einstellung des Reglers bzw. Ursachenanalyse bei schlechtem Regelverhalten. Sie stellt die verlässlichste und einfachste Möglichkeit dar, um einen Regler zu erproben, einzustellen und gleichzeitig andere Stabilitätsprobleme auszuschließen, die z.B. aus der Datenfusion herrühren könnten.

#### 5.2.5 Lineares Kalman Filter (KF) ohne Quaternionendrehung

Um das Problem des Driftens der Orientierung (Gyrodrift bzw. Winkeldrift) zu lösen, kann zur Kompensation ein Accelerometer verwendet werden. Das Accelerometer misst Beschleunigungen, darunter den Gravitationsvektor der Erde. Somit lassen sich die Drehwinkel um die x- und y-Achse (roll, pitch) quasi absolut ermitteln. Der so ermittelte Winkel ist jedoch auf Grund von Vibrationen und anderer Störquellen stark verrauscht. Zudem wird nicht nur das Schwerfeld der Erde gemessen, sondern auch die Eigenbewegung des Quadrokopters sowie andere auf den Quadrokopter wirkende Kräfte wie z.B. Wind. Diese Effekte verursachen bei der Bestimmung des Drehwinkels mittels Accelerometer einen Fehler. Daher ist eine geeignete Datenfusion zu verwenden, die beim Gyrometer als Hochpass- und beim Accelerometer als Tiefpassfilter operiert. D.h. kurzfristige, schnelle Änderungen sollen durch das Gyroskop eingehen, während der Winkel langfristig durch das

Accelerometer bestimmt wird. Eine solche Datenfusion ermöglicht ein Kalman Filter.

Das Kalman Filter ist eine weit verbreitete Lösung zur Datenfusion [118] [119] [120] [121] (vgl. auch Kapitel 2.4.2). Es basiert auf der systemtheoretischen Betrachtung mittels Zustandsraumdarstellung. Grundlage der Zustandsraumdarstellung ist das Zustandsraummodell, das mittels Systemgleichung und Messgleichung beschrieben wird. Das Design des Zustandsraummodells ist auch gleichzeitig Grundlage des Kalman Filters.

$$\vec{x}_t = \mathbf{A} \cdot \vec{x}_t + \mathbf{B} \cdot \vec{u}_t + \vec{w}_t \quad \text{Formel 5.7}$$

$$\vec{y}_t = \mathbf{C} \cdot \vec{x}_t + \mathbf{D} \cdot \vec{u}_t + \vec{v}_t \quad \text{Formel 5.8}$$

Die Systemgleichung (Formel 5.7) beschreibt, wie sich der neue Zustand bzw. die Zustandsänderung  $\vec{x}_t$  berechnen lässt mittels Systemmatrix A, auch Übergangsmatrix genannt, dem aktuellen Zustandsvektor  $\vec{x}_t$  zum Zeitpunkt t, der Eingangsmatrix B, dem Eingangsvektor  $\vec{u}_t$  (Eingriff in das System) und dem Prozessrauschen  $\vec{w}_t$ . Der Messvektor  $\vec{y}_t$ , auch Ausgangsvektor genannt, beinhaltet die Messungen. Die Messgleichung (Formel 5.8) beschreibt, wie sich der Messvektor aus der Ausgangsmatrix C, dem Zustandsvektor, der Durchgangsmatrix D, dem Eingangsvektor und dem Messrauschen  $\vec{v}_t$  zusammensetzt. Der Kalman Filter geht dabei davon aus, dass das Prozess- und Messrauschen durch die Kovarianzen Q und R beschrieben werden kann.

Eine einfache Realisierung ist die Verwendung eines linearen Kalman Filters für die beiden Achsen Roll und Pitch. Für den AQopter18 wurde das Filter-Design aus Formel 5.9 implementiert.

$$\mathbf{A} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} P_{Acc} & 0 \\ 0 & P_{Gyro} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} M_{Acc} & 0 \\ 0 & M_{Gyro} \end{bmatrix} \quad \text{Formel 5.9}$$

$$\vec{x} = \vec{y} = \begin{bmatrix} \varphi \\ \omega \end{bmatrix}, \quad M_{Acc} = a^2 + b + c^2$$

Bei diesem Ansatz werden der Winkel  $\varphi$  mit Hilfe des Accelerometers und die Winkelgeschwindigkeit  $\omega$  mit Hilfe des Gyroskops gemessen und beide miteinander fusioniert. Das Prozessrauschen Q und Teile des Messrauschens R werden als konstant angesehen, wobei das Messrauschen  $M_{Acc}$  des Winkels mit Hilfe der Historie der Beschleunigung  $a$ , dem Betrag der Beschleunigung  $b$  und dem Fehler des Reglers  $c$  dynamisch aktualisiert wird. Mit diesen Anteilen werden dynamische Bewegungen, die zu einem Fehler bei der Gravitationsmessung führen, geringer gewichtet. Die Datenfusion funktioniert am Boden gut. Hier dreht sich der Gravitationsvektor aufgrund der vorhandenen Gegenkräfte entsprechend des Winkels. Im freien Flug dagegen zeigt sich, dass sich der Winkel anders verhält und sich nicht analog zum Boden bestimmen lässt. Dies liegt daran, dass aufgrund fehlender Gegenkraft automatisch eine translatorische Beschleunigung (beschleunigte Bewegung) eintritt. Dies verfälscht die Messung des Winkels. Eine korrigierte Berechnung des Winkels wurde nicht realisiert, da eine Unterscheidung zwischen Boden und Flug unpraktisch erschien.

Eine weitere Schwäche des Filters ist die Modellierung als lineares System, womit die in Abschnitt 5.2.4 beschriebenen Probleme auftreten. Stand der Technik ist die Nutzung eines erweiterten Kalman Filters (EKF) zur Lösung dieses Problems. Lukas Iffländer hat in seiner Bachelorarbeit das EKF von Sabbatini für den AQopter18 implementiert [35][A06]. Auch nach Laufzeitoptimierung benötigte der Algorithmus auf dem verwendeten Mikrocontroller UC3A0512 noch ca. 10ms, während der lineare Kalman Filter auf ca. 1-2ms kommt. Deshalb wird kein EKF verwendet.

### 5.2.6 Lineares Kalman Filter mit Quaternionendrehung

Eine deutliche Verbesserung des Linearisierungsproblems lässt sich durch die sogenannte Quaternionendrehung erreichen (Abbildung 5.2). Die Winkel des Accelerometers werden in raumfesten Koordinaten gemessen, die Winkelgeschwindigkeiten jedoch in Körperfesten. Dieses Problem lässt sich durch Transformation der gemessenen Winkelgeschwindigkeiten lösen. Dazu werden die aktuellen Winkelgeschwindigkeiten zunächst in ein Rotationsquaternion umgewandelt und per Quaternionenmultiplikation mit dem Orientierungsquaternion verrechnet (Formel 2.5).



Abbildung 5.2: Konzept Quaternionendrehung

Mit Formel 2.9 werden aus dem Quaternion die Eulerwinkel bestimmt. Die Winkelgeschwindigkeiten bzw. Winkeländerungen werden dann durch Differenz der aktuellen zu den vorherigen Eulerwinkeln bestimmt, die aus den jeweiligen Quaternionen ermittelt werden. Diese Winkeländerungen gehen dann als Messung in das KF (Formel 5.9) ein, womit die Fehler der Linearisierung deutlich reduziert werden können.

Ein Problem hierbei ist, dass zwar der Drift im Kalman Filter korrigiert wird, diese Korrektur jedoch nicht im Quaternion vorgenommen wird, welches somit weiterhin driftet. Dies führt zu Fehlern, da die Berechnung der Winkeländerungen verfälscht wird.

### 5.2.7 Lineares Kalman Filter mit Quaternionendrehung und Feedback

Das Auseinanderdriften der Orientierungen im Kalman Filter und Orientierungsquaternion lässt sich durch Feedback bzw. Update (Aktualisierung) des Orientierungsquaternions aus der kompensierten Orientierung des Kalman Filters erreichen. Experimente haben gezeigt, dass ein kontinuierliches Feedback zu Stabilitätsproblemen führt. Deshalb wird das Orientierungsquaternion nur alle 500 Durchläufe, d.h. alle 5s aktualisiert. Der Drift in diesem Zeitraum lässt sich vernachlässigen.

Diese Lösung entspricht der aktuellen Implementierung, sofern das Kalman Filter zur Orientierungsbestimmung verwendet wird.

### 5.2.8 Komplementärer Quaternionen Filter (CQF)

Da der beschriebene Ansatz mittels Kalman Filter nicht exakt genug für den autonomen Flug war, wurde mit dem komplementäre Quaternionen Filter (engl.: complementary quaternion filter, kurz: CQF) eine Weiterentwicklung der zuvor erprobten Ansätze unter alleiniger Verwendung von Quaternionen realisiert. Verglichen mit dem linearen Kalman Filter liegt der größte Vorteil darin, dass keinerlei Linearisierungsverluste auftreten und somit gleichzeitige Drehungen um beliebige Achsen stets korrekt abgebildet werden. Des Weiteren kommt diese Orientierungsdarstellung ohne

Verwendung von Eulerwinkeln aus, womit die Probleme der Singularitäten und des Gimbal Locks vermieden werden. Ein Nachteil ist der Verlust der Anschaulichkeit.

Im Vergleich zu anderen gängigen Methoden, wie z.B. einem EKF, sind der Rechenaufwand sowie die Komplexität des CQF gering. Ein besonderer Vorteil dieses Ansatzes ist der Umstand, dass sowohl das Accelerometer als auch das Magnetometer jeweils optional zum Gyrometer genutzt werden kann, um einem Drift entgegenzuwirken, aber die Datenfusion nicht darauf angewiesen ist. Somit können diese Daten beliebig eingepflegt werden. Dies ist insbesondere dann von Vorteil, wenn falsche Messungen des Magnetometers die Orientierung verfälschen.

Eine weitere Besonderheit des Filters liegt darin, dass das Filter zunächst nur das Korrekturquaternion bestimmt und damit das Orientierungsquaternion um seinen Drift und sonstige Fehler kompensiert. Der Biasdrift wird nicht explizit berechnet, sondern befindet sich im Korrekturquaternion. Der Biasdrift könnte aus dem Korrekturquaternion explizit ermittelt werden. Empirische Untersuchungen ergaben, dass dies nicht immer von Vorteil ist, bzw. oftmals zu keiner Verbesserung führt, da sich im Korrekturquaternion je nach Situation auch weitere Fehler als der Biasdrift befinden können. Dies ist ein Vorteil, wenn weitere Fehler im Verhältnis zum Biasfehler groß sind, bei einem dominanten Biasfehler ist dies aber tendenziell von Nachteil. Beim Arbeiten mit dem System zeigte sich, dass sehr unterschiedliche Fehler auftraten. Diese werden z.B. hervorgerufen durch Vibrationen, I<sup>2</sup>C-Fehler und Übersprechen der Leitungen.

### CQF-Algorithmus

Während der Kalibrierung werden zum Zeitpunkt  $t=0$  der initiale Beschleunigungsvektor  $\vec{a}(t=0)$  und Magnetfeldvektor  $\vec{m}(t=0)$  gemessen und abgespeichert (Formel 5.10).

$$\vec{a}(t) = \begin{bmatrix} {}^t a_x \\ {}^t a_y \\ {}^t a_z \end{bmatrix} \quad \vec{m}(t) = \begin{bmatrix} {}^t m_x \\ {}^t m_y \\ {}^t m_z \end{bmatrix} \quad \text{Formel 5.10}$$

Mit Hilfe dieser Vektoren und den aktuellen Messungen kann dann ein Fehler, z.B. aufgrund von Drift, erkannt und korrigiert werden. Dazu werden jeweils die beiden Korrekturquaternionen  $q_a$  und  $q_m$  aus den Beschleunigungsvektoren und aus den Magnetfeldvektoren bestimmt.

Abbildung 5.3 veranschaulicht den Algorithmus für den CQF. Zunächst wird der initiale Beschleunigungsvektor  $\vec{a}(0) = [{}^0 a_x \quad {}^0 a_y \quad {}^0 a_z]^T$  (vgl. Formel 5.10) in die aktuelle Orientierung gedreht. Um Rauschen und Vibrationen weiter zu unterdrücken, werden beide Beschleunigungsvektoren, der transformierte initiale Beschleunigungsvektor und der aktuell gemessene Beschleunigungsvektor  $\vec{a}(t)$  durch einen EWMA-Filter mit einem Cutoff-Faktor von  $10^{-3}$  verarbeitet. Durch Anwendung von Formel 2.11, Formel 2.12 und Formel 2.13 wird aus den beiden Vektoren das Quaternion  $p_a$  bestimmt. Da Drehungen um die z-Achse sich nicht mit Hilfe des Gravitationsvektors erfassen lassen, ist das Quaternion  $p_a$  noch um seine Gier-Komponente zu bereinigen. Dies geschieht durch Anwendung von Formel 5.11. Das Ergebnis ergibt das

Beschleunigungsquaternion  $\underline{q}_a$ .

$$\underline{q}_a = \begin{bmatrix} q_{a0} \\ q_{a1} \\ q_{a2} \\ q_{a3} \end{bmatrix} = \begin{bmatrix} p_{a0} \\ p_{a1} \cdot \frac{1 - (p_{a0})^2}{\sqrt{(p_{a1})^2 + (p_{a2})^2}} \\ p_{a2} \cdot \frac{1 - (p_{a0})^2}{\sqrt{(p_{a1})^2 + (p_{a2})^2}} \\ 0 \end{bmatrix}$$

Formel 5.11:  
Gierbeseitigung

Aus Gründen der besseren Lesbarkeit wurde in Formel 5.11 der Zeitindex t weggelassen.

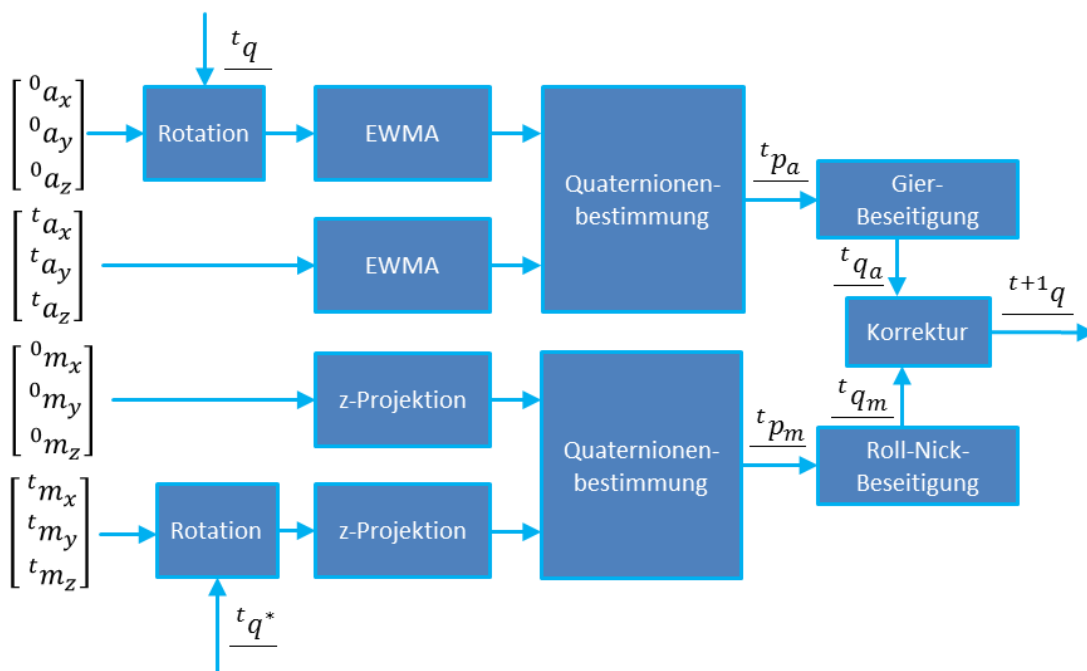


Abbildung 5.3: CQF-Algorithmus

Das Magnetfeldquaternion  $\underline{q}_m$  wird vergleichbar bestimmt. Durch Rotation mit dem konjugiert Komplexen des aktuellen Orientierungsquaternions wird der aktuelle Magnetfeldvektor in die initiale Orientierung transformiert. Anschließend werden die beiden Magnetfeldvektoren auf die z-Ebene projiziert, da der Magnetfeldvektor lediglich zur Gier-Korrektur verwendet werden soll. Nun kann aus den beiden Vektoren das Quaternion  $\underline{p}_m = [p_{m0} \ p_{m1} \ p_{m2} \ p_{m3}]^T$  als der Gierfehler berechnet werden.

Die Magnetfeldvektormessungen sind rotationssymmetrisch um den Magnetvektor, d.h. Drehungen um diese Achse können nicht durch das Magnetfeldquaternion erkannt werden. Dadurch und durch Fehler bei der Messung des Erdmagnetfeldes kann das Quaternion fehlerhafte Komponenten enthalten. Um eine fehlerhafte Kompensation von Roll und Pitch von vornherein auszuschließen, wird die Roll-Nick-Beseitigung durchgeführt. Dazu wird nur die Drehung um die Gierachse weiter verwendet (Formel 5.12). Das Ergebnis ergibt das Magnetfeldquaternion  $\underline{q}_m$ .

$$\underline{q}_m = \begin{bmatrix} q_{m0} \\ q_{m1} \\ q_{m2} \\ q_{m3} \end{bmatrix} = \begin{bmatrix} p_{m0} \\ 0 \\ 0 \\ \text{sign}(p_{m3}) \cdot \frac{1}{\sqrt{1 - (p_{m0})^2}} \end{bmatrix}$$

Formel 5.12:  
Roll-Nick-  
Beseitigung

Anschließend erfolgt die Korrektur des Orientierungsquaternion durch Quaternionenmultiplikation, wobei für das Accelerometer pro Sample nur ein Bruchteil der Drehung ausgeführt wird und es für beide Teile einen Maximalwinkel gibt, der pro Sample korrigiert wird. Dies spiegelt ebenfalls den komplementären Charakter des Filters wieder, womit Accelerometer und Magnetometer langsam eingehen.

Zudem wird pro Sample abwechselnd mit Accelerometer und Magnetometer korrigiert, sofern beide Sensoren verwendet werden. Dies geschieht, um Kreuzeffekte abzuschwächen.

## 5.3 Höhenbestimmung

Unter Höhenbestimmung ist die Erfassung des vertikalen translatorischen Freiheitsgerades (z-Achse) zu verstehen. Da der AQuadropter für den Einsatz in Innenräumen (z.B. Gebäuden) konzipiert ist, wird unter Höhe bzw. Flughöhe der Abstand zum Boden verstanden. Das Wissen um die Höhe ist Grundvoraussetzung für den autonomen Flug, wobei permanent die Höhe zu regeln ist. Zur Bestimmung der Höhe kommen 5 Sensoren zum Einsatz:

- Langer IR-Sensor GP2Y0A710K0F
- Kurzer IR-Sensor GP2Y0A02YK
- US-Sensor SRF02
- Luftdrucksensor BMP085
- IMU

### 5.3.1 Datenfusion zur Höhenbestimmung

Grundlage der Datenfusion ist der Weighted Filter mit dem Design aus Abbildung 5.4 [P3]. Der Abbildung ist zu entnehmen, dass das Filter-Design zwei Schritte vorsieht und somit im Grunde aus 2 WF besteht. Da die IR Sensoren nur einen sehr kleinen gemeinsamen Messbereich haben, selektiert der Filter zunächst den besseren IR Sensor mit Hilfe von US Sensor, der IMU und Luftdrucksensor, die als Referenzsensoren dienen. Anschließend erfolgt die Gesamtfusion mit dem Ergebnis der IR Sensoren und dem US Sensor als Hauptsensoren und der IMU und dem Luftdrucksensor als Referenzsensoren.

Die IR Sensoren werden per ADC (engl.: Analog Digital Converter) mit einer Wiederholrate von 10ms ausgelesen. Die US Sensoren können nur mit einer Wiederholrate von 70ms ausgelesen werden. Deshalb erfolgt auch die Datenfusion mit einer Wiederholrate von 70ms. Die IR Messungen werden vorher durch einen Mittelwertfilter der Länge 7 geglättet. Die Implementierung, insbesondere die Treiber der Sensoren, greift auf die Arbeiten von Thilo Müller [A01] und Julian Rothe [A02] zurück.

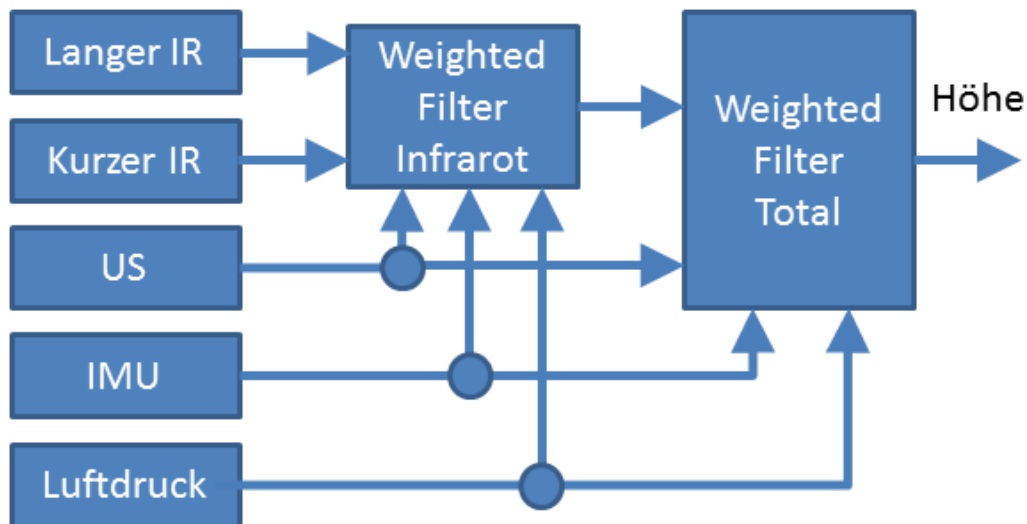


Abbildung 5.4: WF Design zur Höhenbestimmung

### 5.3.2 Weitere Lösungen zur Höhenbestimmung

Die Höhe des AQopter18 lässt sich auch durch weitere Ansätze erfassen. Für Flüge im Freien außerhalb des Messbereichs bietet sich eine alleinige Bestimmung auf Grundlage des Luftdrucksensors an. Der Luftdruck ist auch dazu gedacht, zwischen verschiedenen Stockwerken differenzieren zu können. Des Weiteren kann die Höhe mittels GPS und OTS bestimmt werden. Zudem sind mittels Fourier Tracking ebenfalls Höhenänderungen bestimmbar.

Da die implementierte Lösung zur Datenfusion sehr gute und robuste Ergebnisse liefert und gleichzeitig kostengünstig ist, sowie alle gemachten Anforderungen erfüllt, wurden weitere Ansätze nicht verfolgt.

## 5.4 Positionsbestimmung

Zur Positionsbestimmung existieren derzeit vier Lösungen:

- OF Sensor (ADNS-3080)
- Fourier Tracking
- GPS (Global Positioning System)
- OTS (Optisches Trackingsystem)

Diese vier Lösungen werden im Folgenden kurz beschrieben.

### 5.4.1 OF Sensor (ADNS-3080)

Der optische Flusssensor ADNS-3080 wird üblicherweise in optischen Mäusen verbaut und kann bei Verwendung geeigneter Optik Positionsänderungen eines Quadropters bestimmen. Rotationen stören dabei die Messungen und führen zu Fehlern. Höhenänderungen manipulieren ebenfalls die Sensormessungen, sofern sie nicht kompensiert werden. Mit Formel 5.13 wird unter Verwendung der aktuellen Höhe  $h$  aus den Sensormessungen  $\vec{m}_{OF}$  eine gültige Positionsänderung  $\vec{\Delta p}_{OF}$  bestimmt.



$$\overrightarrow{\Delta p_{OF}} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \overrightarrow{m_{OF}} \cdot \frac{h}{h_0} \cdot s$$

Formel 5.13: OF  
Höhenskalierung

Der Skalierungsfaktor  $s$  skaliert dabei die Sensormessungen bei der Höhe  $h_0$  in Meter. Der Treiber für den ADNS-3080 wurde von Michael Strohmeier [A04] implementiert.

#### 5.4.2 Fourier Tracking

Mit Fourier Tracking ist das im Stand der Technik (S. 37) beschriebene und von Eric Reinthal implementierte Verfahren gemeint, dass aus Kamerabildern Positionsänderungen in allen drei Raumachsen sowie Rotationen um die Gierachse ermitteln kann. Die Berechnungen finden auf dem LP-180 mit einer Wiederholrate von 100ms statt und als Sensor wird die C920 Kamera von Logitech verwendet. Das Fourier Tracking ist Voraussetzung für den gerichteten Flug [P08][A23] [63] [64] [60] [122].

#### 5.4.3 GPS

Für den Flug im Freien wurde im Rahmen der Bachelorarbeit von Andrea Tüchert [A28] ein Treiber für das 3DR uBlox GPS Modul implementiert, womit sich die Position des Quadropters in körperfesten, lokalen Koordinaten bestimmen lässt. Hintergrund ist die Überlegung, dass der AQopterI8 je nach Einsatzgebiet zeitweise auch im Freien operieren können soll. Hierzu bietet sich das bereits schon erprobte GPS- bzw. GNSS zur Positionsbestimmung an.

#### 5.4.4 OTS (Optisches Tracking)

Die Position des Quadropters kann mit Hilfe des OTS OptiTrack der Firma Natural Point [82] bestimmt werden. Dazu wird die Position vom Tracking Computer via W-LAN VRPN (engl.: Virtual-Reality Peripheral Network) an den onBoard PC übermittelt. Dieser sendet sie weiter an den AVR.

Das OTS hilft bei der Evaluierung und Erprobung des Systems und wurde vor allem für zukünftige Arbeiten zu den Themenfeldern „Kooperative Schwärme“ und „Manipulation“ in das System integriert.

## 5.5 Hinderniserkennung und Situationserfassung

Zur Hinderniserkennung gehören das Auslesen der OD Sensoren (S. 16, S. 56) und die Situationserfassung. Die Aufgabe der Situationserfassung ist die Datenfusion, dessen Ergebnis die von der Kollisionsvermeidung benötigten Abstände und Geschwindigkeiten sind. Der AQopterI8 verfügt zur Hinderniserkennung grundsätzlich über fünf Ansätze:

- Hinderniserkennung mittels Infrarot und Ultraschall [P10]
- PMD-basierte Hinderniserkennung
- Hinderniserkennung mittels Lidar (Laser Scanner)
- Radar-basierte Abstandsmessung
- Stereo-Optische Abstandsmessung

Mit dem WF (Weighted Filter, S. 65) können die verschiedenen Ansätze sehr einfach kombiniert werden. Die stereo-optische Abstandsmessung basiert auf der Implementierung von Helge

Lauterbach und wurde bereits im Kapitel Stand der Technik (Seite 40) kurz beschrieben. Die übrigen Ansätze zur Hinderniserkennung werden im Folgenden näher erläutert.

### 5.5.1 Hinderniserkennung mittels Infrarot und Ultraschall

Abbildung 5.5 veranschaulicht das Konzept der Hinderniserkennung basierend auf den 28 Abstandsmessungen  $\vec{d}$  der 16 IR und 12 US Sensoren. Herz des Systems sind 12 WF (S. 65) und 4 TAF (S. 70).

Als Stützinformation werden die gemessenen Positionsänderungen  $\Delta p$  vom OF und die Lagewinkel  $\alpha$  (roll) und  $\beta$  (pitch) der Orientierungsbestimmung hinzugezogen. Die optische Schätzung  $O_i$  (Formel 5.14) und die inertielle Schätzung  $E_i$  (Formel 5.15) ergeben die jeweiligen Positionsänderungen für die Weighted Filter mit dem Bereichs-Index  $i$ , der Erdbeschleunigung  $g$  und der Samplezeit  $T$ .

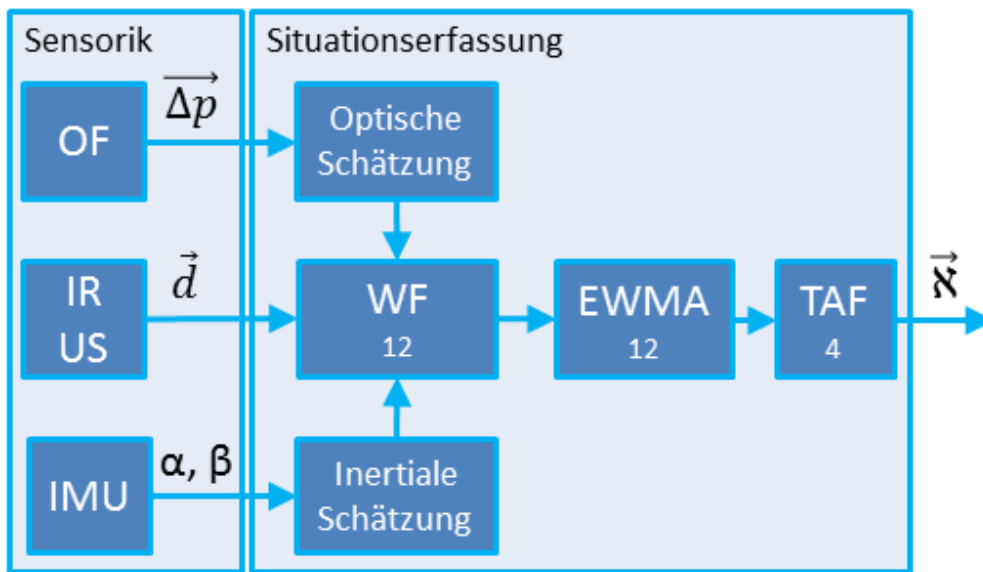


Abbildung 5.5: Konzept der Hinderniserkennung basierend auf IR und US Sensoren zur Abstandsmessung

$$O_i = \Delta p_x^{OF} \cos\left(i \frac{\pi}{2}\right) + \Delta p_y^{OF} \sin\left(i \frac{\pi}{2}\right) \quad \text{Formel 5.14}$$

$$E_i = g \cdot T \left( \sin(\alpha) \cdot \sin\left(i \frac{\pi}{2}\right) + \sin(\beta) \cdot \cos\left(i \frac{\pi}{2}\right) \right) \quad \text{Formel 5.15}$$

Die Fusion der unterschiedlichen Sensordaten erfolgt mittels WF (S. 65). Die 12 WF aus Abbildung 5.5 haben das in Abbildung 5.6 zu sehende identische Design. Jeder WF fusioniert jeweils 5 Sensoren: 2x IR (lang und kurz), US, OF und IMU. Genau genommen besteht in diesem Fall jeder WF aus zwei WF bzw. aus zwei Schritten. Der IR WF fusioniert den langen und den kurzen IR Sensor mit den übrigen Sensoren als Referenzsensoren. Das ist erforderlich, weil einer der beiden IR Sensoren häufig falsche Messwerte liefert und somit ein fehlerhafter Wert die Gewichtung verschlechtern würde. Im Anschluss an die IR Fusion wird das Ergebnis der IR Sensoren mit dem US Sensor fusioniert.



Sensoren ist notwendig, um eine möglichst zuverlässige Abdeckung der Umgebung zu gewährleisten. Somit können Fehler einzelner Sensoren kompensiert werden. Die US Sensoren haben einen Öffnungswinkel von 45°. Um eine doppelte Abdeckung zu erreichen, wären 16 Sensoren notwendig. Da die US Sensoren einander bei der Messung stören, müssen die Messungen zeitlich versetzt durchgeführt werden, wobei um 90° versetzte Sensoren gleichzeitig messen können. Die gleichzeitig messenden Sensoren werden zu einer Gruppe zusammengefasst. Bei 16 Sensoren wären dies 4 Gruppen, bei 12 Sensoren genügen drei Gruppen. Es besteht also ein Trade-Off Problem aus möglicher Wiederholrate der einzelnen Sensoren und Gesamtanzahl an eingesetzter Sensoren bzw. Abdeckung. In der Diplomarbeit von Thilo Müller wurde dies untersucht und die beschriebene Konfiguration aus 12 Sensoren erwies sich als das Optimum. Es werden für jeden Sektor je zwei IR Sensoren benutzt, um eine Messweite von 20cm bis 5m zu erreichen.

Die Ergebnisse der WF durchlaufen zunächst jeweils einen EWMA-Filter (S. 16), womit Rauschen unterdrückt wird. Anschließend werden die Messwerte durch 4 TAF (Triple Awareness Filter, S. 70) zum Vektor  $\vec{\mathfrak{N}}$  (gesprochen: Aleph) weiterverarbeitet und ergeben somit das Resultat der Situationserfassung (Datenfusion).

$$\vec{\mathfrak{N}} = \begin{pmatrix} d_{Front} \\ d_{Right} \\ d_{Rear} \\ d_{Left} \\ v_{Forth} \\ v_{Strafe} \end{pmatrix} = \begin{pmatrix} \vec{d}_f \\ \vec{v} \end{pmatrix} \quad \text{Formel 5.17}$$

Dieser Ansatz, insbesondere dessen Implementierung, greift auf die Arbeiten von Thilo Müller und Paul Benz zurück [A01][A10], welche teilweise gemeinsam entwickelt bzw. weiterentwickelt wurden.

### 5.5.2 PMD-basierte Hinderniserkennung

Als Sensor kommt die PMD nano der Firma pmdtechnologies zum Einsatz [123]. PMD steht für Photomischdetektor (S.25). Ein Bild der Kamera besteht aus 19.800 Pixeln. Zu jedem Pixel liefert der Treiber die Amplitude, den Abstand zur Linsenmitte und drei Werte, welche die 3D Koordinaten repräsentieren. Die Amplitude ist ein Maß für die Verlässlichkeit der Messung. Alle Daten sind Fließkommazahlen. Für eine 32-bit Floatdarstellung ergibt das pro Bild ca. 400kB. Dementsprechend speicher- und rechenintensiv ist die Auswertung. Aufgrund der hohen Rechenlast und weil aktuell nur ein x86-Treiber für die PMD Kamera verfügbar ist, findet die PMD-basierte Hinderniserkennung auf dem LP-180 Micro-ITX PC statt.

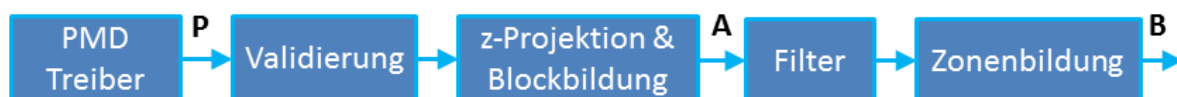


Abbildung 5.8: PMD-basierte Hinderniserkennung

Das Konzept der PMD-basierten Hinderniserkennung wird in Abbildung 5.8 verdeutlicht. Zunächst wird mit Hilfe des PMD Treibers das Tiefenbild **P** aufgenommen. Basierend auf der gemessenen Amplitude wird für alle Pixel eine Validierung durchgeführt, wobei ungültige Pixel verworfen werden. Anschließend erfolgt die z-Projektion und Blockbildung. Dabei werden die Bilder entsprechend der x-y-Koordinaten in Blöcke eingeteilt, wobei der z-Wert dem gemessenen Abstand und somit dem Inhalt der Blöcke entspricht. Dies ergibt die Blockmatrix **A**. Jeder Block besteht aus einer Liste mit

gemessenen Abständen. Die Dimension von A lautet 10 x 5. Anschließend erfolgt die Filterung, die aus einer Min-Funktion und einem EWMA-Filter pro Block besteht. Als Resultat enthalten die Blöcke nun die Abstände der jeweiligen nächsten Objekte (Abbildung 5.9).

In der Zonenbildung werden wiederum mehrere Blöcke zu einer logischen Zone zusammengefasst. Es werden fünf Zonen unterschieden: Oben, Links, Mitte, Rechts und Unten. Die Einteilung ist eine Vorleistung zur Kollisionsvermeidung und Navigation, welche abhängig vom Wissen der freien und belegten Zonen die Position des Quadropters manipuliert.

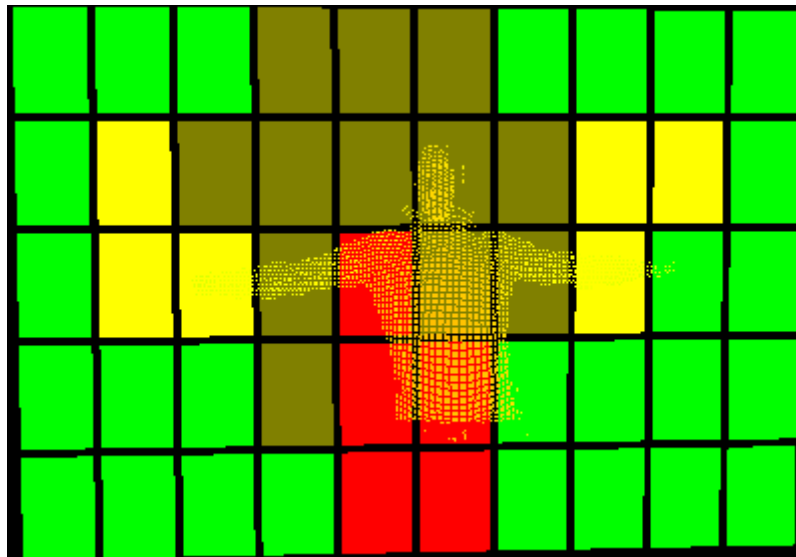


Abbildung 5.9: PMD Blockbildung

Abbildung 5.10 zeigt links die Blockbildung bestehend aus 20 Blöcken sowie rechts die Zonenbildung bestehend aus fünf Zonen. Die Farben repräsentieren die Entfernung, wobei rot einer Entfernung von 60cm bis 80cm und grün einer Entfernung zwischen 160cm und 250cm entspricht.

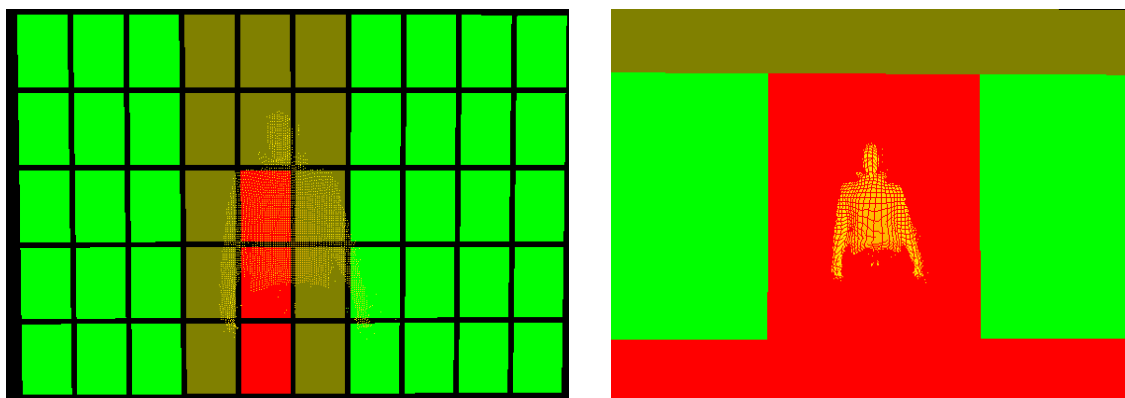


Abbildung 5.10: PMD-Bilder: Blockbildung (links) und Zonenbildung (rechts)

### 5.5.3 Hinderniserkennung mittels Lidar (Laser Scanner)

Als Lidar bzw. Laser Scanner kommt der rplidar der Firma robo peak zum Einsatz [112]. Dieser misst Entfernungen bis zu 6m mit einer Auflösung im Millimeterbereich und in einer 360° Ebene mit einer Winkelauflösung von ca. 1°.

In Abbildung 5.11 ist das Konzept der Hinderniserkennung mittels Lidar vereinfacht dargestellt. Wie schon bei der US+IR Hinderniserkennung werden die Messungen in Sektoren eingeteilt. Eine analoge

Aufteilung zur US+IR Hinderniserkennung hat den Vorteil der Kompatibilität, wodurch dieselben Ansätze zur Kollisionsvermeidung verwendet werden können.

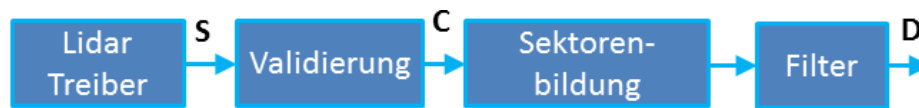


Abbildung 5.11: Hinderniserkennung mittels Lidar

Der Treiber liefert mit einer Wiederholrate von ca. 7 Hz die Abstandsmessungen  $S$  des Sensors sowie die dazugehörigen Winkel und einen Qualitätswert. Dabei variiert die Anzahl der Messungen pro Umlauf leicht, was die Verarbeitung etwas erschwert, da nicht von einer festen Anzahl an Messungen ausgegangen werden kann. Daher werden die Messungen in Listen verwaltet.

Zunächst werden in der Validierung Messungen, deren Qualitätswert unterhalb eines Schwellwerts liegen, verworfen. Ergebnis der Validierung sind ca. 360 Strahlen gültiger Abstandsmessungen  $C$ , wobei diese Messungen noch verrauscht sind (vgl. Abbildung 5.12 links). Aus Sicht der Kollisionsvermeidung sind dies zu viele, teilweise widersprüchliche Informationen. Daher werden die Daten bei der Sektorenbildung in 12 Listen entsprechend der 12 Sektoren, die jeweils einem  $30^\circ$  Bereich um den Quadrokopter herum entsprechen, eingeteilt. Teil der Sektorenbildung ist die Winkelkorrektur, d.h. die Projektion der Messungen auf die Mittellinie des Sektors, damit die Messungen eines Sektors vergleichbar (winkelunabhängig) sind.

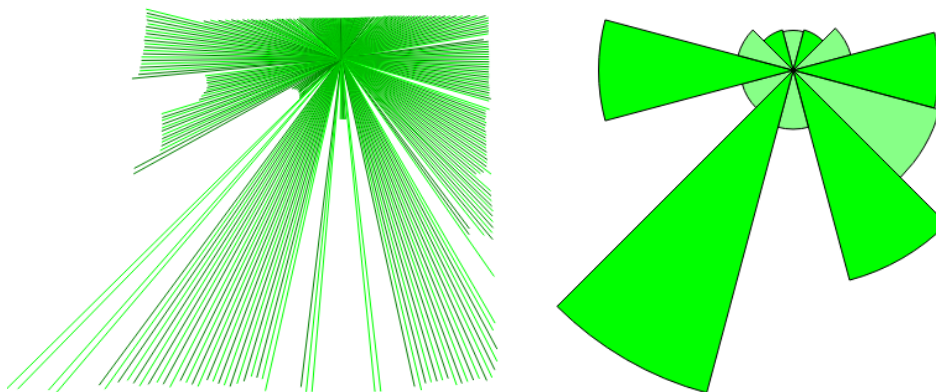


Abbildung 5.12: Lidar Messung: Abstandswerte nach Validierung (links) und Sektorenbildung + Filterung (rechts)

Jeder Sektor durchläuft anschließend ein Filter, um Rauschen und Fehler zu unterdrücken, sowie den minimalen gültigen Abstand zu bestimmen. Dazu kommt ein Histogramm Filter zum Einsatz, welches die Liste der Messungen eines Sektors in ein Histogramm mit der Bereichsbreite 20cm einteilt. Der niedrigste Histogramm-Bereich, der in der aktuellen und letzten Messung mindestens 3 Einträge aufweist, wird weiter verwendet. Alle Messungen eines Sektors werden gemittelt, wobei bei der Mittelwertbildung die aktuellen Messungen zu 70% und die vorherigen Messungen zu 30% einfließen. Diese Prozedur wird für alle 12 Sektoren wiederholt und ergibt die gefilterten Sektoren  $D$  (vgl. Abbildung 5.12 rechts).

#### 5.5.4 Radar-basierte Abstandsmessung

Die Radar-basierte Abstandsmessung konnte bisher nur experimentell erprobt werden. Die verfügbaren Lösungen sprengen die gemachten Anforderungen in Hinblick auf Kosten, Größe und Gewicht. Mit kleinen, kostengünstigen Sensoren können zwar Abstände gemessen werden. Die Genauigkeit dieser Sensoren schwankt aber stark in Abhängigkeit von Faktoren wie Bewegung,

Material, Größe und Entfernung des Objekts [A12][A26][A30].





*The way to get started is to quit talking  
and begin doing.*  
[Walt Disney]

## 6 Regel- und Steuerung (Aktuatorik)



## 6.1 Überblick Regel- und Steuerung

Aufgabe der Regel- und Steuerung ist das Umsetzen des gewünschten Systemverhaltens durch Ansprechen der Aktuatoren und unter Verwendung der Resultate der Signalverarbeitung, welche der Bestimmung des Systemzustandes dienen.

Die Regel- und Steuerung setzt sich zusammen aus der Regelung, d.h. den Reglern selbst, aus der Kollisionsvermeidung, der eigentlichen oder primären Steuerung (Flight Management) und weiteren Funktionen. Die Steuerung kann vom einfachsten Fall der Soll-Wert-Vorgabe der Regler bis hin zu komplexen Funktionen, die eine gesamte Mission oder ein bestimmtes komplexes Systemverhalten steuern, gehen. Ein Beispiel für eine komplexe Mission ist die autonome Objektsuche.

Zu den weiteren Funktionen der Steuerung gehören das Kippgelenk, der Gerichtete Flug und die Looping-Steuerung. Hilfsmittel zur Anzeige von Systemzuständen und Interfaces zur Befehlseingabe werden in diesem Kapitel ebenfalls beschrieben. Dazu gehören die Steuerungen via GPS / GNSS und Kartenmaterial, optisches Tracking, PC, Smartphone und Miniaturgerät sowie die HMD-Steuerhilfe.

## 6.2 Überblick Regelung

Der AQopter18 verwendet 8 Regler mit insgesamt ca. 16 verschiedenen Parametersätzen:

- 6 DOF Regelung bestehend aus Lage RPY, Position XY und Höhe Z
- Kollisionsvermeidung mit Abstandsregelung:  $d_x$  in x- und  $d_y$  in y-Richtung

Dabei sind jeweils 3 Regler-Paare (RP, XY,  $d_x$ ,  $d_y$ ) vollkommen analog und unterscheiden sich lediglich in den Eingangsgrößen sowie den inneren Zuständen. Die Parametersätze berücksichtigen die unterschiedlichen Konfigurationen des Systems, wobei die wesentlichen Faktoren hierbei das Gewicht sowie die verwendete Sensorik zur Regelung sind.

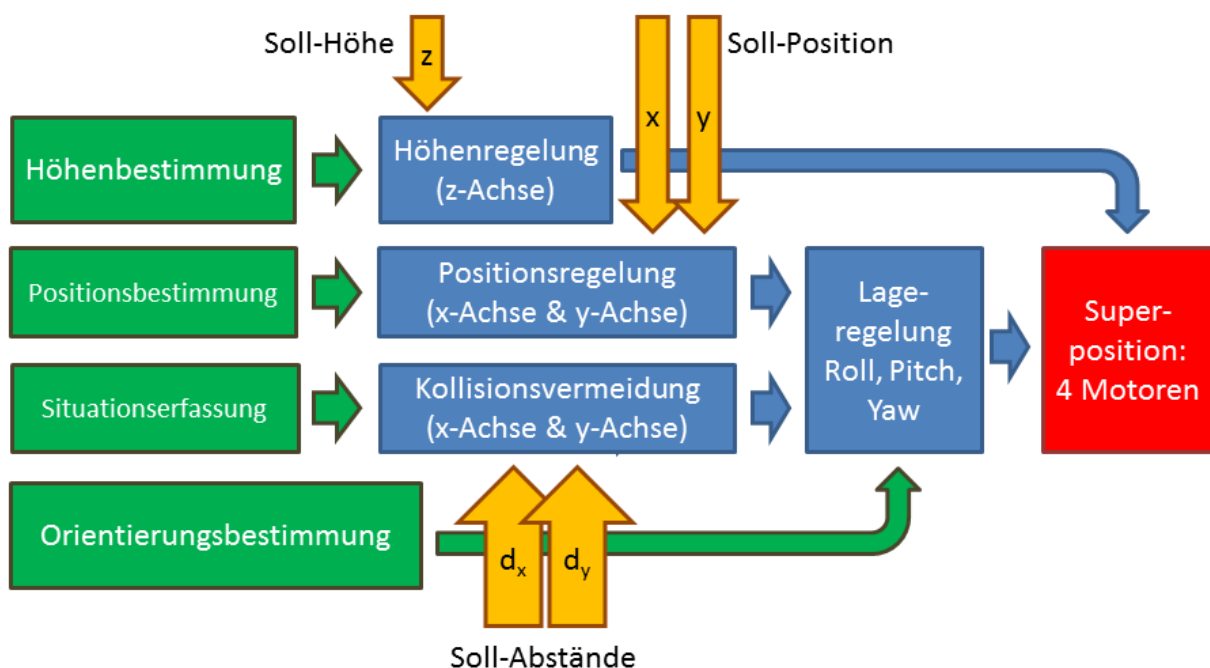


Abbildung 6.1: Überblick zur Regelung

Abbildung 6.1 veranschaulicht das Zusammenspiel der Regler. Die Höhenregelung gibt den Gesamtschub der Motoren vor und ermittelt dies auf Grundlage der ihr vorgegebenen Soll-Höhe und der durch die Höhenbestimmung gemessenen Ist-Höhe. Der so ermittelte Gesamtschub wird abhängig von den Resultaten der Lageregelung auf die 4 Motoren aufgeteilt. Die Lageregelung wiederum wird durch die Positionsregelung und - falls aktiv - auch durch die Kollisionsvermeidung manipuliert. Dabei hat die Kollisionsvermeidung Vorrang gegenüber der Positionsregelung und kann diese überschreiben. Die Ist-Werte der Positionsregelung, Kollisionsvermeidung und Lageregelung entstammen der Positionsbestimmung, Situationserfassung und der Orientierungsbestimmung. Vorteil dieses modularen Aufbaus ist, dass die einzelnen Regler weitestgehend unabhängig voneinander entwickelt, optimiert und verwendet werden können, wobei lediglich die Lageregelung Grundvoraussetzung ist. So kann das System z.B. optional mit Höhenregelung, Positionsregelung und Kollisionsvermeidung betrieben werden und Erweiterungen und Modifikationen sind mit geringem Aufwand möglich.

Zur Realisierung des autonomen Fluges wurden zwei Lösungen entwickelt, die in den beiden nachfolgenden Abschnitten näher erläutert werden. Die 6 DOF Regelung, bestehend aus Lageregelung, Höhenregelung und Positionsregelung, ermöglicht es, Positionen autonom anzufliegen. Als Alternative zur Positionsregelung oder ergänzend ermöglicht die Kollisionsvermeidung einen autonomen Flug unter Manipulation von Abständen in x- und y-Richtung.

Das Hintereinanderschalten mehrere Regler nennt sich Kaskadenregelung und ist ein weit verbreitetes Verfahren in der Regelungstechnik zur Regelung von Mehrgrößensystemen [46] [124] (vgl. Kapitel 2.5).

## 6.3 Regelung (6 DOF)

Die Regelung der 6 Freiheitsgrade RPYXYZ, auch 6 DOF Regelung genannt, ist in Abbildung 6.2 näher beschrieben. Diese setzt sich zusammen aus dem Höhenregler für die Höhenregelung, dem Lateral- und Longitudinal-Regler für die Positionsregelung und der Lageregelung bestehend aus Roll-Regler, Pitch-Regler und Yaw-Regler. Dabei gehen der Longitudinal-Regler (X) in den Pitch-Regler und der Lateral-Regler (Y) in den Roll-Regler ein, da beim Quadropter Positionsänderungen durch Lageänderungen erreicht werden.

### 6.3.1 Lageregelung (Roll, Pitch, Yaw)

Der Roll- und der Pitch-Regler sind identisch und unterscheiden sich vom Yaw-Regler lediglich in den Regler-Parametern. Es kommt ein PID-Regler mit ein paar Modifikationen zum Einsatz. Zu den Modifikationen gehören eine Schranke für das Integral, die Verwendung der Winkelgeschwindigkeit für den D-Anteil sowie ein Cut-Off für den Zuwachs des Integrals. Mit Hilfe des Faktors  $i_c$  wird der Zuwachs des Integrals abhängig vom Winkelfehler in drei Stufen unterteilt:

- Bei einem Winkelfehler unter  $1^\circ$  ist  $i_c = 1$ , um der stationären Regelabweichung entgegenzuwirken. Hier genügt der P-Anteil nicht mehr, um kleine Fehler zu korrigieren.
- Bei einem Winkelfehler zwischen  $1^\circ$  und  $5^\circ$  ist  $i_c = 0.01$ , damit der I-Anteil nicht zum Schwingen anregt. Fehler werden hier durch den P- und D-Anteil vermieden.
- Bei einem Winkelfehler über  $5^\circ$  ist  $i_c = 0.0001$ , um zusätzlich Überschwingen zu unterdrücken. In diesem Fall ist der P-Anteil groß genug, um Fehler zu korrigieren.

Eine Besonderheit stellen zwei weitere Konfigurationsmöglichkeiten dar. Beim Flug mit Manipulator ist eine schnellere Integration wünschenswert, die im Normalfall eher Fehlerpotential aufweist. Deshalb kann dies modular zugeschaltet werden und ist besonders bei Verwendung des Manipulators notwendig, da nur der Integralteil die ungleiche Last des Roboterarms kompensieren kann. Für diesen Fall wird der I-Anteil stärker aufintegriert, sofern sich der Fehler über einen Zeitraum von 1s nicht zumindest um 1° geändert hat und gleichzeitig zumindest 1° beträgt. Ebenso gibt es für den Loopingflug ein paar Anpassungen, wie eine Beschränkung des Gradbereichs auf +/-180° sowie eine Reduktion des D-Anteils während des Loopings.

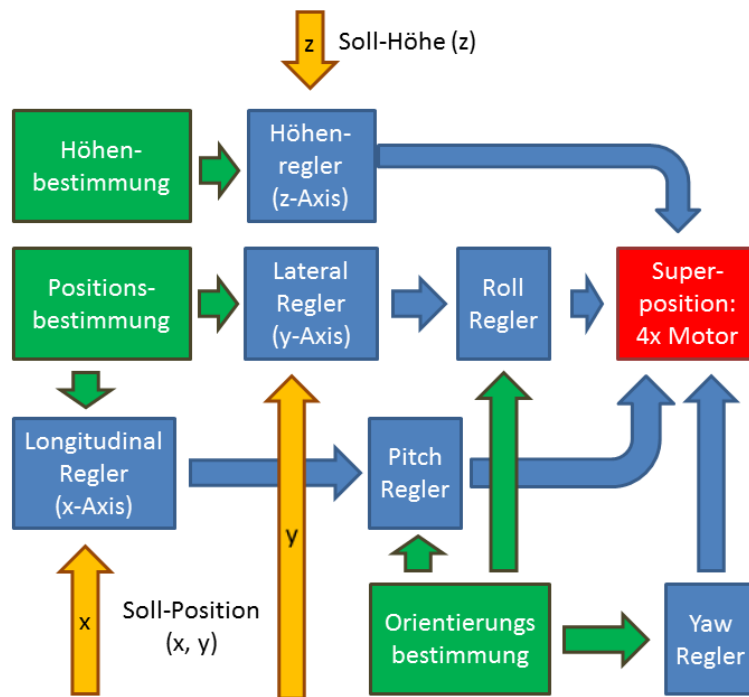


Abbildung 6.2: 6 DOF Regelung

### 6.3.2 Höhenregelung (Z)

Zur Höhenregelung kommt ein modifizierter PID-Regler zum Einsatz, welcher den Gesamtschub, die Summe der Steuerwerte aller vier Motoren, bestimmt. Aus Stabilitätsgründen wird der I-Anteil nur aufintegriert, sofern sich der Fehler nicht ändert. Zudem wird der Zuwachs des Integrals beschränkt. Der P- und der D-Anteil werden ebenfalls limitiert, um die Stabilität zu erhöhen. Das Ergebnis des Reglers, der Gesamtschub, wird entsprechend der Spannung skaliert, um den Einfluss der Batteriespannung zu berücksichtigen. Zudem wird der Schub nach oben durch die Neigung des Quadropters reduziert. Deshalb wird der Schub um den Skalierungsfaktor  $s_h$  aus Formel 6.1 verstärkt.

$$s_h = \frac{1}{\cos\left(\frac{R+P}{4}\right)} \quad \text{Formel 6.1}$$

Der Skalierungsfaktor  $s_h$  wurde empirisch eingestellt. Eine mathematisch bestimmte Lösung aus der

Geometrie entsprechend dem Schubvektor erwies sich als zu stark. Der Skalierungsfaktor  $s_n$  wird aus Sicherheitsgründen auf 1.4, d.h. ca.  $\sqrt{2}$  beschränkt, da davon ausgegangen werden kann, dass keine allzu großen Neigungen jenseits der 45° auftreten. Am Ende wird der so ermittelte Gesamtschub noch auf den gültigen und gewünschten Bereich saturiert, damit die Höhenregelung nicht die Lageregelung behindert [P3, A02].

### 6.3.3 Positionsregelung (X,Y)

Die Positionsregelung verwendet einen modifizierten PID-Regler, der einmal implementiert und mit zwei Datensätzen (zwei Structs) für die beiden Achsen arbeitet. Der Positionsregler ist so konfiguriert, dass er die Position mittels GPS, OTS, OF und Fourier Tracking (FT) regeln kann.

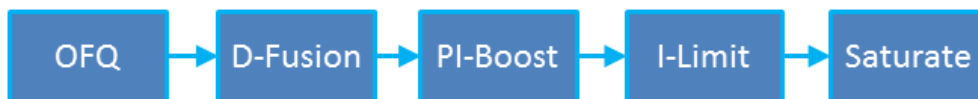


Abbildung 6.3: Positionsregler

Abbildung 6.3 illustriert das Konzept des modifizierten PID-Reglers zur Positionsregelung. Abhängig von der Qualität der Messungen des OF Sensors geht das Fourier Tracking dynamisch in die Regelung ein (OFQ). Da das Fourier Tracking mit einer langsameren Wiederholrate (100ms statt 10ms) misst und auch ein anderes Rauschverhalten aufweist, müssen die PID-Parameter angepasst werden, um eine stabile Regelung zu gewährleisten. Eine Anpassung der PID-Parameter wird für jeden Sensor separat vorgenommen, im Fall OF und Fourier-Tracking wird aber dynamisch pro Sample entschieden, welche Parameter verwendet werden, da auch pro Sample OF und Fourier Tracking entsprechend der Lichtverhältnisse unterschiedlich eingehen bzw. dynamisch zwischen diesen Sensoren im Betrieb gewechselt wird. Entsprechend werden dann die Reglerparameter angepasst.

Im zweiten Schritt werden in der D-Fusion die D-Anteile, d.h. die Geschwindigkeitskomponenten, die aus OF und Fourier Tracking berechnet werden, entsprechend der Qualität von OF und Fourier Tracking fusioniert.

Aus Stabilitätsgründen wurde der I-Anteil des Reglers niedrig ausgelegt und der Regler integriert in Abhängigkeit von Fehler und D-Anteil. Dies kann dazu führen, dass das System zuweilen sehr langsam integriert, so dass es in ungünstigen Fällen nicht mehr gegen andere Fehlerquellen ankommt und sein Ziel nicht erreicht. Es bleibt dann kurz davor hängen, weil in einem solchen Fall der P-Anteil nicht mehr ausreichend groß ist, um diesen letzten Fehler zu überwinden und der I-Anteil zu schwach bzw. zu langsam ist. Damit das System dennoch in angemessener Zeit sein Ziel erreicht, wurde der PI-Boost eingeführt. Nähert sich der Quadrocopter nach einem Timeout von 200ms nicht zumindest 20cm seinem Ziel, werden sowohl der P-Anteil als auch der Integrationszuwachs um einen eigenen Faktor gesteigert. Dies bewirkt, dass das System auch bei Fehlern wie z.B. einem Drift der Orientierung, zuverlässig sein Ziel erreicht, ohne insbesondere im stationären Fall die Stabilität zu verschlechtern.

Schließlich werden das Integral des I-Anteils limitiert, sowie die P-, D- und PD-Anteile saturiert.

## 6.4 Superposition

Das Konzept der Superposition bietet eine Lösung für das komplexe Regelungsproblem des Quadrocopters (vgl. Kapitel 2.5). Das MIMO 4I6O System wird aufgeteilt in 6 bzw. 8 Regler. Die

einzelnen Regler werden überlagert, überschrieben bzw. bilden eine Kaskade (vgl. Abbildung 6.1 und Abbildung 6.2). In die Motorstellwerte gehen damit letztlich nur noch die Höhenregelung und die Lageregelung direkt ein. Diese vier Regler sind zu superpositionieren. Prinzipiell erfolgt dies durch Anwendung von Formel 6.2.

$$\vec{M} = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{pmatrix} = \begin{pmatrix} s - r + y \\ s + p - y \\ s + r + y \\ s - p - y \end{pmatrix} \quad \text{Formel 6.2}$$

Hierbei steht  $\vec{M}$  für die Motorstellwerte,  $s$  für das Schubgas als Ergebnis der Höhenregelung und  $r$ ,  $p$  und  $y$  für die Ergebnisse von Roll-, Pitch- und Yaw-Regler (RPY-Regelung).

Bei der Anwendung von Formel 6.2 ist zu beachten, dass die Regler miteinander um die Leistung der Motoren konkurrieren. Dieser Zusammenhang kann bereits aus Formel 6.2 abgelesen werden. Bei vollem Schubgas  $s$  kann die Lageregelung z.B. gar nicht mehr eingreifen. Dies liegt daran, dass alle Motoren bereits auf Anschlag sind. Die Lageregler regeln aber den Quadrocopter durch Differenz und zwar über zwei Differenzen. Für den Yaw-Regler muss die Differenz einer Achse im Verhältnis zur anderen Achse möglichst viel Spielraum liefern, während für Roll und Pitch (beide analog) die Differenz innerhalb einer Achse von Bedeutung ist. Damit sich die Regler nicht gegenseitig in die Quere kommen, ist eine Lösung zur Aufteilung der Motorkapazitäten erforderlich.

Eine sehr einfache Lösung ist die fixe Aufteilung. Diese Lösung hat den Nachteil, dass häufig Potential verschenkt wird. Eine bessere Alternative bietet die dynamische Aufteilung mit einer fixen Minimalkapazität. Darunter ist zu verstehen, dass ein bestimmter Anteil garantiert und der Rest dynamisch zugewiesen wird, sofern der konkurrierende Regler diesen nicht anfordert.

Für den AQopter18 wurde eine dynamische Aufteilung mit einer fixen 90-10 Minimalkapazität realisiert. Dies bedeutet, die RP-Regelung, bestehend aus Roll-Regler und Pitch-Regler, erhält in jedem Fall 90% des verfügbaren Schubes. Hintergrund ist die Überlegung, dass für einen stabilen Flug die RP-Regelung wesentlich wichtiger ist als die Yaw-Regelung. Zudem hat die Yaw-Regelung ohnehin meist wenig zu tun.

Die Yaw-Regelung erhält zumindest 10% der Kapazität und alles, was die Lageregelung nicht anfordert. Obwohl die Yaw-Regelung somit sehr wenig Schub zugesichert bekommt, kann der AQopter18 sehr schnell gieren, d.h. sich um 90° in ca. 1-2s drehen. Dies liegt daran, dass im horizontalen Flug die RP-Regelung ihre Kapazität bei weitem nicht ausschöpft und die Yaw-Regelung über die dynamische Zuweisung darauf zugreifen kann.

## 6.5 Kollisionsvermeidung

Dieser Abschnitt beschreibt die Kollisionsvermeidung, wie sie unter Verwendung der IR+US Sensorik und der dazu in Kapitel 5.4 beschriebenen Datenfusion, auch Hinderniserkennung genannt, implementiert ist [P2] [P10] [A01] [A10]. Die Kollisionsvermeidung mittels Lidar ist dazu analog implementiert.

Ziel der Kollisionsvermeidung ist es, durch Abstandsregelung Kollisionen zu vermeiden, um nicht stärker als nötig in die Steuerung einzugreifen. Dadurch wird die Menge möglicher Manöver eben

soweit eingeschränkt, wie unbedingt nötig. Dieses Vorgehen ermöglicht einen kontrollierten, autonomen Flug. Damit unterscheidet sich dieser Ansatz fundamental von einfachen Lösungen, bei denen das System in die Gegenrichtung ausweicht und sich so stets von Objekten entfernt.

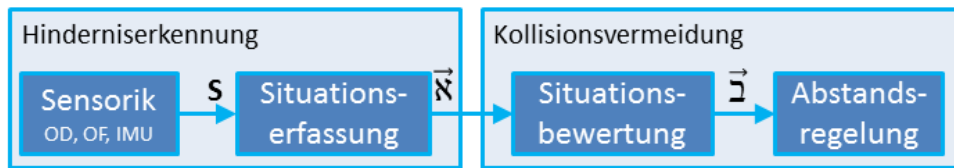


Abbildung 6.4: Konzept Hinderniserkennung und Kollisionsvermeidung

Das Konzept der Hinderniserkennung und Kollisionsvermeidung wird in Abbildung 6.4 zusammengefasst. Die Hinderniserkennung bestehend aus der Sensorik und der Situationserfassung wurde bereits in Abschnitt 5.5.1 beschrieben.  $S$  ist die Messmatrix der Sensoren (Formel 5.16). Das Ergebnis der Hinderniserkennung ist der Vektor  $\vec{x}$  (Formel 5.17). Dieser besteht aus den Abständen in alle vier „Himmelsrichtungen“ sowie der Geschwindigkeit. Das Problem der Kollisionsvermeidung, unter Beschränkung auf die x-y-Ebene, ist vollständig durch den Vektor  $\vec{x}$  beschrieben, da nur die minimalen Abstände sowie die Geschwindigkeiten der orthogonalen Achsen von Interesse sind. Alle sonstigen Abstands- und Geschwindigkeitsinformationen lassen sich auf diese zurückführen, bzw. durch diese ausdrücken.

Der Vektor  $\vec{x}$  wird von der Kollisionsvermeidung verwendet, um zu erkennen, ob und inwiefern in die Lageregelung eingegriffen werden muss, damit Kollisionen vermieden werden. Die Kollisionsvermeidung gliedert sich in die Situationsbewertung und die Abstandsregelung. Aufgabe der Situationsbewertung ist es zu beurteilen, ob und wie prinzipiell reagiert werden sollte, während die Abstandsregelung einen Regler beinhaltet, der dies zu konkreten Vorgaben für die Lageregelung umsetzt.

Hierbei sind sowohl Situationsbewertung als auch Abstandsregelung für jede Achse analog ausgelegt, d.h. sowohl für Roll als auch für Pitch gibt es je eine identische Situationsbewertung und Abstandsregelung.

Pro Achse verbleiben somit zwei unabhängige Abstände und eine Geschwindigkeit, die ausreichen, um die Situation zu bewerten. Das sind der Abstand in Achsrichtung, auch positiver Abstand  $d_+$  genannt, der Abstand entgegen der Achsrichtung, auch negativer Abstand  $d_-$  genannt, sowie die Geschwindigkeit  $v$  in Achsrichtung.

### 6.5.1 Situationsbewertung

Die Situationsbewertung teilt die beiden Abstände  $d_+$  und  $d_-$  zunächst separat in drei Zonen (vgl. Abbildung 6.5) ein [P2][A01]:

- Im Fernbereich ist der Abstand größer als die Grenzdistanz  $d_g$ .
- Im Nahbereich befindet sich der Abstand zwischen Minimaldistanz  $d_m$  und Grenzdistanz  $d_g$ .
- Im Gefahrenbereich ist der Abstand kleiner als die Minimaldistanz  $d_m$ .

Entsprechend dieser Einteilung erfolgt der Eingriff in die Regelung:

- Im Fernbereich ist das Hindernis weit genug entfernt, so dass die Kollisionsvermeidung nicht

in die Steuerung eingreift.

- Im Nahbereich wird die Geschwindigkeit gedämpft, indem diese als Differenzialteil in das System eingeht. Dadurch wird verhindert, dass der Quadrokopter zu schnell auf das Hindernis zusteuert.
- Im Gefahrenbereich wird das System aktiv auf die Minimaldistanz  $d_m$  geregelt.

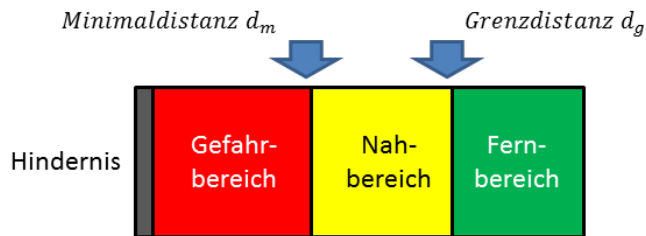


Abbildung 6.5: Zoneneinteilung der Situationsbewertung

Im nächsten Schritt müssen die beiden Abstände miteinander in Beziehung gebracht werden. Dazu dienen Formel 6.3, Formel 6.4 und Formel 6.5 sowie die Zustandsmaschine aus Abbildung 6.6. Dabei existieren grundsätzlich vier Möglichkeiten:

- Sind beide Abstände größer als die Grenzdistanz, so befindet sich kein Hindernis in der Nähe und die Kollisionsvermeidung muss nicht eingreifen (A).
- Andernfalls, und sofern Formel 6.4 wahr ist, befindet sich auf beiden Seiten ein Hindernis (D). In dem Fall muss der Quadrokopter seine Position in der Mitte der Hindernisse regeln.
- Ist beides nicht zutreffend und Formel 6.5 falsch, dann befindet sich lediglich ein Hindernis in Axialrichtung, worauf der Quadrokopter reagieren muss.
- Im vierten Fall befindet sich das Hindernis lediglich entgegen der Axialrichtung.

$$d_g < \min(d_+, d_-) \quad \text{Formel 6.3}$$

$$(d_+ < d_g) \& (d_- < d_g) \& (d_+ + d_- < 2 \cdot d_m) \quad \text{Formel 6.4}$$

$$d_+ < d_- \quad \text{Formel 6.5}$$

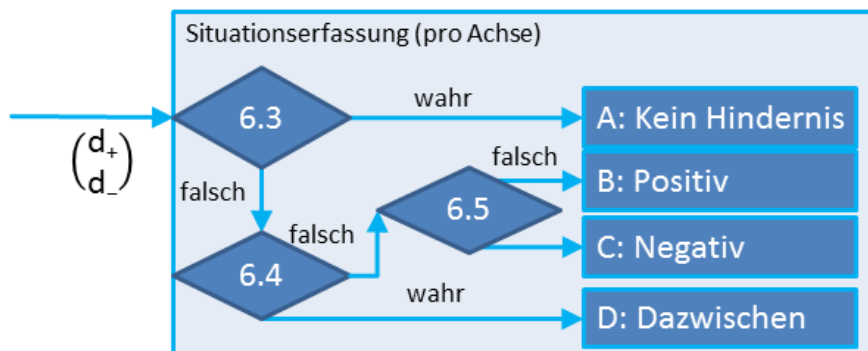


Abbildung 6.6: Zustandsmaschine zur Situationsbewertung

In Abhängigkeit vom Zustand übergibt die Situationsbewertung an die Abstandsregelung die Matrix  $\mathcal{D}$  (gesprochen: Beth), bestehend aus den beiden Vektoren  $\vec{\mathcal{D}}_x$  und  $\vec{\mathcal{D}}_y$  für die beiden Achsen (Formel 6.6) und dem Fehler  $e$ , dem Abstand  $d$ , der Geschwindigkeit  $v$ , der Geschwindigkeit in Richtung



Hindernis  $w$  und weiteren zustandsabhängigen Reglerparametern.

$$\mathfrak{z} = \begin{pmatrix} \vec{x}_x & \vec{x}_y \end{pmatrix} = \begin{pmatrix} e_x & e_y \\ d_x & d_y \\ v_x & v_y \\ w_x & w_y \\ \vec{c}_x & \vec{c}_y \end{pmatrix} \quad \text{Formel 6.6}$$

Die Reglerparameter werden im Zustand D sanfter eingestellt als im Fall B und C, da das System sonst zu heftig reagiert bzw. sich aufschaukelt.

### 6.5.2 Abstandsregelung

Aufgabe der Abstandsregelung ist das Vermeiden von Kollisionen durch das Halten des Abstandes zum Hindernis, ohne sich dabei weiter vom Hindernis zu entfernen, als gewünscht. Dazu bewirkt die Regelung Bewegungen sowohl vom Hindernis weg als auch auf dieses zu. Dabei darf die Regelung nicht zu stark oder zu schwach reagieren, da sich der Quadrokopter sonst vom Hindernis wegbewegt oder gegen dieses stößt.

Zur Abstandsregelung wird ein modifizierter PID-Regler verwendet. Abbildung 6.7 veranschaulicht die Modifikationen des Reglers. Die Steigung des P-Anteils ist vom Betrag des Fehlers abhängig und in drei Bereiche unterteilt. Bei kleinen Fehlern steigt der P-Anteil zunächst moderat an, damit das System darauf angemessen reagiert. Im zweiten Bereich wird die Steigung reduziert, damit sich das System nicht aufschaukelt bzw. zu heftig reagiert. Im dritten Bereich schließlich nimmt die Steigung deutlich zu, da hier eine Kollision droht und das System entsprechend stärker reagieren sollte, um diese zu vermeiden.

Der P-Anteil wird aus Stabilitätsgründen begrenzt. Darüber hinaus wird der P-Anteil für den Fall, dass sich der Abstand zum Hindernis erhöht, reduziert, weil sich nun der Quadrokopter vom Hindernis weg entfernt und eine Kollision bereits vermieden ist (Richtungsmodus). Dies verhindert, dass sich das System zu stark wegbewegt. Für den Fall, dass der Abstand 68cm (B,C) bzw. 50cm (D) unterschreitet, wird der P-Anteil deutlich gesteigert, da eine Kollision droht (Boost).

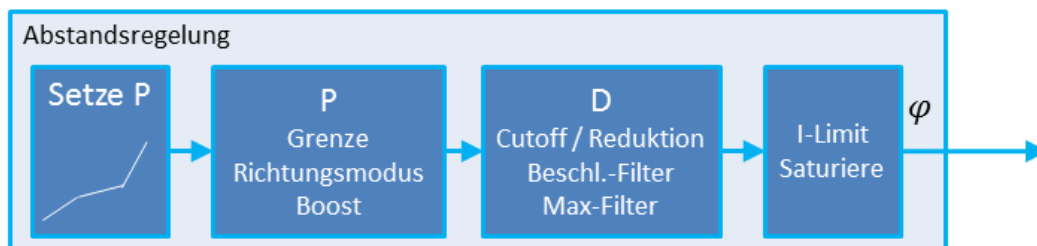


Abbildung 6.7: Abstandsregler

Für den D-Anteil wird die aus den Abstandsmessungen durch Datenfusion gewonnene Geschwindigkeit verwendet. Da die Abstandsmessungen mit zunehmender Entfernung fehleranfälliger bzw. verrauschter werden, wird der D-Anteil in Abhängigkeit von der Entfernung reduziert. Zudem werden mittels Beschleunigungsfilter ungültige Beschleunigungen durch

Begrenzung ausgefiltert. Der D-Anteil wird zu null gesetzt, sofern eine Plausibilitätsgrenze überschritten ist (Max-Filter). Damit werden offenkundig ungültige Geschwindigkeiten verworfen.

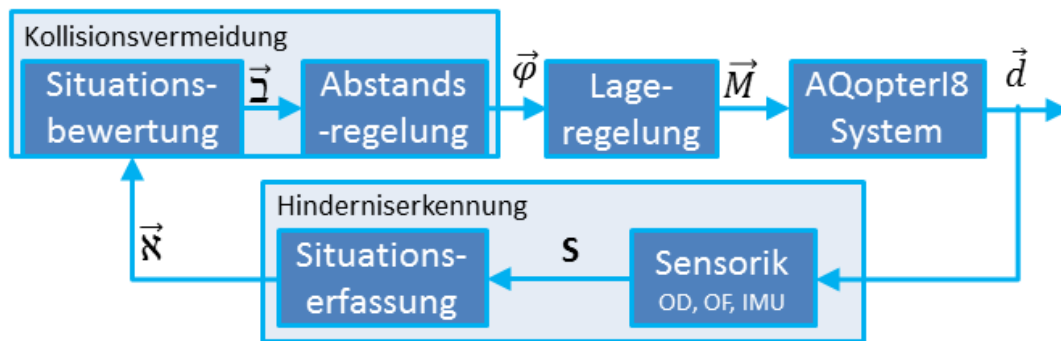


Abbildung 6.8: Regelkreis Kollisionsvermeidung

Schließlich wird das Integral begrenzt und es werden alle Anteile sowie das Ergebnis des Reglers saturiert. Das Resultat des Reglers ist der durch die Lageregelung einzustellende Winkel  $\varphi$ . Der Regelkreis in Abbildung 6.8 fasst den Ablauf zur Hinderniserkennung und Kollisionsvermeidung nochmals zusammen.

## 6.6 Flight Management (Steuerung)

Zur höchsten Steuerungsschicht, dem Flight Management, gehören die Steuerkontrolle, die den Betriebsmodus umsetzt, die Flugkontrolle, die die Navigation durchführt, sowie die Höhensteuerung, die das Starten und Landen steuert.

### 6.6.1 Steuerkontrolle (SteerCtrl)

Aufgabe der Steuerkontrolle ist das Umschalten zwischen den verschiedenen Betriebsmodi (vgl. Abbildung 6.9). Abhängig vom Betriebsmodus werden die verschiedenen Regler aktiviert. Dabei ist die Lageregelung in allen Betriebsmodi mit Ausnahme des initialen Modus (Off) aktiv und regelt den jeweils vorgegebenen Winkel. Die folgende Übersicht listet die Betriebsmodi auf:

- Off: Dies ist der initiale Betriebsmodus, bei dem die Motoren ausgeschaltet sind.
- RP: In diesem Betriebsmodus und allen weiteren sind die Motoren angeschaltet. Beim manuellen Flug (engl. remotely piloted, kurz RP) steuert der Pilot den kompletten Quadrokopter selbst, indem er per Fernsteuerung Gas und Lage vorgibt. Über die Manipulation von Gas und Lage werden Höhe und Position gesteuert.
- RPH: Beim manuellen Flug mit Höhenkontrolle wird die Höhe geregelt, sodass der Quadrokopter eine vorgegebene Höhe hält. Die Soll-Höhe kann per Fernsteuerung oder Kommando verändert werden. Weiterhin steuert der Pilot die Lage und darüber die Position.
- R6: In diesem Modus, der Regelung der 6 Freiheitsgerade, fliegt das System autonom und alle 6 Freiheitsgerade des Quadrokopters werden geregelt. Der Quadrokopter hält somit die aktuell vorgegebene Position, die per Kommando verändert werden kann.
- AI: In diesem Modus, dem autonomen Idlezustand, ist das System bereit („scharf“) einen autonomen Flug durchzuführen. Es wartet lediglich auf das Kommando. Der Modus ist erforderlich, damit per PC-Kommando der gesamte autonome Flug einschließlich Start und Landung durchgeführt werden kann, ohne die Fernsteuerung berühren zu müssen. Andernfalls wäre das System im Zustand R6 und somit bereits in der Luft.

- AF1: Beim autonomen Flug 1 arbeitet der Quadrocopter eine vorher einprogrammierte Wegpunktliste ab, wobei die Wegpunkte nacheinander angefliegen werden. Es wird der sogenannte Wegpunktflug ausgeführt. Dem Modus AF1 geht der Modus AI voraus. Nachdem der letzte Wegpunkt erreicht ist, landet der Quadrocopter an diesem.
- AF2: Beim autonomen Flug 2 kann die Wegpunktliste im Betrieb ergänzt oder verändert werden. Die gesamte Wegpunktliste wird dann nicht mehr vollständig vom Mikrocontroller verwaltet, sondern vom LP-180 on-Board PC. Während beim AF1 der Mikrocontroller entscheidet, ob ein Wegpunkt erreicht wurde, entscheidet es beim AF2 der PC. Dadurch kann dieser in die Navigation jederzeit eingreifen. Dies wird Wegpunktkommandierung (WPC) genannt, da die Wegpunkte vom on-Board PC oder der Bodenstation per Kommando an den Mikrocontroller verschickt werden. Die Wegpunktkommandierung ist erforderlich für den gerichteten Flug.
- RPL: Der manuelle Flug mit Loopingunterstützung ist ein Sonderfall und ermöglicht, dass der Quadrocopter durch Umlegen eines Schalters ein Looping durchführt. Durch eine Makroanweisung werden Modus R6 und alle folgenden durch diesen Modus ersetzt.

Zum Einstellen des aktuell gültigen Betriebsmodus dienen 3 Schalter an der Fernsteuerung (vgl. Abbildung 6.10) sowie 3 Kommandos, die per PC verschickt werden. Die drei Schalter lauten:

- PSW: Der Powerschalter (engl. Power Switch, kurz PSW) startet und stoppt die Motoren. Auf der Fernsteuerung ist er mit Aux beschriftet (roter Punkt in Abbildung 6.10).
- HSW: Der Height-Switch (HSW) aktiviert die Höhenregelung. Auf der Fernsteuerung ist er mit Gear beschriftet (grüner Punkt in Abbildung 6.10).
- ASW: Der Autonomieschalter (engl. Automation Switch, kurz ASW) dient dem Umschalten in den autonomen Modus und der Aktivierung der autonomen Landung. Er hat drei Zustände: Aus, An und Landung. Aus Sicherheitsgründen muss der Schalter angestellt sein, damit in einen autonomen Betriebsmodus gewechselt werden kann. Auf der Fernsteuerung ist er mit FLT Mode beschriftet (blauer Punkt in Abbildung 6.10).

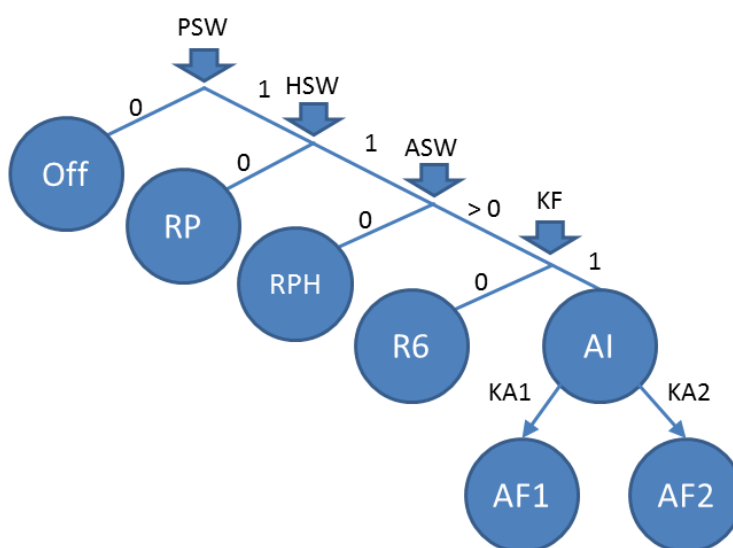


Abbildung 6.9: Steuerkontrolle (Entscheidungsgraph für den gültigen Betriebsmodus)



Abbildung 6.10: Fernsteuerung (RC) von Spektrum

Die drei Kommandos werden von der Bodenstation als Befehl per Funk verschickt. In der Flugsteuerung wird das entsprechende Flag aktiviert bzw. zurückgenommen. Die drei Kommandos lauten:

- KF: Dieses Kommando gibt den autonomen Flug frei (Kommando-Freigabe). Es dient der Sicherheit sowie dem Wechsel in den Modus AI.
- KA1: Dieses Kommando dient dem Wechsel in den Zustand AF1 und startet die Ausführung des Wegpunktfluges.
- KA2: Dieses Kommando dient analog zu KA1 dem Wechsel in den Zustand AF2 und startet die Ausführung der Wegpunktkommandierung.

Abbildung 6.9 fasst den Ablauf zur Bestimmung des Betriebsmodus in einem Entscheidungsgraph zusammen. Neben der Verwaltung des Betriebsmodus gehört zur Steuerkontrolle auch das Fusionieren der Steuereingaben. Dazu werden die Sollwertvorgaben für die Regler aus den unterschiedlichen Quellen (Positionsregler, Kollisionsvermeidung, PC, Fernsteuerung, Smartphone, etc.) aufbereitet und fusioniert.

### 6.6.2 Flugkontrolle (FlightCtrl)

Zur Aufgabe der Flugkontrolle gehört das Ausführen der Navigation, d.h. das Abarbeiten der gewünschten Wegpunktliste. Dabei greift die Flugkontrolle auf die Daten der Signalverarbeitung insbesondere der aktuell gültigen Position zurück. Diese Daten (Position, Höhe, etc.) werden von den jeweils verwendeten Sensoren zur Verfügung gestellt.

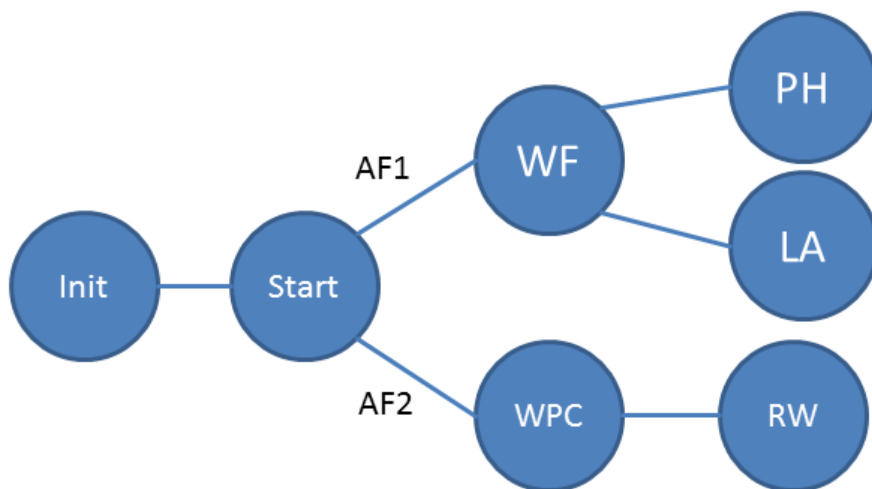


Abbildung 6.11: Zustandsmaschine zur Flugkontrolle

Die Flugkontrolle wird von der Steuerkontrolle gesteuert und nur in den Betriebsmodi AF1 und AF2 aufgerufen, welche auch das Verhalten der Flugkontrolle vorgeben. Zu Beginn befindet sich die Flugkontrolle stets im initialen Zustand *Init*, in dem Position, Höhe und aktueller Wegpunkt zurückgesetzt werden. Vom Zustand *Init* geht die Flugkontrolle im nächsten Sample in den Zustand *Start* über. Der Zustand *Start* wird solange gehalten, bis der Quadrocopter die vorgegebene Flughöhe erreicht hat. Nachdem die gewünschte Flughöhe bis auf 10cm angenähert wurde, wechselt die Flugkontrolle für den Betriebsmodus AF1 in den Zustand WF. In diesem Zustand wird der Wegpunktflug ausgeführt, d.h. die Wegpunktliste wird nacheinander abgearbeitet, wobei ein Wegpunkt erreicht ist, sofern die Abweichung in X und Y jeweils weniger als 25cm beträgt. Per

Kommando kann in den Zustand *PH* (Position halten) gewechselt werden, um den Wegpunktflug zu pausieren und später fortzusetzen. Nachdem der letzte Wegpunkt erreicht ist, wird in den Zustand *LA* (Landung) gewechselt und der Quadrokopter landet.

In vielen Szenarien mag zu Beginn der Mission nicht bekannt sein, welche Wegpunkte anzufliegen sind. Darüber hinaus können neue Informationen es erforderlich machen, während des Fluges in die Ausführung der Navigation einzugreifen. Dazu wurde die Flugkontrolle um die Wegpunkt-kommandierung (WPC) erweitert. Diese ist zum Teil auf dem Mikrocontroller und zum Teil auf dem onBoard-PC implementiert. Dabei ist der Mikrocontroller nur noch ausführendes Organ und entscheidet nicht mehr selbst, ob ein Wegpunkt erreicht ist und welcher Wegpunkt als nächstes anzufliegen ist. Diese Aufgaben übernimmt der onBoard-PC, welcher genug Rechenleistung für komplexe Aufgaben, wie z.B. das Mapping, zur Verfügung hat. Hinzu kommt die Unterstützung des gerichteten Fluges, d.h. der Quadrokopter kann sich vor dem Anfliegen eines neuen Wegpunktes in Richtung dieses Wegpunktes ausrichten. Dazu wechselt er zum Rotieren in den Zustand *RW*.

### 6.6.3 Höhensteuerung

Die Höhensteuerung steuert den Höhenregler und ist erforderlich für die Sonderfälle Start und Landung [P3, A02]. Aufgrund des beim Landen auftretenden Bodeneffekts, da die Sensoren keine geringeren Abstände als ca. 20cm verlässlich erkennen und weil beim Start das Schwebegas womöglich nicht an das aktuelle Gewicht angepasst ist, hat ein einfacher Höhenregler keine guten Ergebnisse geliefert. Abbildung 6.12 veranschaulicht die Zustandsübergänge der Höhensteuerung in üblicher Reihenfolge. Abhängig vom aktuellen Zustand greift die Höhensteuerung in die Höhenregelung ein.

Dabei befindet sich das System zunächst im Zustand *Start* und der Startregler wird ausgeführt. Durch eine stärkere Integration als üblich sorgt der Startregler dafür, dass das Schwebegas schneller angepasst wird. Gleichzeitig steigert er den P-Teil, damit die Soll-Höhe schnell erreicht wird. Nachdem die Soll-Höhe bis auf 10cm erreicht ist, wechselt das System in den Normalzustand.

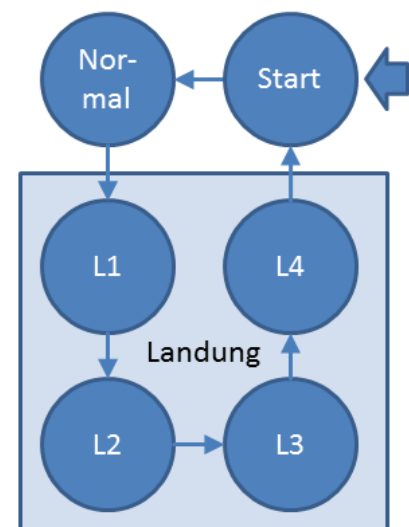


Abbildung 6.12: Zustandsmaschine zur Höhensteuerung

Wird die Landeregulation aktiviert, wechselt die Höhensteuerung in den Zustand *L1*. Im Zustand *L1* wird eine Soll-Höhe von 30cm eingestellt, damit der Quadrokopter sich senkt. Dieser Wert wurde gewählt, um nach unten einen Puffer zu haben und somit eine Kollision mit dem Boden zu vermeiden. Das war erforderlich, da die Sensoren minimal ca. 20cm messen und weil das System aus verschiedenen Höhen unterschiedlich schnell landet.

Fällt die Höhe unter 60cm, wechselt die Höhensteuerung in den Zustand *L2* und die Soll-Höhe wird auf 10cm gedrittelt. Fällt die Höhe unter 30cm, wechselt die Höhensteuerung in den Zustand *L3*. Der Quadrokopter ist nun fast gelandet. Da in diesem Bereich die Höhe nicht mehr akkurat gemessen werden kann, wird von einer Höhenregelung zu einer Höhensteuerung gewechselt, d.h. der Gesamtschub wird langsam sukzessive reduziert, so dass der Quadrokopter landet und auf dem Boden bleibt. Wie der Gesamtschub zu reduzieren ist, wurde empirisch optimiert. Unterschreitet der Gesamtschub eine minimale Schwelle, wird in den Zustand *L4* gewechselt, der Gesamtschub zu Null gesetzt und die Motoren werden je nach Implementierung ganz ausgeschaltet.

## 6.7 Weitere Funktionen zur Regel- und Steuerung

### 6.7.1 Objektsuche

Der AQopter18 ist in der Lage, die Anzahl an Objekten, hier rote Bälle, die auf dem Boden verteilt sind, autonom, d.h. nur unter Verwendung bordeigener Ressourcen, zu zählen und deren Positionen wiederzugeben. Diese Funktion wird Objektsuche genannt und im Weiteren näher beschrieben [P6].

Das Konzept der Objektsuche sieht vor, dass der Suchbereich vom Operator als quadratisches Feld vorgegeben wird. Dieser Bereich wird vom Quadrokopter autonom abgeflogen und mit Hilfe einer nach unten gerichteten Kamera wird die Anzahl roter Bälle gezählt. Gleichzeitig wird deren Position abgespeichert. Am Ende ergibt sich eine Karte, aus der die Lage der roten Bälle hervorgeht. Der Sichtbereich (engl. Field of View, FOV) ist dabei der Bereich, den die Kamera am Boden abdeckt.

Der virtuelle Sichtbereich (engl. Virtual FOV, VFOV) gibt den Bereich an, mit dem der Suchalgorithmus rechnet.

Die Wegpunktliste wird dabei zu Beginn der Suche automatisch vom Suchalgorithmus in Abhängigkeit des Suchbereichs, des VFOV und der Suchstrategie erstellt. Der Suchalgorithmus stellt sicher, dass der komplette Suchbereich zumindest einmal im Sichtbereich auftaucht. Somit wird quasi der komplette Suchbereich „abgescannt“ (vgl. Abbildung 6.13).

Der Sichtbereich kann als Parameter eingestellt werden. Je kleiner der Sichtbereich ist, desto mehr Wegpunkte werden generiert und desto engmaschiger wird die Suche (vgl. Abbildung 6.15). Die Suchstrategie kann ebenfalls eingestellt werden. Aktuell sind zwei Suchstrategien implementiert. Die Tiefensuche (engl. Depth First Search, DFS) sucht zunächst den kompletten Suchbereich im Äußeren ab und pflanzt sich dann zirkulär nach innen fort. Dagegen deckt die Breitensuchen (engl. Breadth First Search, BFS) zunächst die Bereiche nahe dem Startpunkt ab und fährt sukzessive mit den weiter entfernten Bereichen fort.

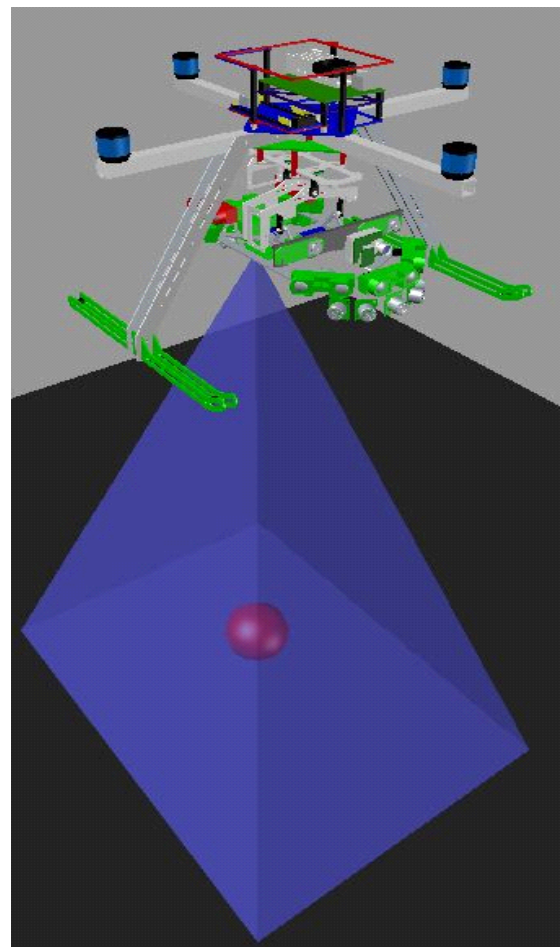


Abbildung 6.13: Konzept Objektsuche mit Sichtbereich (FOV) am Boden (CAD-Zeichnung Sascha Dechend)

Mehrfachzählungen desselben Objektes werden durch Maskierung verhindert, d.h. über jedes bereits gefundene Objekt (Ziel, engl. Target) wird an der Position eine Maske gelegt. Objekte, die im Maskierbereich gefunden werden, werden nicht aufgenommen. Mit Maskierbereich ist die Distanz gemeint, die das Quadrat um das Ziel definiert (vgl. Abbildung 6.14).

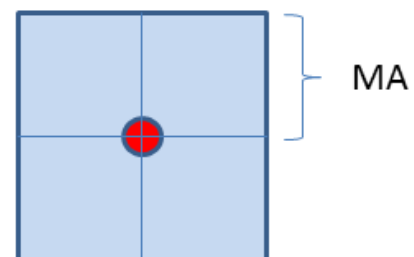


Abbildung 6.14: Maskierbereich MA um Target (roter Punkt)

Somit lässt sich das Problem der Objektsuche aufteilen in die Wegpunktsuche und die Zielsuche. Die Wegpunktsuche besteht aus dem Abfliegen der Wegpunktliste zum Absuchen des Suchbereichs. Die Zielsuche verwaltet die Ziele und fügt neue hinzu. Das Ergebnis der Objektsuche ist eine Liste mit Zielen bestehend aus Positionen  $\vec{P}_Z$ . Den Ablauf der Zielsuche veranschaulicht Abbildung 6.16.

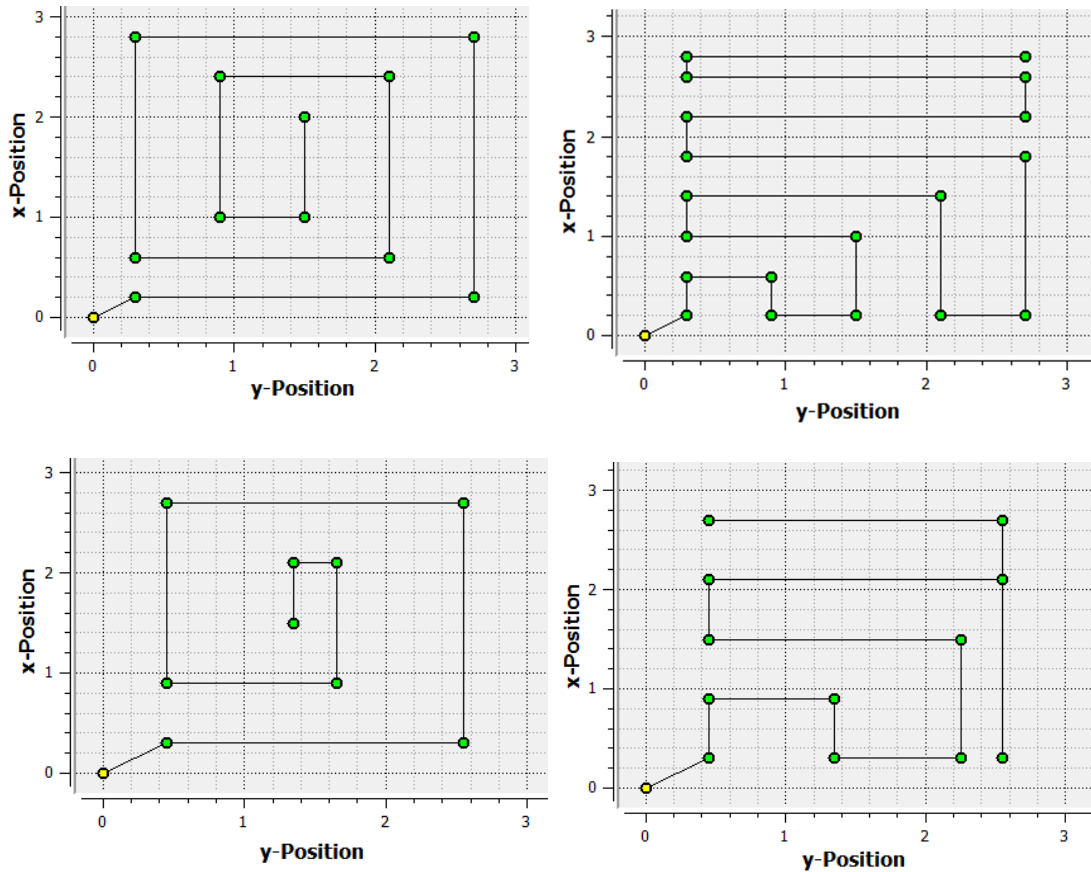


Abbildung 6.15: Wegpunkte der Tiefensuche (DFS) mit VFOV 40x60 (links oben) und VFOV 60x90 (links unten) und Breitensuche (BFS) mit VFOV 40x60 (rechts oben) und VFOV 60x90 (rechts unten)

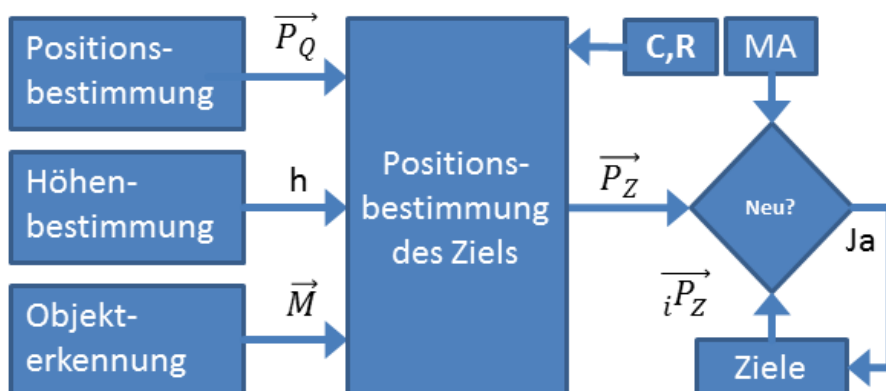


Abbildung 6.16: Ablauf der Zielsuche bei Objekterkennung (Teil der Objektsuche)

Für jedes von der Objekterkennung erkannte Ziel wird mittels Formel 6.7 dessen Position  $\vec{P}_Z$  bestimmt. Die Position  $\vec{P}_Z$  ist die Summe aus der Position  $\vec{P}_R$  des Objekts relativ zum Quadrocopter,

der Position  $\vec{P}_Q$  des Quadropters und dem Kamera-Offset  $\vec{C}_O$ .

$$\vec{P}_Z = \vec{P}_R + \vec{P}_Q + \vec{C}_O \quad \text{Formel 6.7}$$

Die Position  $\vec{P}_R$  wird berechnet mit Hilfe der Höhe  $h$  aus der Höhenbestimmung, dem Mittelpunkt  $\vec{M}$  des Ziels in Pixeln aus der Objekterkennung sowie weiteren Parametern wie der Kalibrierungshöhe  $C_h$ , dem FOV  $C_x$  in  $x$ - und  $C_y$  in  $y$ -Richtung (Formel 6.8 bis Formel 6.10) und der Auflösung ( $R_x \times R_y$ ) der Kamera.

$$\vec{M} = \begin{pmatrix} M_x \\ M_y \end{pmatrix} \quad \text{Formel 6.8}$$

$$\vec{R} = \begin{pmatrix} \frac{M_x}{R_x} - 0,5 & \frac{M_y}{R_y} - 0,5 \end{pmatrix} \quad \text{Formel 6.9}$$

$$\vec{P}_R = \frac{h}{c_h} \cdot \vec{R} \cdot \begin{pmatrix} C_x & 0 \\ 0 & C_y \end{pmatrix} \quad \text{Formel 6.10}$$

## 6.7.2 Kippgelenk

Kippbewegungen (RP) des Quadropters führen zu Messfehlern der Sensoren. Dies betrifft die nach unten gerichteten Sensoren, insbesondere die OF Sensoren, als auch alle Sensoren, die zur Seite messen. Um die Sensorplattform, bestehend aus Bodenplatte (vgl. Abbildung 6.17) und Abstandssensorik, in der Waagerechten zu halten und somit die Auswirkungen der genannten Fehler zu reduzieren, wurde ein Kippgelenk, eine Art kardanische Aufhängung (engl. Gimbal) konstruiert. Weit verbreitet ist dieser Lösungsansatz als Kameraaufhängung von Quadroptern.

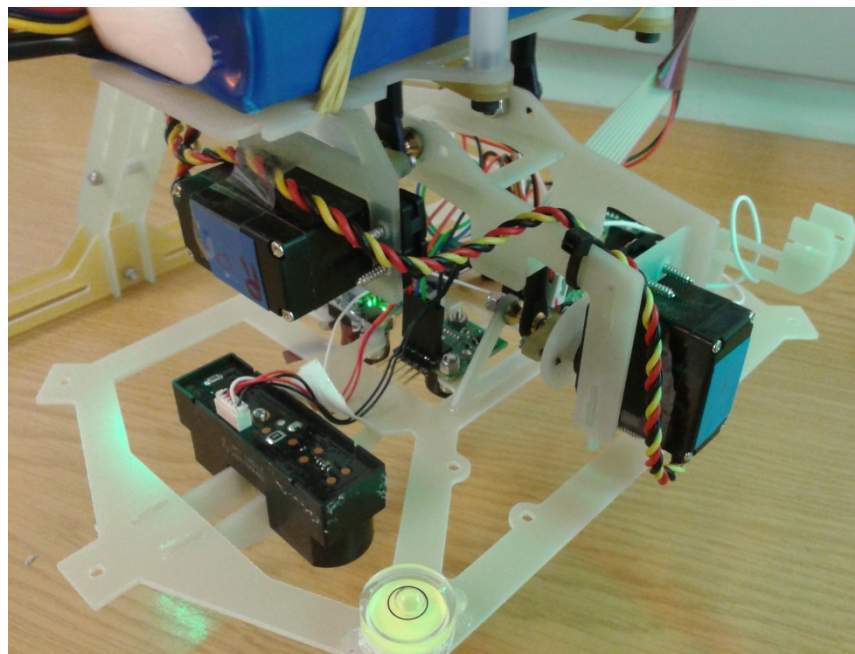
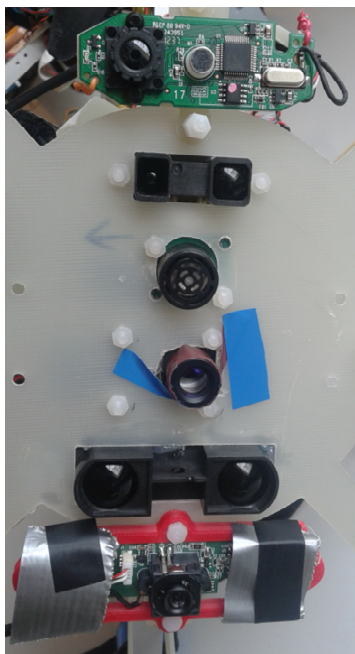


Abbildung 6.17: Bodenplatte (links) und Kippgelenk (rechts) von unterschiedlichen AQopter18-Versionen



Das Kippgelenk wird von zwei Servomotoren gesteuert, die es in zwei Freiheitsgraden kippen können. Die Servomotoren werden per PWM angesteuert. Das PWM Signal wird entsprechend der Lage des Quadropters manipuliert und balanciert so stets die Kippbewegungen des Quadropters aus.

### 6.7.3 Gerichteter Flug

Der gerichtete Flug (engl. Directed Flight) ist eine besondere Fortbewegungsart. Darunter ist zu verstehen, dass der Quadropter bei einer Translation stets zunächst in Flugrichtung giert und sich dann vorwärts bewegt. Somit fliegt der Quadropter wie ein Flugzeug, d.h. ausgerichtet mit der Vorderseite nach vorne. Ein solches Verhalten ist bei einem Quadropter zwar grundsätzlich nicht erforderlich, da dieser nicht gieren muss, um seine Flugrichtung zu ändern. Der gerichtete Flug wird dann aber erforderlich, wenn der Quadropter eine Vorzugsrichtung besitzt. Dies kann dadurch bedingt sein, dass nur an der Vorderseite eine bestimmte Sensorik (z.B. Kamera) angebracht ist und diese in Flugrichtung zeigen soll.

Von Interesse für den AQopter18 ist dies, da der Quadropter in die Lage zu versetzen ist, autonom gerichtet zu fliegen. Dies ist erforderlich, da Sensoren wie die PMD Kamera sehr kostspielig sind. Die PMD Kamera ist für hochgenaue Abstandsmessungen, wie z.B. beim Durchfliegen einer Tür erforderlich. Eine 360° Abdeckung mit PMD ist dagegen nicht unbedingt notwendig. Es bietet sich also an, eine PMD Kamera an der Vorderseite zu montieren. Wird die Kamera auf einer anderen Seite benötigt, giert der Quadropter im Betrieb.

Dies stellt grundsätzlich kein Problem dar, sofern die verwendete Sensorik Gier-Bewegungen unterstützt. Der verwendete OF Sensor, der ADNS-3080, ist dazu jedoch nicht ausgelegt, sondern erwartet ausschließlich translatorische Bewegungen. Gier-Bewegungen werden als Translationen fehlinterpretiert und führen zu einem Positionsfehler. Abhilfe schafft das von Eric Reinthal [A23] entwickelte Fourier Tracking (vgl. Seite 37), welches eine Voraussetzung für den autonomen gerichteten Flug schafft.

Da das Fourier Tracking selbst sehr rechenintensiv ist, wurde eine Lösung geschaffen, die das Fourier Tracking nur aktiviert, wenn dieses erforderlich ist. Ein weiterer Nachteil der hohen Rechenlast ist die vergleichsweise niedrige Wiederholrate des Fourier Tracking von 100ms. Daher werden beide Sensorsysteme dynamisch fusioniert [P8]. Abbildung 6.18 veranschaulicht das Konzept zur dynamisch gesteuerten Datenfusion.

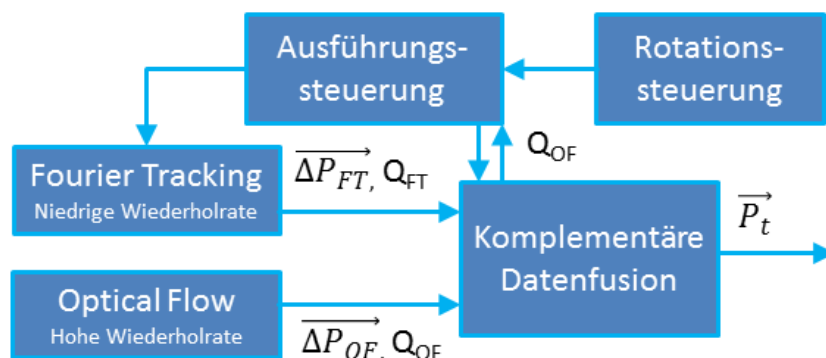


Abbildung 6.18: Konzept der dynamisch gesteuerten Datenfusion für den gerichteten Flug

Die Ausführungssteuerung aktiviert bzw. deaktiviert das Fourier Tracking. Dieses wird während jeder Rotation um die Gier-Achse sowie bei schlechten Lichtverhältnissen, die ein Problem für den ADNS-3080 darstellen, aktiviert. Die mittels Fourier Tracking bestimmte Positionsänderung  $\overrightarrow{\Delta P_{FT}}$  und die Positionsänderungen  $\overrightarrow{\Delta P_{OF}}$  des OF Sensors werden komplementär fusioniert (vgl. Formel 6.11).

$$\overrightarrow{P}_t = \overrightarrow{P}_{t-1} + \alpha \cdot \overrightarrow{\Delta P_{OF}} + (1 - \alpha) \cdot \overrightarrow{\Delta P_{FT}} \quad \text{Formel 6.11}$$

Der Faktor  $\alpha$  hängt dabei von den normierten Qualitätswerten  $Q_{FT}$  und  $Q_{OF}$  der Sensoren ab. Ist die Qualität des ADNS-3080 unterhalb eines Schwellwerts, so wird der Faktor  $\alpha$  zu Null gesetzt, womit der ADNS-3080 nicht in die Position  $\overrightarrow{P}_t$  einfließt. Versagt das Fourier Tracking oder ist es deaktiviert, so wird der Faktor  $\alpha$  zu Eins gesetzt. Somit wird allein dem ADNS-3080 vertraut. In allen anderen Fällen errechnet sich der Faktor  $\alpha$  aus den Qualitätswerten (Formel 6.12).

$$\alpha = \frac{Q_{OF}}{Q_{OF} + Q_{FT}} \quad \text{Formel 6.12}$$

Die Rotationssteuerung steuert die Rotation und teilt der Ausführungssteuerung für das Fourier Tracking mit, wann eine Rotation beginnt und wann sie abgeschlossen ist. Gleichzeitig dient die Rotationssteuerung der Steuerung der Rotationsgeschwindigkeit. Dies ist erforderlich, da das Fourier Tracking schnelle Rotation nicht korrekt misst. Der Yaw-Regler schafft eine 90°-Drehung in ca. 1s. Durch die Rotationssteuerung wird die Drehgeschwindigkeit auf ca. 25°/s herabreguliert.

#### 6.7.4 Looping-Steuerung

Für einige weiterführende Anwendungsszenarien ist ein sehr dynamisches Flugverhalten erforderlich. Es ist denkbar, dass das System später einmal an einer Wand waagrecht landen können soll. Eine Vorleistung dazu stellt das Looping dar. Auch sollte untersucht werden, ob für die Orientierungsbestimmung und Lageregelung Einschränkungen existieren, die bei kritischen Fluglagen zu einem Versagen führen und somit einen Absturz verursachen. Aus diesen Gründen wurde eine Looping-Steuerung implementiert und erprobt.



Abbildung 6.19: AQuadcopter18 beim Looping

Mit der Looping-Steuerung kann der AQuadcopter18, wenn im Flug per Knopfdruck aktiviert, automatisch ein Looping durchführen. Während des Loopings werden andere Reglerparameter verwendet, um das Looping schnell und stabil durchzuführen. Die Steuerung ist als Zustandsmaschine realisiert, die fünf Zustände unterscheidet. Der Anfangszustand ist der initiale Zustand, der den Looping entsprechend der gewünschten Richtung nach Eingabe des Kommandos von der Fernsteuerung aktiviert. Dazu wird der Soll-Wert (Winkel der jeweiligen Achse) der Lageregelung auf +120° bzw. -120° gesetzt. Abhängig von der Richtung wird in den zweiten bzw. dritten Zustand gesprungen. Nach Erreichen des Winkels (mit 60° Toleranz) wird der neue Soll-Wert in Zustand 2 auf -120° bzw. in Zustand 3 auf +120° gesetzt und in den vierten Zustand gesprungen. Nachdem auch dieser erreicht

ist, wird der Soll-Wert auf 0° gesetzt. Erreicht der Winkel +/- 30° wird in den Anfangszustand gesprungen. Die Reglerparameter werden zurückgesetzt und der Looping ist abgeschlossen.

Die Fotos aus Abbildung 6.19 illustrieren den Ablauf beim Looping, wobei der aufgrund des entzogenen und teilweise negativen Schubes verursachte Höhenverlust besonders im letzten Bild deutlich wird. Dem kann durch Schub-Zugabe vor dem Looping entgegengewirkt werden.

### 6.7.5 PC Steuerung

Zur Steuerung des Quadropters per PC stehen sechs verschiedene Alternativen zur Verfügung. Auf die beiden Möglichkeiten der Steuerung via OTS und GPS wird in den beiden folgenden Abschnitten separat eingegangen. Die vier weiteren Möglichkeiten werden in diesem Abschnitt kurz beschrieben:

- **Manuelle Flugsteuerung:** Die Lage des AQopterI8 kann am PC mit der Maus sowohl per 3D-Drehung der 3D-Orientierungsanzeige also auch per Scroll-Bar manipuliert werden. Damit lässt sich der Quadropters im Flug steuern und die Lage in drei Dimensionen anzeigen. Zu Testzwecken lässt sich darüber hinaus das aktuell gültige Quaternion der Orientierungsbestimmung manipulieren, um z.B. Driftfehler zu simulieren.
- **Wegpunktsteuerung:** Per Mausklick lassen sich am PC Wegpunkte generieren und per Button einzeln oder als Liste an den Quadropters schicken. Ursprünglich wurde das dazugehörige Interface von Michael Strohmeier im Rahmen seiner Bachelorarbeit implementiert und diente der Positionssteuerung unter Verwendung des OF Sensors [A04].
- **Objektsuche:** Das Interface zur Objektsuche wird per PC bedient und darüber wird der Quadropters gesteuert. Als Teilkomponente der Objektsuche [S. 118] lässt sich ein Suchbereich festlegen und daraus eine Wegpunktliste generieren, die an den Quadropters per Bluetooth übertragen werden kann. Das Starten, Abfliegen (Suchen) und Laden erfolgt dann mit einem Kommando, womit der gesamte Ablauf autonom ist.
- **Wegpunktkommandierung:** Der gerichtete Flug und die Wegpunktkommandierung werden ebenfalls per PC gesteuert (vgl. S. 121).

### 6.7.6 GPS / GNSS – Steuerung via Kartenmaterial

Der AQopterI8 ist zwar für den Einsatz in Gebäuden konzipiert, in denen meist kein GNSS-Empfang besteht, jedoch ist eine Reihe an Szenarien denkbar, bei denen ein GNSS-Empfänger hilfreich wäre. Dazu gehört der Anflug ans Gebäude sowie beim Wechsel zwischen Gebäudeteilen. Der Quadropters könnte einen Gebäudeteil z.B. über ein Fenster verlassen und einen anderen Bereich über ein weiteres Fenster betreten. Zwischen den Fenstern würde sich dann eine Navigation mittels GPS anbieten.

Der AQopterI8 kann mittels GPS seine Positionen halten sowie neue Positionen anfliegen [A28]. Die Positionen werden per Mausklick auf einer Weltkarte (Google Maps, Open Layer) als Wegpunkte generiert und an den Quadropters via Bluetooth übertragen. Andrea Tüchert hat in ihrer Bachelorarbeit das dazu nötige Interface entwickelt [vgl. Abbildung 6.20].

Die GPS-Positionen sind üblicherweise angegeben in geographischen Koordinaten (Breite, Länge, Höhe; engl. LLH; Latitude, Longitude, Height). Damit der Quadropters mittels GPS seine Position regeln kann, muss diese in das lokale Koordinatensystem des Quadropters transformiert werden. Dazu werden die Positionen zunächst in NED-Koordinaten (engl. North East Down) transformiert und dann mit Hilfe der Nord-Ausrichtung des Quadropters in lokale Quadropters-Koordinaten

umgerechnet.

Die Nord-Ausrichtung wird einmal bei der Initialisierung mittels Magnetometer bestimmt sowie fortlaufend aus der Startorientierung und dem aktuellen Gier-Winkel berechnet. Somit kann sich der Quadrocopter gieren, d.h. um die z-Achse rotieren, und simultan Positionen mittels GPS halten und anfliegen.

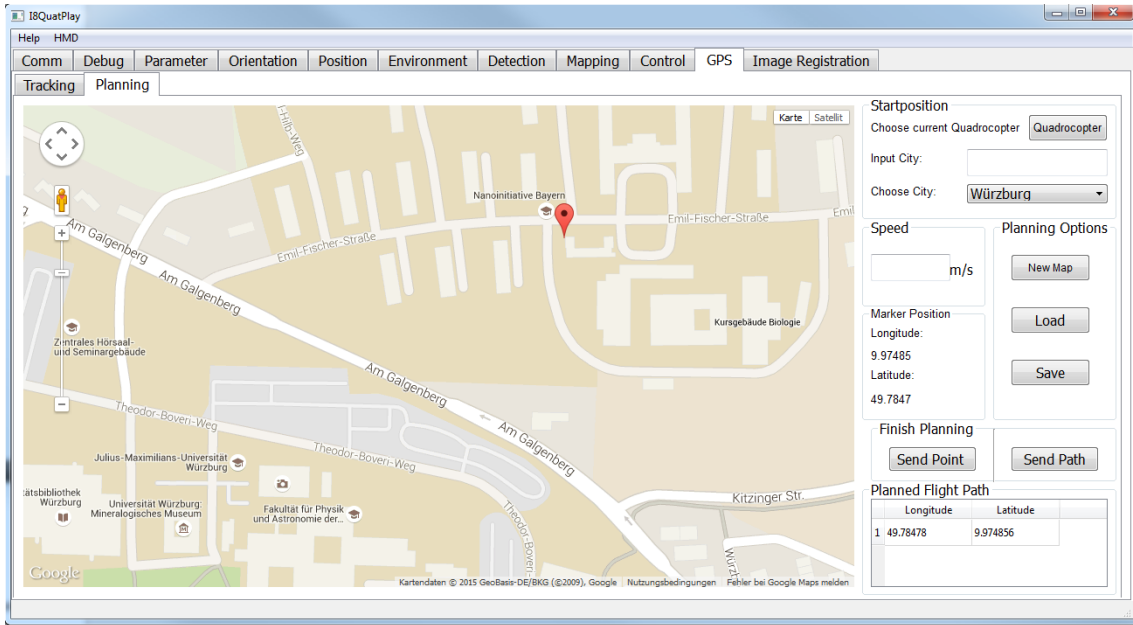


Abbildung 6.20: AQopterI8 GPS-Steuerung [Andrea Tüchert, A28]

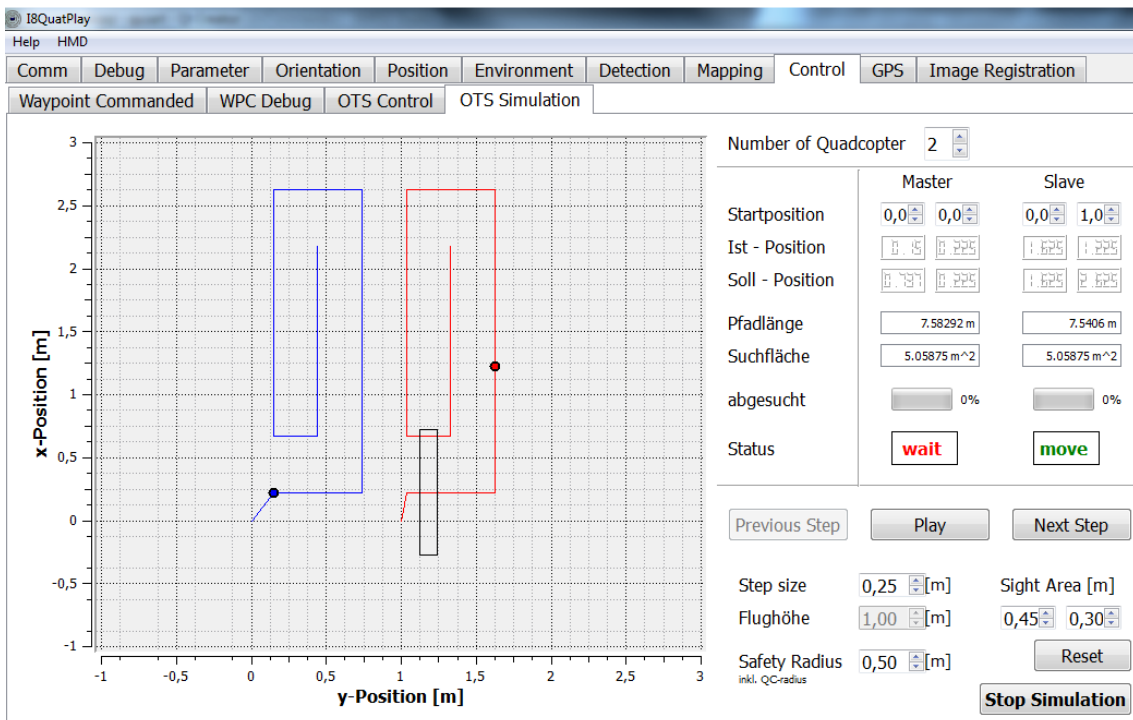


Abbildung 6.21: AQopterI8 OTS-Steuerung [Arthur Scharf, A29]

### 6.7.7 OTS Steuerung

Für einige sehr komplexe autonome Aufgaben ist eine hochgenaue Positionsbestimmung erforderlich. Beispiele sind der Einsatz des Manipulators zum Greifen und kooperative Manöver mehrerer Quadrokopter. Das OTS (Optische Trackingsystem) bietet zwar keine Lösung für ein autonomes System, liefert aber hochgenaue Positionen und kann daher bei der Evaluierung und Analyse des Systems hilfreich sein. Mit dem OTS können Methoden und Verfahren zur Regel- und Steuerung unabhängig von den Problemen der sonstigen Sensorik untersucht werden. Dazu wird die Positionsbestimmung des autonomen Systems durch die des OTS ersetzt. Dies erlaubt eine getrennte Analyse der Subsysteme.

Aus diesem Grund wurde das OTS zur Positionsregelung in den AQopterI8 integriert. Das Interface zur Steuerung wurde von Arthur Scharf im Rahmen seiner Bachelorarbeit entwickelt (Abbildung 6.21) und ermöglicht eine Steuerung basierend auf einer Regelung mittels OTS [A29].

### 6.7.8 Smartphone- und Miniatur-Steuerung

Eine Steuerung sollte möglichst ergonomisch sein, um dem Benutzer die Verwendung des Systems zu erleichtern und die Einsatzmöglichkeiten nicht unnötig zu begrenzen. Hierbei sind Größe und Einfachheit besondere Kriterien. Die Steuerung sollte klein sein und einfach zu bedienen. Daher wurden für den AQopterI8 zwei weitere Steuerungsalternativen, die Steuerung per Smartphone und Miniaturgerät, entwickelt. Bei der Steuerung per Smartphone berechnet ein Programm (App) mit Hilfe der eingebauten IMU auf dem Smartphone dessen aktuelle Lage und sendet diese per Funk an den Quadrokopter. Die Lage des Smartphones geht als Sollwert in die Lageregelung ein. Wird das Smartphone gekippt, so kippt sich der Quadrokopter entsprechend und kann darüber gesteuert werden. Dieser Ansatz ist nicht neu. Die App für den AQopterI8 wurde von Maximilian Becker im Rahmen eines Praktikums entwickelt.



Abbildung 6.22: AQopterI8 auf dem Campus Festival

Auf dem Campus Festival 2014 der Universität Würzburg hatten Kinder die Möglichkeit, mit Hilfe des Smartphones den Quadrokopter zu steuern, wobei die Höhe geregelt war (Abbildung 6.22). Wurde das Smartphone mehr als  $90^\circ$  um die z-Achse gedreht, landete der Quadrokopter selbstständig. Die Aufgabe der Kinder bestand darin, durch Steuerung des Quadrokopters zwei Plastikbälle von umgedrehten Blumentöpfen zu pusten und hinterher auf einem blauen Teppich zu landen. Die Begeisterung bei den Kindern war so groß, dass das System bis auf Pausen zum Akkuwechsel den ganzen Tag über im Einsatz war. Zur Sicherheit konnte der Quadrokopter jederzeit per Fernsteuerung

gesteuert werden.

Die Miniatur-Steuerung ist eine analoge Realisierung bestehend aus Mikrocontroller, IMU und Funkmodul und wurde von Hasan Tuncay im Rahmen seiner Bachelorarbeit entwickelt [A27]. Mit ihr lässt sich der Quadrocopter nach demselben Prinzip steuern. Um eine sanfte Steuerung zu ermöglichen, geht in beiden Systemen jeweils nur der halbe Winkel als Sollwert ein. Zudem wird der Winkel aus Sicherheits- und Stabilitätsgründen saturiert.

### 6.7.9 HMD Steuerhilfe

Je nach Anwendungsszenario sind zur Steuerung Video- und Telemetriedaten des Quadrocopters erforderlich bzw. hilfreich. Die Anzeige der Daten sollte dabei ergonomisch erfolgen. Dies kann durch Verwendung einer Videobrille (engl. Head-Mounted-Display, kurz HMD) erreicht werden. Dabei kann der Operator den Quadrocopter aus dessen Sicht (engl. First Person View, kurz FPV) steuern. In das Videobild können dann weitere, aufbereitete Telemetriedaten eingeblendet werden wie z.B. Abstände, Lage und Akkustand.

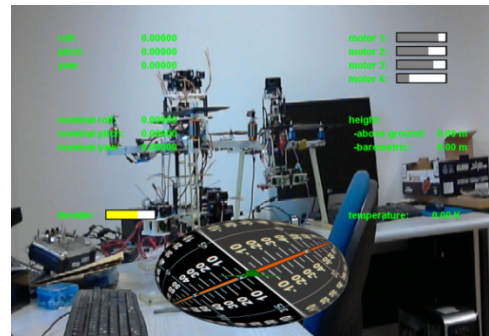


Abbildung 6.23: HMD-Telemetrie-Anzeige [ A25]

Eine solche Lösung hat Deniz Caliskan für den AQopter18 im Rahmen seiner Bachelorarbeit und unter Verwendung der Oculus Rift [125] implementiert. In das Kamerabild können weitere Daten, wie z.B. die aktuelle Lage, eingeblendet werden (Abbildung 6.23).







*Achte auf Deine Gedanken, denn sie werden Deine Worte.  
Achte auf Deine Worte, denn sie werden Deine Taten.  
Achte auf Deine Taten, denn sie werden Deine Gewohnheiten.  
Achte auf Deine Gewohnheiten, denn sie werden Dein Charakter.  
Achte auf Deinen Charakter, denn er wird Dein Schicksal.*  
[Talmud]

# 7 Softwarearchitektur

The image displays a collection of software development tools and logos. At the top right, there is a blue circular logo with a white maze pattern. Below it is the green Qt logo. In the center, a screenshot of a C++ code editor shows a Qt application with a TCP socket interface. At the bottom left, the AVR32 Studio IDE is shown with a 3D model of a quadcopter and the text 'AVR32 STUDIO Loading Workbench'. The background is a light blue gradient.

## 7.1 Überblick Softwarearchitektur

Dieses Kapitel behandelt die Software des AQopter18. Dabei wird unterschieden zwischen Low Burden (Niedrige Last) und High Burden (Hohe Last) Software, d.h. Software mit wenig bzw. viel Rechenlast. Die Low Burden Software läuft auf einem AVR Mikrocontroller und übernimmt alle Aufgaben, die harter Echtzeit unterliegen und dabei mit wenig Rechenkapazität auskommen. Dazu gehören in diesem Fall primär das Auslesen der Sensorik (ausgenommen Sensoren mit USB-Interface), die Durchführung der Regelung und das Ansteuern der Motoren. Diese Software wurde einmal unter Verwendung eines klassischen Designs, einer großen Whileschleife (Big Loop) sowie mit Hilfe des Betriebssystems Rodos umgesetzt.

Die High Burden Software läuft dagegen auf einem High End Rechner, in diesem Fall einem PC mit ausreichend Rechenkapazität und Speicher für Aufgaben aus den Gebieten Bildverarbeitung und Mapping. Dies kann der on-Board PC LP-180 sein oder ein entfernter PC, der als Bodenstation dient. Somit ist der Einsatz der Software sehr flexibel. Dieser Teil der Software bietet gleichzeitig eine auf Qt-basierte GUI (engl.: Graphical User Interface, grafische Benutzerschnittstelle) als Operator- und Piloteninterface.

## 7.2 Klassisches Softwaredesign

### 7.2.1 Überblick (Klassisches Softwaredesign)

Abbildung 7.1 stellt grob vereinfacht das klassische Softwaredesign mit einem seriellen Programmablauf dar. Zur Initialisierung der Hardware gehören alle in bzw. durch Software zu tätigen Initialisierungen von Hardwarekomponenten. Darunter fällt das Initialisieren und Einstellen der CPU, der Interrupts und dazugehörigen Interrupt-Tabelle, des Displays, der Buttons, der ADCs, der LEDs, des Timer-Counters sowie der Schnittstellen USART, SPI und I<sup>2</sup>C (TWI). Des Weiteren sind die Aktuatoren und Sensoren, namentlich die Servos (PWM) und Brushless-Controller, die Ultraschallsensoren, die Infrarotsensoren, die IMU, der Luftdrucksensor, der ADNS-3080 und der GPS-Receiver zu initialisieren. Zur Initialisierung der Software gehört das Initialisieren der Filter (Kalman, Mittelwert, Ausreißer, CQF), der Regler, der Steuerung, der Hinderniserkennung und der Kollisionsvermeidung.



Abbildung 7.1: Klassisches Softwaredesign

Die Kalibrierung besteht aus der Kalibrierung der IMU (S. 88) und der Kalibrierung der Fernsteuerung. Die Fernsteuerung wird aus Sicherheitsgründen kalibriert. Dies bedeutet, es werden die initialen Hebelstellungen der Fernsteuerung bei Programmstart ausgelesen und überprüft, ob sich alle im Ausgangszustand (Aus) befinden. Befindet sich ein Hebel nicht in der korrekten Stellung, so schlägt die Kalibrierung fehl. Damit wird verhindert, dass bei Programmstart ungewollt Funktionen aktiviert sind, die z.B. die Motoren oder die Höhenregelung einschalten. Unfälle, wie z.B. ein ungewolltes Starten des Systems, konnten damit vermieden werden und sind nicht vorgekommen.

Nach der Kalibrierung befindet sich das System mit einer Wiederholrate von 100Hz in der großen While-Schleife. Zur Signalverarbeitung gehören die Orientierungsbestimmung (IMU), die Höhenbestimmung, die Positionsbestimmung und die Hinderniserkennung. Zur Regel- und Steuerung gehören die Regler, die Kollisionsvermeidung und das Flight Management. Hinzu kommen weitere Funktionen für Telemetrie und Telekommando.

Die Software wurde in C mit AVR32 Studio entwickelt und besteht insgesamt aus 49 Modulen und ca. 15.000 Zeilen Code.

### 7.2.2 Timing-Diagramm des klassischen Softwaredesigns

Wie bereits erwähnt, ist die Darstellung aus Abbildung 7.1 eine grobe Vereinfachung und soll lediglich einen prinzipiellen Einblick ermöglichen. Bei dem Konzept der seriellen Ausführung besteht die Schwierigkeit darin, dass die insgesamt erforderliche Rechendauer aus der Summe aller Funktionen (Signalverarbeitung, Steuerung, Regelung) besteht und diese die Sampledauer von 10ms nicht überschreiten darf. Anderenfalls versagt die Regelung, was zu einem schlechten Flugverhalten bis hin zu Abstürzen führen kann. Da es aufgrund zunehmender Funktionen nicht mehr möglich war, alle Funktionen innerhalb von 10ms auszuführen, musste eine neue Lösung gefunden werden. Das Problem wurde durch folgende Modifikationen im Timing gelöst.

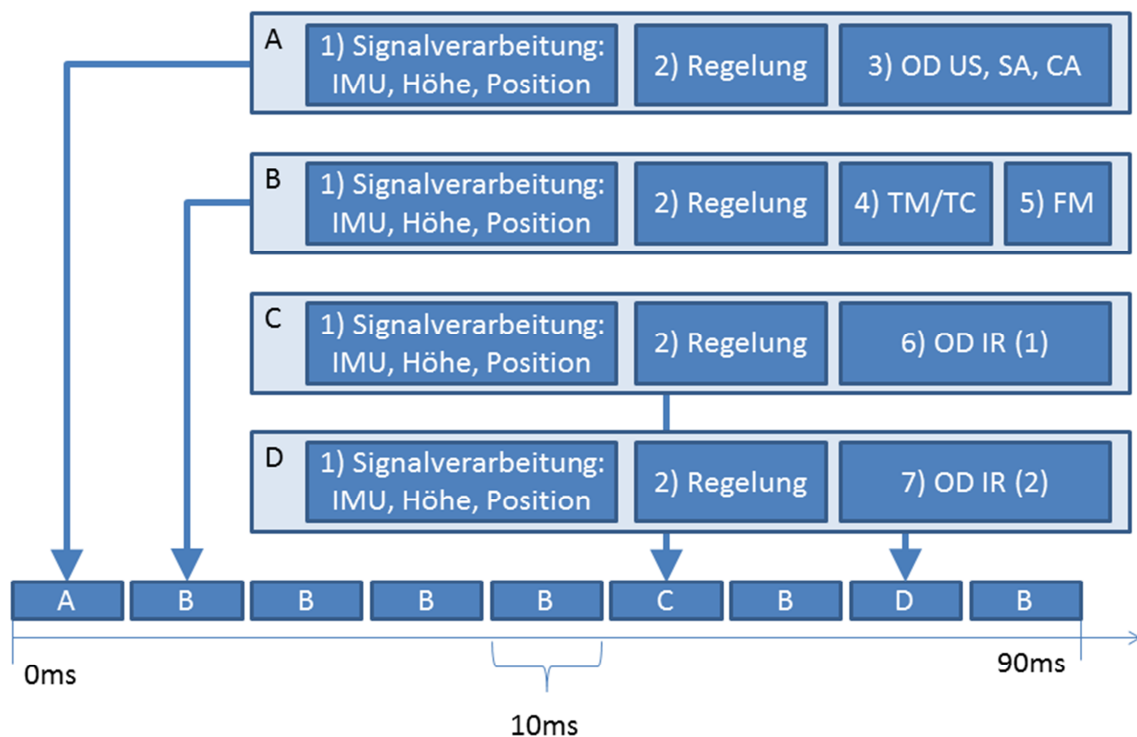


Abbildung 7.2: Timing-Diagramm des klassischen Softwaredesigns

Im Timing-Diagramm (Abbildung 7.2) wird die exakte zeitliche Abfolge der 7 wesentlichen Funktionsblöcke deutlich. Diese Blöcke bilden die 4 Gruppen A, B, C und D. Alle 10ms werden die Funktionsblöcke einer Gruppe ausgeführt, wobei sich die Ausführungsreihenfolge insgesamt nach 90ms identisch wiederholt.

Die Signalverarbeitung zur Orientierungs-, Höhen- und Positionsbestimmung mit anschließender Regelung erfolgt alle 10ms. Üblicherweise, d.h. in 6 von 9 Durchläufen, folgen die Funktionen zum

Versenden und Verarbeiten der Telemetrie und Telekommandos sowie die Funktionen des Flight Managements (Gruppe B). Eine Ausnahme bilden die Funktionen zur Hinderniserkennung (OD), Situationserfassung (SA) und Kollisionsvermeidung (CA). Deren Ausführungszeit aufaddiert, überschreitet die gewünschte Wiederholrate von 10ms. Dies liegt daran, dass das Auslesen der IR-Sensoren ungewöhnlich lange dauert. Daher wird das Auslesen der OD Sensoren in drei Blöcke eingeteilt:

- OD US für die Ultraschallsensoren zur Hinderniserkennung (Gruppe A)
- OD IR (1) für die langen IR Sensoren zur Hinderniserkennung (Gruppe C)
- ID IR (2) für die kurzen IR Sensoren zur Hinderniserkennung (Gruppe D)

Die Fusion der Sensordaten, die Situationserfassung (SA) und die Kollisionsvermeidung wird nach OD US ausgeführt.

Das Timing-Diagramm wurde in 9 Blöcke aufgegliedert, weil die Wiederholdauer für OD, SA und CA 90ms beträgt. Die US Sensoren benötigen 90ms zur Messung, wobei zu berücksichtigen ist, dass 12 Sensoren zum Einsatz kommen. Dabei agieren die vier jeweils 90° versetzten Sensoren simultan und jeder dieser drei 4er-Gruppen arbeitet 30ms versetzt. Thilo Müller hat diesen Zusammenhang in seiner Diplomarbeit ausgiebig analysiert. Das Ergebnis der Untersuchungen ist eine Wiederholrate von 90ms als optimales Trade-Off zwischen Geschwindigkeit (geringer Wiederholdauer) und Cross-Talk [A01].

Um das Timing permanent kontrollieren zu können, wurde eine Methode implementiert, die das Timing überprüft und, sofern es nicht eingehalten wird, eine Nachricht verschickt.

Ein großer Vorteil des klassischen Softwaredesigns ist die durch serielle Befehlsausführung gewonnene Übersichtlichkeit. Dies erleichtert die Fehleranalyse, da auch der Mensch seriell und nicht simultan denkt. Der Ablauf der Funktionen folgt stets einer festgelegten Reihenfolge. Damit lassen sich Fehler auch durch zeitliche Abfolge sehr einfach analysieren und zuordnen. Der große Nachteil liegt in der geringen Flexibilität bei der Befehlsausführung und der zeitlichen Abhängigkeit prinzipiell voneinander unabhängiger Funktionen. So lässt sich die Wiederholrate einzelner Prozesse nicht ohne weiteres beliebig einstellen. Für diese Arbeit war die verwendete serielle Befehlsausführung vollkommen ausreichend, wenngleich dies nicht unbedingt die eleganteste Lösung darstellt. Eine Abhilfe der genannten Problematik liefert die Verwendung eines Echtzeitbetriebssystems.

### 7.2.3 Flexibilität durch Modularität

Die Software ist dank ihres modularen Aufbaus sehr flexibel in der Verwendung. Dies ist eine Voraussetzung für die simultane Entwicklung der Subsysteme des AQopter18-Projekts. Aktuell existieren 10 Instanzen des AQopter18 (Abbildung 7.3). Diese Anzahl ist erforderlich oder zumindest sehr hilfreich, um ohne großen Umbauaufwand verschiedene Lösungen unabhängig voneinander simultan entwickeln, optimieren und evaluieren zu können. Darüber hinaus ermöglichen es die unterschiedlichen Systeme, insbesondere im Fehlerfall, diese unmittelbar miteinander vergleichen zu können. Dadurch wird die Fehlersuche stark vereinfacht und es können einfacher Hardware- von Softwarefehlern abgegrenzt werden.

Die Herausforderung liegt dabei weniger in der Hardware, da diese mittlerweile vergleichsweise

günstig zu beschaffen und schnell aufgebaut ist, sondern in der Software. Damit durch separate Softwarestränge kein Overhead entsteht, gibt es für alle Instanzen dieselbe Software. Neue Entwicklungen werden, wie es heute üblich ist, als Branch (Zweig) erstellt und bei Fertigstellung mit dem Trunk (Stamm) gemerged (vereint).

Die Schwierigkeit liegt darin, die Software stets zu allen Instanzen des AQopter18 kompatibel zu halten. Dies erfolgt durch Makroanweisungen, die in der basics.h eingestellt werden. Dadurch wird der Code für die aktuell gewünschte Konfiguration selektiert, wobei dies im Wesentlichen Parameter, z.B. Reglerparameter sowie das Zu- und Abschalten von Funktionen betrifft. Die Software ist nicht nur kompatibel mit den fliegenden Quadroptern, sondern auch mit den Versionen des QCS.

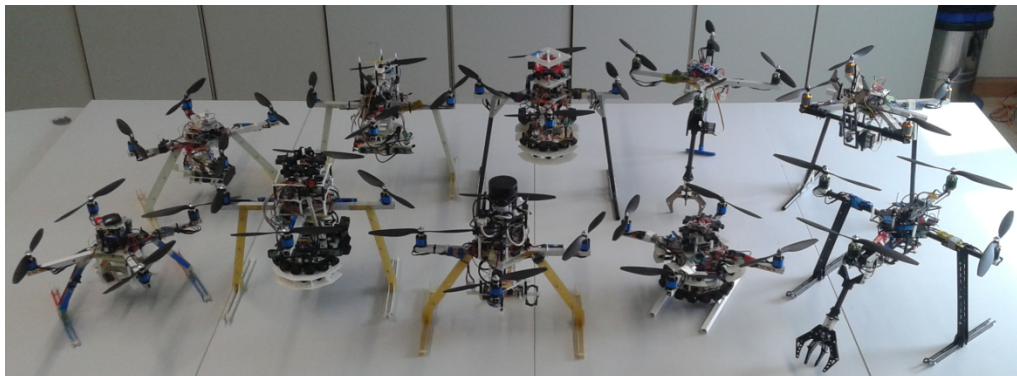


Abbildung 7.3: 10 HW-Instanzen des AQopter18

## 7.3 Rodos-basiertes Softwaredesign

Eine Portierung der Software auf das am Lehrstuhl und beim DLR entwickelte Echtzeitbetriebssystem Rodos existiert ebenfalls, wobei bisher nicht alle Funktionalitäten portiert wurden. Dies liegt nicht zuletzt an dem Aufwand, diese ausgiebig zu testen. Dabei sei angemerkt, dass das klassische Softwaredesign bereits einige Jahre zuverlässig funktioniert und es die gestellten Anforderungen bisher vollends erfüllt, wenngleich es langsam an seine Grenzen stößt.

An dieser Stelle sind zumindest die beiden Portierungs-Stränge für die beiden Mikrocontroller der Hersteller Atmel (AVR UC3A0512) und STM (STMicroelectronics, Discovery STM32F407) zu nennen. Die neuere Portierung für den STM32F407 Mikrocontroller wird im Folgenden kurz beschrieben.

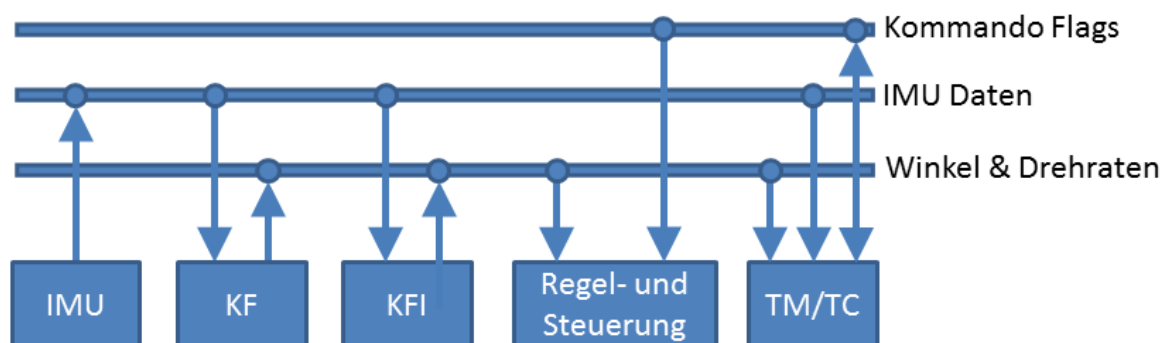


Abbildung 7.4: Topics und Threads des Rodos-basierten Softwaredesigns

Die Portierung dient hauptsächlich der Lehre und besteht aus 5 voneinander unabhängigen Threads.

Diese Threads (vgl. Abbildung 7.4) bestehen aus dem IMU-Thread zum Auslesen der Lage-Sensorik, dem KF-Thread, der die Orientierungsbestimmung mittels Kalman Filter durchführt sowie dem KFI-Thread, einer alternativen Lösung zum Kalman Filter. Die beiden übrigen Threads dienen der Regel- und Steuerung sowie dem Versenden und Empfangen der Telemetrie und Telekommandos.

Die Threads sind unabhängig, da sie über ein festgelegtes Interface, bestehend aus drei Topics, miteinander interagieren, jedoch einander zur Kompilierung nicht benötigen. Diese Topics bestehen aus Kommando Flags, IMU Daten sowie den aktuell gültigen Winkeln und Drehraten.

Der KF-Thread sowie der KFI-Thread lesen die IMU Daten und schreiben die Winkel und Drehraten, die der Regler benötigt. Dabei erkennt der KFI-Thread automatisch, dass der KF-Thread mitkompiliert ist. In diesem Fall publiziert er sein Ergebnis nicht, damit er nicht in Konflikt mit dem KF-Thread gerät und die Regelung womöglich stört.

## 7.4 Operator- und Piloteninterface

Das Operator- und Piloteninterface wurde mittels Qt, einer verbreiteten C++ Klassenbibliothek, implementiert und wird auf einem PC (on-Board LP-180 oder Bodenstation) ausgeführt. Die Software wird I8QuatPlay genannt. Neben der Funktion als Interface enthält die Software auch alle Methoden, die eine hohe Rechenlast benötigen.

**Tabelle 7.1: Softwaremodule von I8Quatplay und deren Autoren / Editoren**

Modulname	Funktion	Autor, Editor
Mainwindow	Hauptfenster, Hauptteil	Gageik, Alle
Navigationcontrol	Navigation	Gageik
Situation Awareness	Situationserfassung und deren Anzeige	Gageik, Müller
Waypoint Generator	Verwaltung von Wegpunkten (Suche)	Gageik
Glwidget	3D Anzeige	Gageik, Kempf
Graphview	Graphenanzeige	Dilger
Opticaltracking	Treiber für optisches Tracking	Dilger
OTS_Control	OTS Steuerung	Scharf
Quaternion	Quaternionenoperationen	Gageik
Telemetry	Telemetry und Telekommandos	Dilger
Stereocamera	Stereo-Optische Abstandsmessung (SV)	Lauterbach
Search	Suchsimulator	Barth
rpLidar	Treiber für Lidar	Gageik
PMD	PMD Kamera Treiber und Auswertung	Dechend, Gageik
objectRec	Objekterkennung	Reul
Mapping	Erstellung einer Karte	Schmitt
Localization	Lokalisierung mit Hilfe einer Karte	Rothkegel
Hmd	HMD Telemetrieanzeige	Caliskan
Gpscontrol	GPS Steuerung	Tüchert
Fouriertracking	Fourier Tracking	Reinthal
Driver	Treiber für UART und TCP	Gageik, Dilger
Cam	Anzeige und Steuerung für OF	Strohmeier
myMath	Mathematische Operationen	Gageik
Histogramm Filter	Histogramm Filter für Lidar	Gageik
Client	Client für TCP Kommunikation	Smolka
CommServer	Server für TCP Kommunikation	Smolka

### 7.4.1 Softwaredesign I8Quatplay

Die Software I8Quatplay setzt sich aus 26 Modulen mit insgesamt ca. 25.000 Zeilen Code zusammen. Eine Übersicht aller Module, ihrer Funktionen und Autoren gibt Tabelle 7.1. Während die Low-Burden Software (AVR, STM) größtenteils vom Autor dieser Arbeit entwickelt wurde, waren viele Studenten an der Entwicklung der Software I8Quatplay beteiligt. Die Beteiligungen erfolgten im Rahmen von Abschlussarbeiten und als studentische Hilfskraft im Rahmen von Projektarbeit.

Das wichtigste Modul stellt das Hauptfenster (mainwindow) dar, welches alle weiteren Objekte generiert und initialisiert. Die verschiedenen Klassen und Module sind per Signals und Slots miteinander verbunden, wobei die Steuerung jeweils über Timer funktioniert. Somit sind die einzelnen Prozesse scheinbar unabhängig voneinander und können simultan durchgeführt werden, wobei natürlich die Rechenleistung des Systems den begrenzenden Faktor darstellt. Dies bedeutet, dass die Prozesse einander beeinflussen, sofern die Rechenleistung nicht ausreichend ist. Entsprechend ist die Wiederholrate der Prozesse einzustellen, wobei dies von den aktuell verwendeten Modulen abhängt. Welche Module kompiliert und verwendet werden sollen, kann per Makroanweisung konfiguriert werden.

Allgemein ist Qt plattformunabhängig, jedoch werden in diesem speziellen Fall ein paar Programmbibliotheken (\*.dll) verwendet, die Windows voraussetzen. Deshalb ist die Software in ihrer aktuellen Fassung auf Windows angewiesen. Ein Betriebssystem wie Windows bietet keine harte Echtzeit. Die damit verbundenen Schwankungen im mehrstelligen Millisekunden-Bereich stellen aber für die hier betrachteten Operationen meist kein allzu großes Problem dar.

### 7.4.2 GUI für Operator und Piloten

Die GUI zur TM/TC-basierten Steuerung des AQOptersI8 gliedert sich in 12 Haupt-Reiter, von denen einige bereits im Kapitel 6.7 vorgestellt wurden. Ausgewählte weitere wesentliche Reiter werden im Folgenden kurz beschrieben.

#### Kommunikationsreiter (Comm)

Der Comm-Reiter (Abbildung 7.5), auch Kommunikationsreiter genannt, ist der initiale Reiter zum Einstellen der Kommunikationsparameter und Aufbauen der Verbindung. Wesentliche Bestandteile sind die Konsole (1), die Kombo-Boxen zur Auswahl des COM-Ports (2) und der Baudrate (3) sowie ein Button (4) zum Aufbauen der seriellen Verbindung. Damit können Telemetriedaten per Bluetooth empfangen und Telekommandos gesendet werden. Mit dem Connect-Button (6) kann eine Verbindung zum OTS aufgebaut werden.

#### Menü für Telemetrie und Telekommandos (Debug, Parameter)

Die Telemetrie kann im Debug-Reiter während des Betriebs dynamisch konfiguriert werden, d.h. es werden nur solche Telemetriedaten verschickt, die aktuell gewünscht sind. Aktuell stehen 78 Telemetrie-Nachrichten mit je bis zu 16 Fließkommazahlen zur Verfügung. Das ergibt ca. 5kB. Diese Daten permanent zu verschicken und zu verarbeiten, würde eine unnötig hohe Last bedeuten. Bei einer Wiederholrate von 100ms würde zudem die verwendete Baudrate von 57.600 Baud/s bei Weitem nicht ausreichen. Daher ist eine Möglichkeit der Selektion der Telemetriedaten erforderlich, die mit der dynamischen Konfiguration gegeben ist.

Im Reiter *Parameter* können Parameter, wie z.B. PID-Werte für die Regler, eingestellt werden. Dies sind Telekommandos, die an den Quadrokopter verschickt werden. Die beiden Reiter *Debug* und

Parameter gehören zum Telemetry-Modul, das im Wesentlichen von Erik Dilger implementiert wurde.

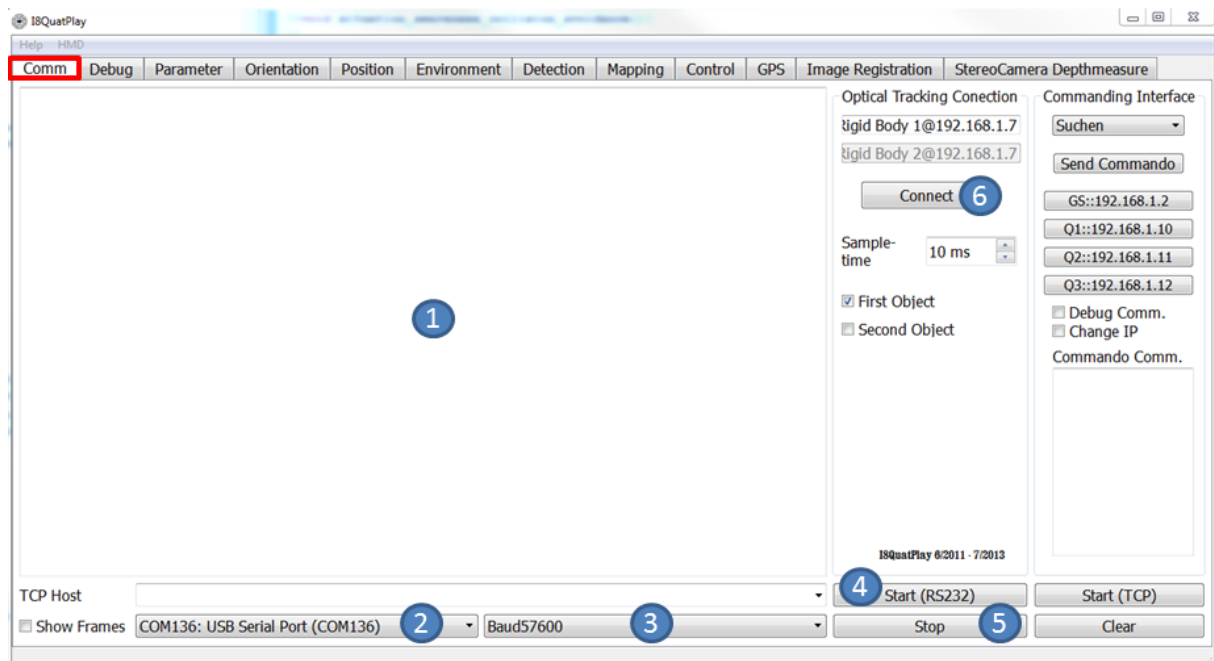


Abbildung 7.5: I8Quatplay Comm-Reiter

### 3D-Anzeige (Orientation)

Die aktuelle Lage des Quadropters in drei Dimensionen, das aktuelle Quaternion, die RPY-Winkel und die aktuelle Rotationsmatrix können im Reiter *Orientation* verfolgt werden. Zudem kann hier der Quadropters gesteuert und die gültige Orientierung manipuliert werden, womit die Drift- und Fehlerkompensation getestet werden kann.

### Anzeige zur Situationserfassung (Environment)

Der Environment-Reiter dient der Anzeige der Ergebnisse der Situationserfassung, wobei dies nur die Resultate unter Verwendung der Ultraschallsensoren, der Infrarotsensoren und dem Lidar betrifft. Dazu können die folgenden Optionen ausgewählt werden: Ultraschall, Infrarot, Ultraschall und Infrarot (fusioniert), Lidar Rohdaten und Lidar-Sektoren. Hauptaufgabe dieses Reiters ist das Überprüfen der Situationserfassung zum Optimieren und Evaluieren der Kollisionsvermeidung. Die Anzeige der Lidar Rohdaten und Sektoren wurde bereits in Abbildung 5.7 gezeigt und die weiteren Anzeigen sind analog. Das Design zur Anzeige geht dabei teilweise auf die Arbeit von Thilo Müller zurück [A01].

### Reiter zur Objektsuche (Detection)

Die Objektsuche wird im Reiter *Detection* (Abbildung 7.6) konfiguriert und gestartet. Dazu werden zunächst der gewünschte Such- (1) und der Sichtbereich (VFOV, 2) sowie der Maskierbereich (3) eingestellt. Mittels Kombo-Box (4) wird die Suchstrategie ausgewählt. Durch Betätigung des Buttons "Send Waypoints" (5) werden, entsprechend der gemachten Einstellungen, die Wegpunkte automatisch generiert und als Liste an den Quadropters übertragen. Die Wegpunkte werden zusätzlich in der Karte (6) als Feedback für den Benutzer angezeigt. In der Karte werden zudem die Position des Quadropters sowie die gefundenen Ziele in Echtzeit aktualisiert. Durch Betätigung des „Start, Count and Stop“ (7) Buttons wird die Objektsuche gestartet. Der AQuopterI8 führt dann die



komplette Suche voll autonom durch und landet am letzten Wegpunkt.

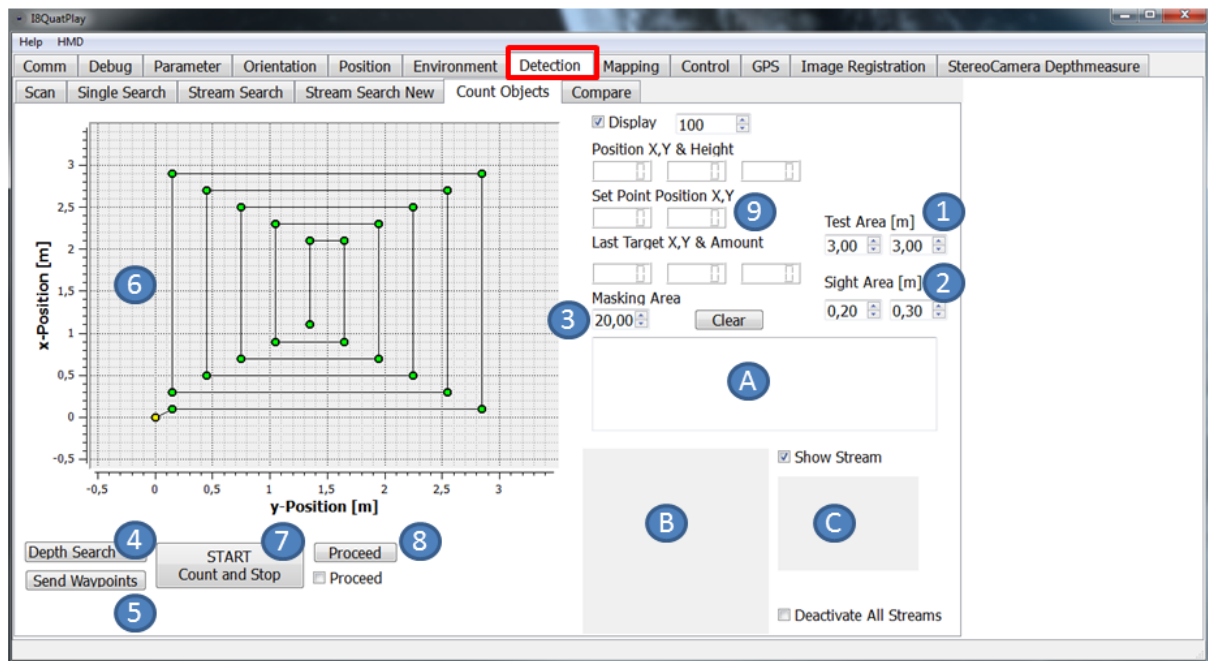


Abbildung 7.6: I8Quatplay Detection-Reiter

Mit der Proceed-Option kann eingestellt werden, dass die Suche beim Erreichen eines Targets gestoppt wird, so dass der Quadrokopter über dem Ziel verweilt. Mit dem Proceed-Button wird die Suche dann fortgesetzt (8). Die aktuelle Position, Höhe und der nächste Zielpunkt sowie die Position des letzten Ziels werden als *LCD-Number* in Echtzeit angezeigt (9). Zudem wird eine Liste der gefundenen Ziele mit ihren Positionen (A) geführt. Der Live-Stream der Kamera zur Objekterkennung (B) und die aktuellen Ergebnisse der Objekterkennung (C) werden ebenfalls dargestellt.

### Reiter zur Wegpunktkommandierung (Control)

Die Wegpunktkommandierung und der gerichtete Flug werden im Reiter *Control* (Abbildung 7.7) gesteuert. In der Karte (1) werden die Wegpunkte dargestellt und es können per Mausclick Wegpunkte generiert werden. Die aktuelle Position wird ebenfalls in Echtzeit dargestellt. Durch Betätigen des Buttons „Set Up“ (2) wird die Wegpunktkommandierung initialisiert und aktiviert. An den Quadrokopter wird das Kommando KA2 gesendet, das dem Wechseln in den Betriebsmodus AF2 (S. 114f) dient. Dieser Schritt ist zum Durchführen der Wegpunktkommandierung erforderlich und dient der Sicherheit.

Nachdem die Wegpunktkommandierung aktiviert ist, können Positionen (3) und Yaw-Winkel (4) als Kommando übertragen werden. Diese Kommandos werden unmittelbar vom Quadrokopter ausgeführt. Durch Betätigung des Buttons „Start All“ (5) wird der gerichtete Flug durchgeführt. Dabei dreht sich der Quadrokopter zunächst in Richtung des nächsten Wegpunktes, bevor er diesen anfliegt. Die Wegpunktliste kann in einer Datei gespeichert, aus einer Datei gelesen, gelöscht (6) sowie manuell erweitert werden (7). Die Wegpunktliste (8) sowie die ID des aktuellen Wegpunktes und die Gesamtanzahl an Wegpunkten werden dargestellt. Zudem kann die Grenzqualität für den ADNS-3080 modifiziert werden (9). In Echtzeit werden für den Benutzer als *LCD-Number* die Position des Quadrokopters in raumfesten und in körperfesten Koordinaten angezeigt. Höhe und Yaw-Winkel

können ebenfalls verfolgt werden (A).

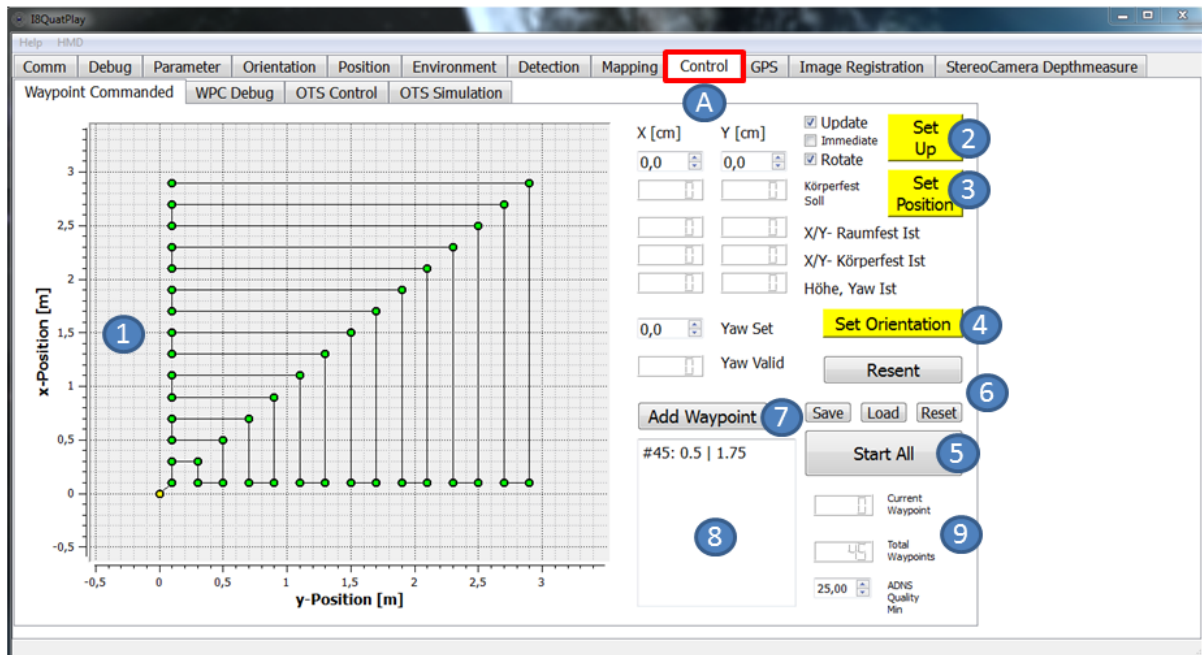


Abbildung 7.7: I8Quatplay Control-Reiter





*The great questions of the day will not be settled by means of speeches  
and majority decisions, but by iron and blood.*  
[Bismarck]

## 8 Evaluierung



## 8.1 Überblick Evaluierung

Das Kapitel zur Evaluierung analysiert die wesentlichen Entwicklungen und Komponenten des Systems durch empirische Versuche. Da aufgrund der Komplexität des Systems und des damit verbundenen Umfangs an Versuchen unmöglich alle Aspekte und Parameter untersucht werden können, werden in diesem Kapitel einige ausgewählte signifikante Experimente anschaulich dargestellt. Ziel der Evaluierung ist es, die Machbarkeit der Ansätze sowie wichtige Eigenschaften und Zusammenhänge aufzuzeigen und detaillierter zu analysieren.

Die wesentlichen Themen der Evaluierung sind die Orientierungsbestimmung, die Höhenbestimmung, die Lageregelung, die Höhenregelung, die Kollisionsvermeidung und Abstandsregelung, der autonome Flug, die Objektsuche sowie der gerichtete Flug. Für eine weitergehende Analyse von Positionsbestimmung und -regelung wird auf sonstige Arbeiten des Gesamtprojekts verwiesen [A04 [A23] [P5].

## 8.2 Orientierungsbestimmung

### 8.2.1 Überblick: Evaluierung Orientierungsbestimmung

Zur Evaluierung der Orientierungsbestimmung wurden vier verschiedene Versuchsreihen unternommen. Zunächst wurde das Verhalten der Orientierungsbestimmung über eine sehr lange Versuchslaufzeit am Boden und ohne Bewegung aufgezeichnet (8.2.2). Anschließend wurde die Orientierungsbestimmung im stationären Flug (8.2.3), d.h. die Position wird gehalten, sowie im dynamischen Flug, d.h. beim Ändern der Lage bzw. der Position (8.2.4), analysiert. Zudem wurde die Orientierung mit Fehlereinspeisung untersucht (8.2.5). Mit Fehlereinspeisung ist gemeint, dass die Orientierung künstlich auf einen falschen Wert gesetzt wurde. Damit wurde das Kompensationsverhalten auf große Orientierungsfehler betrachtet.

### 8.2.2 Stationärer Versuch am Boden

#### Versuchsaufbau und Beschreibung

Ziel der stationären Experimente ist die Analyse von Rausch- und Driftverhalten der Orientierung unter idealen Bedingungen, d.h. in Ruhelage und ohne Einfluss der Motoren. Das System wurde dazu für ca. 18h nicht bewegt. Gleichzeitig wurde mit einer Samplerate von 300ms die aktuell gültige Orientierung RPY aufgezeichnet. Die Orientierungsbestimmung mittels CQF verwendet eine Wiederholrate von 10ms. Es wurden 300ms zum Aufzeichnen gewählt, um die Datenmenge in Grenzen zu halten. Als Sensoren kam das Gyroskop der IMU3000 sowie der LSM303DLHC bestehend aus Accelerometer und Magnetometer der MinIMU-9 v2 zum Einsatz.

Das Experiment wurde zwei Mal wiederholt. Bei jedem Versuch wurden 200.000 Messungen gemacht, womit sich eine theoretische Versuchslaufzeit von 60.000s bzw. 16,67h ergibt. Aufgrund von Jitter (zeitliche Schwankungen) und fehlender Echtzeitfähigkeit des Messsystems (Windows PC mit QT) wurden 68.151s (Versuch 1) bzw. 64.944s (Versuch 2) gemessen. Dies ergibt eine Versuchslaufzeit von ca. 18h.

Um eine Vergleichbarkeit der Ergebnisse zu gewährleisten, wurden dieselben Filterparameter verwendet, die auch im Flug zum Einsatz kommen.

## Auswertung des Versuchs

In Abbildung 8.1, Abbildung 8.2 und Abbildung 8.3 ist der gemessene, zeitliche Verlauf des Roll-, Pitch- und Yaw-Winkels während des Versuchs wiedergegeben. Im Idealfall sollten alle Werte konstant sein.

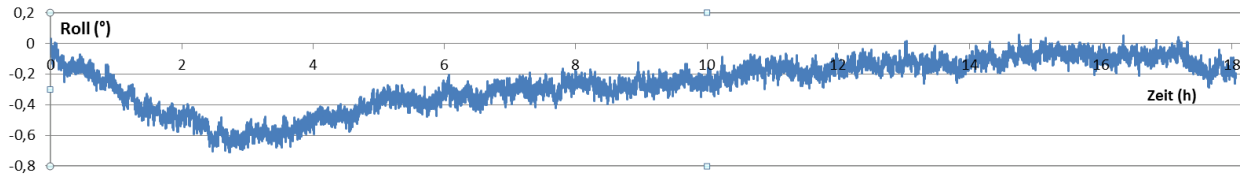


Abbildung 8.1: Zeitlicher Verlauf des Roll-Winkels (stationär am Boden, Versuch 2)

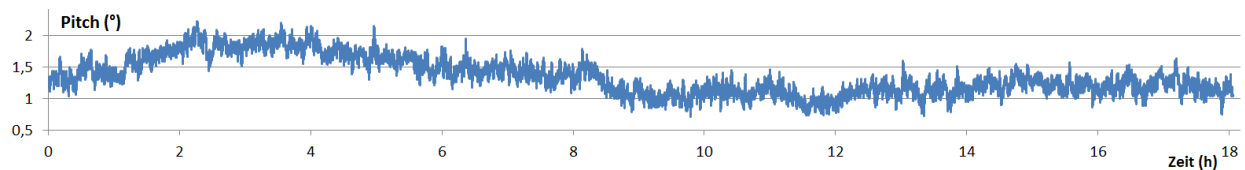


Abbildung 8.2: Zeitlicher Verlauf des Pitch-Winkels (stationär am Boden, Versuch 2)

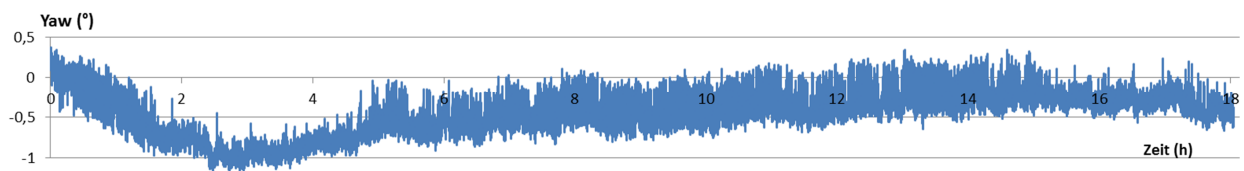


Abbildung 8.3: Zeitlicher Verlauf des Yaw-Winkels (stationär am Boden, Versuch 2)

Tabelle 8.1 fasst die Ergebnisse der beiden 18h-Versuche zusammen. Die Spalte „Max“ enthält den größten gemessenen Wert. Die Spalte „Min“ den niedrigsten gemessenen Wert. Die Spalte „Differenz“ ist die Differenz aus „Max“ und „Min“ und beinhaltet somit die größte Abweichung. Das Maximum der Differenz über alle Achsen beträgt  $1,65^\circ$ . Somit ist der maximale Fehler kleiner als  $1,65^\circ$ . Die Varianz der gemessenen Winkel liegt zwischen  $0,025^\circ$  und  $0,085^\circ$ . Es fällt auf, dass die Ergebnisse des Roll-Winkels wesentlich besser sind als die Ergebnisse der Pitch- und Yaw-Winkel.

Tabelle 8.1: Stationäre Versuche am Boden: Drift- und Rauschverhalten

Stationär Boden	Versuch 1				Versuch 2			
	Max	Min	Differenz	Varianz	Max	Min	Differenz	Varianz
Roll ( $^\circ$ )	0,46	-0,35	0,81	<b>0,025</b>	0,06	-0,71	0,77	0,026
Pitch ( $^\circ$ )	2,22	0,57	<b>1,65</b>	0,060	2,22	0,72	1,50	<b>0,085</b>
Yaw ( $^\circ$ )	1,16	-0,42	1,58	0,053	0,38	-1,18	1,56	0,079
Zeit (s)	71950	3799	68151	-	71299	6305	64994	-

Dem zeitlichen Verlauf der RPY-Winkel (Abbildung 8.1, Abbildung 8.2 und Abbildung 8.3) kann entnommen werden, dass der Winkel nicht konstant bleibt, sondern sich langsam ändert. Es kann davon ausgegangen werden, dass sich die Temperatur in diesem Zeitraum signifikant geändert hat. Der Versuch wurde in einem Büroraum im Sommer 2015 unternommen. Hinzu kommen andere Umwelteinflüsse wie z.B. Druckänderungen. Welcher Faktor letztlich die Abweichung bewirkt, bleibt unklar. Es kann aber festgehalten werden, dass die maximale Änderung über 18h innerhalb eines

Intervalls von  $1,65^\circ$  liegt.

Bei genauerer Betrachtung der Graphen fällt auf, dass die größte Abweichung nach ca. 3h aufgetreten ist und danach wieder abgenommen hat. Daher ist zu erwarten, dass auch bei deutlich längerer Versuchslaufzeit kein signifikantes Driften auftritt. Somit sind die Fehler- und Driftkompensation der Orientierungsbestimmung mittels CQF sowie die Langzeitstabilität dieser Lösung nachgewiesen.

### 8.2.3 Stationärer Versuch im Flug

#### Versuchsaufbau und Beschreibung

Ziel des stationären Versuchs im Flug ist die Analyse von Rausch- und Driftverhalten der Orientierung unter realen Bedingungen, d.h. im Flug. Dies ist deshalb bedeutsam, da am Boden der Gravitationsvektor sehr viel zuverlässiger gemessen und zur Orientierungskompensation herangezogen werden kann. Filterparameter, die am Boden selbst bei laufenden Motoren sehr gute Ergebnisse liefern, mögen im Flug versagen. Weitere Fehlerfaktoren, die durch sehr dynamische Flugmanöver auftreten können, sind dabei zunächst ausgenommen.

Es wurden zwei Experimente unternommen, bei denen der Quadrocopter autonom seine Position und Höhe (80cm) hielt. Beim ersten Experiment wurde ein Akku verwendet, weswegen der Versuch auf ca. 6 Minuten begrenzt wurde. Beim zweiten Experiment wurde die Stromversorgung durch ein Kabel ersetzt, womit eine erheblich längere Versuchslaufzeit von 1h möglich wurde. Mit diesem zweiten Versuch wurde die Korrektheit der Orientierungsbestimmung im Flug über einen längeren Zeitraum analysiert. Zur besseren Lesbarkeit wird im Weiteren auf diese beiden Experimente mit „kurz“ und „lang“ Bezug genommen.

Während der Versuche wurde mit einer Samplerate von 100ms die aktuell gültige Orientierung RPY des Quadrocopters, die on-Board mittels CQF berechnet und auch zur Lageregelung verwendet wird, aufgezeichnet. Als einzige weitere Änderung zum stationären Versuch wurde das Accelerometer dynamisch kalibriert, da es Schwierigkeiten gab, den Sensor akkurat zu kalibrieren und diesen mit dem OTS gleich auszurichten. Als wahre Referenz wurde die Orientierung des Quadrocopters mit dem OTS OptiTrack simultan aufgezeichnet [82].

#### Auswertung des Versuchs

Abbildung 8.4 gibt den zeitlichen Verlauf des Roll-Winkels während der ersten Hälfte des kurzen Versuchs wieder. Die Ergebnisse des Versuchs sind in Tabelle 8.2 und Tabelle 8.3 zusammengefasst. Dabei steht OTS für die Messungen des optischen Trackings und IMU für die aus der IMU mittels CQF on-Board bestimmte Orientierung. Das Delta  $\Delta d$  ist jeweils die Differenz der beiden Messungen und kann als Fehler bzw. Abweichung interpretiert werden.

Die Tabellen enthalten die Messungen zu unterschiedlichen Zeitpunkten sowie in der letzten Spalte die Varianzen der jeweiligen Messreihen. Der Zeitpunkt  $t_1$  ist jeweils der Beginn des Experiments. Die Messung zum Zeitpunkt  $t_1$  kann als initiale Abweichung beider Systeme (Registrierung) angesehen werden und muss daher bei der Fehlerbestimmung abgezogen werden.

Zum Zeitpunkt  $t_2$  ist das System abgehoben, womit für Roll ein sprunghafter Fehler von ca.  $2^\circ$  (kurz) bzw. ca.  $5^\circ$  (lang) verbunden ist. In Abbildung 8.4 ist dieser Sprung klar zu sehen. Dabei und bei allen weiteren Experimenten ist zu bedenken, dass das OTS zwar generell im Millimeterbereich, d.h. sehr



genau, messen kann, aber nicht fehlerfrei ist. Einige fehlerhafte Messungen des OTS, die an Hand eines signifikanten Sprunges erkennbar waren, wurden im Vorfeld der Auswertung entfernt.

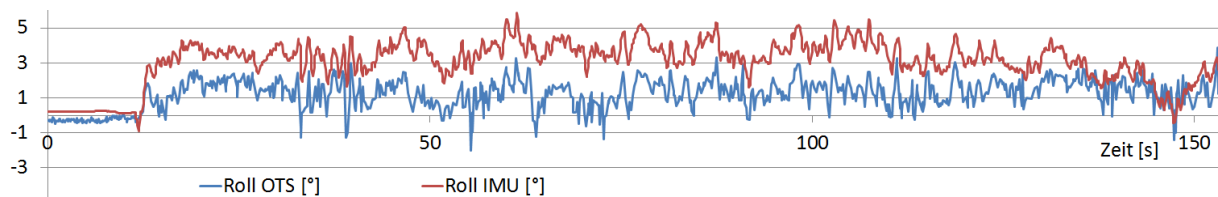


Abbildung 8.4: Zeitlicher Verlauf des Roll-Winkels (stationärer Flug, Dauer ca. 6 Minuten, erhöhte zeitliche Auflösung)

Die Landung wiederum beeinflusst den Fehler für Roll und Yaw. Dieser Fehler wird am Boden nach einiger Zeit kompensiert (kurz:  $t_5$ ). Es lässt sich somit festhalten, dass die Orientierungsbestimmung sensibel auf Höhenänderungen reagiert.

Tabelle 8.2: Stationärer Versuch im Flug mit Akku (kurz): Drift- und Rauschverhalten

Stationär Flug Akku	R = Roll Winkel [°]			P = Pitch Winkel [°]			Y = Yaw Winkel [°]		
	OTS	IMU	$\Delta d$	OTS	IMU	$\Delta d$	OTS	IMU	$\Delta d$
$t_1 = 0s$	-0,29	0,21	0,50	0,31	0,33	0,02	0,86	0,18	-0,68
$t_2 = 18s$	1,49	3,95	2,45	0,49	1,78	1,30	-0,74	0,62	1,36
$t_3 = 150s$	0,92	1,72	0,79	2,00	2,42	0,42	5,45	1,69	-3,76
$t_4 = 374s$	1,55	4,16	2,60	0,55	0,92	0,37	-1,13	1,60	2,73
$t_5 = 389s$	1,49	1,83	0,34	0,57	-0,14	-0,71	-1,63	-0,60	1,03
Varianz	0,63	1,22	0,59	0,42	1,44	1,01	5,81	0,76	-5,05

Tabelle 8.3: Stationärer Versuch im Flug mit Kabel (lang): Drift- und Rauschverhalten

Stationär Flug Akku	R = Roll Winkel [°]			P = Pitch Winkel [°]			Y = Yaw Winkel [°]		
	OTS	IMU	$\Delta d$	OTS	IMU	$\Delta d$	OTS	IMU	$\Delta d$
$t_1 = 0s$	-0,11	-0,18	-0,06	-0,43	0,10	0,53	1,53	-0,08	-1,61
$t_2 = 13s$	8,01	3,24	-4,78	3,25	2,10	-1,15	0,23	0,42	0,19
$t_3 = 1213s$	1,73	4,58	2,85	-0,60	0,69	1,29	2,01	1,55	-0,45
$t_4 = 2413s$	0,36	6,32	5,96	0,97	0,39	-0,58	-0,54	0,18	0,72
$t_5 = 3770s$	1,63	4,64	3,01	0,03	1,20	1,17	0,66	3,63	2,97
Varianz	1,07	1,95	0,88	1,07	1,82	0,75	9,01	1,19	-7,83

Die weiteren Messzeitpunkte aus Tabelle 8.2 und Tabelle 8.3, die während des Fluges (kurz:  $t_3$ ; lang:  $t_3, t_4, t_5$ ) datieren, sowie die Graphen aus dem Anhang (Abbildung 11.1 bis Abbildung 11.6) zeigen, dass der Fehler der Orientierung ständig variiert, jedoch nicht monoton steigt. Nach über 1h beträgt der Fehler ca.  $3^\circ$  für Roll, weniger als  $1^\circ$  für Pitch und ca.  $5^\circ$  für Yaw. Somit ist auch die Langzeitstabilität der Orientierungsbestimmung im Flug nachgewiesen.

Eine Betrachtung der Varianzen zeigt, dass diese mit einer Ausnahme im  $1^\circ$  bzw.  $2^\circ$  Bereich liegen, wobei die Varianzen der IMU von Roll und Pitch ca.  $1^\circ$  höher sind als die Varianzen der OTS Messungen. Daraus kann gefolgert werden, dass die Orientierung aus dem CQF entsprechend

verrauscht ist.

Beim Yaw Winkel zeigt sich ein anderes Phänomen. Die Varianz des OTS ist signifikant größer als bei der IMU. Eine Erklärung können Messfehler des OTS sein bzw. Schwierigkeiten, den Yaw Winkel bei der gegebenen Kamera- und Markerkonstellation korrekt zu bestimmen.

#### 8.2.4 Dynamischer Versuch im Flug

Die Ergebnisse zur Orientierungsbestimmung im Flug bei dynamischen Orientierungsänderungen finden sich bei der Evaluierung zur Lageregelung.

#### 8.2.5 Stationärer Versuch mit Fehlereinspeisung im Flug

##### Versuchsaufbau und Beschreibung

Um die Fähigkeit der Fehlerkompensation genauer zu analysieren und nachzuweisen, wird bei diesem Versuch die Orientierung per Telemetrie-Kommando verfälscht. Wie beim vorangegangenen Versuch befindet sich der Quadrokopter dabei im autonomen stationären Flug, d.h. er regelt eine feste Höhe und Position. Die Konfiguration des Systems bleibt unverändert.

Der Versuch wurde einmal für die Roll-Achse und einmal für die Pitch-Achse durchgeführt. Für die Roll-Achse wurde der Roll-Winkel zu den Zeitpunkten  $t = 50s, 107s, 190s$  und  $220s$  auf  $10^\circ, -10^\circ, -7^\circ$  und  $7^\circ$  gesetzt. D.h. der aktuelle Winkel wird überschrieben und durch den jeweiligen Wert ersetzt. Da der Winkel als Quaternion vorliegt, erfolgt der Eingriff über eine beidseitige Euler-Winkel-Transformation mit Überschreiben der Roll-Komponente. Im zweiten Versuch wird analog für die Pitch-Achse zu den Zeitpunkten  $t = 17s$  und  $224s$  der Pitch-Winkel durch  $-10^\circ$  und  $10^\circ$  ersetzt.

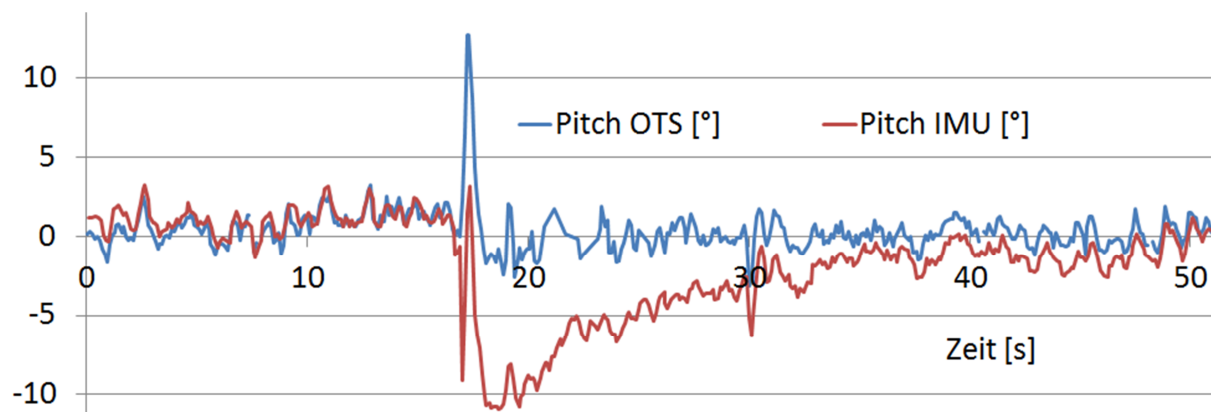


Abbildung 8.5: Zeitlicher Verlauf des Pitch-Winkels mit höherer zeitlicher Auflösung (stationärer Flug, Fehlereinspeisung)

##### Auswertung des Versuchs

Dem zeitlichen Verlauf des Pitch-Winkels (Abbildung 8.5, Abbildung 8.6) sowie der Positionsänderung in x-Richtung (Abbildung 8.7), die mit dem OTS gemessen wurde, kann die Reaktion des Quadrokopters auf die Manipulation des Winkels klar entnommen werden. Der Quadrokopter reagiert unmittelbar auf die Manipulation, indem der Pitch-Regler den vermeintlich falschen Winkel korrigiert. Dies geschieht so schnell, dass es der Grafik schwer entnommen werden kann, da die zeitliche Abtastrate zu niedrig ist. Es zeigt sich aber deutlich im Peak beim OTS und den beiden schnell aufeinander folgenden Peaks der IMU (Abbildung 8.5). Nachdem der vermeintliche Fehler korrigiert ist, steht der Quadrokopter laut IMU gerade. Tatsächlich steht der Quadrokopter nun ca.  $10^\circ$  schief und fliegt in negative x-Richtung davon. Der Positionsregler fängt ihn schließlich ab, wobei

sich ein Positionsfehler von ca. 1,5m aufgebaut hat. Nach ca. 10s bis 20s sind der Orientierungsfehler und der daraus resultierende Positionsfehler kompensiert. Die Graphen zur Roll-Achse befinden sich im Anhang und weisen ein analoges Verhalten nach (Abbildung 11.7, Abbildung 11.8). Somit ist die Fähigkeit der Orientierungsfehlerkompensation des CQF im Flug nachgewiesen.

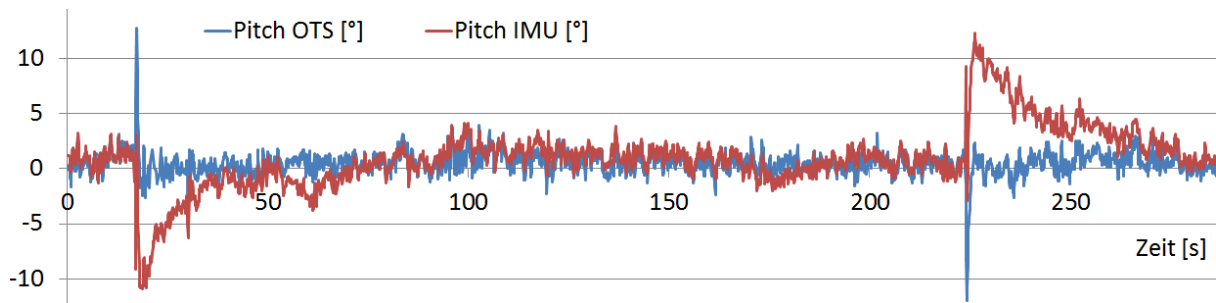


Abbildung 8.6: Zeitlicher Verlauf des Pitch-Winkels (stationärer Flug, Fehlereinspeisung)

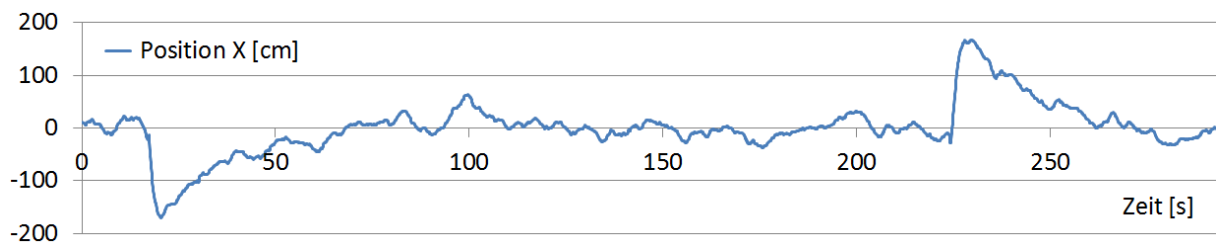


Abbildung 8.7: Zeitlicher Verlauf der x-Position mit OTS gemessen (stationärer Flug, Fehlereinspeisung)

## 8.3 Höhenbestimmung

### Versuchsaufbau und Beschreibung

Zur Evaluierung der Höhenbestimmung wurde der Quadrocopter manuell auf verschiedenen Höhen geflogen. Dabei wurde die Höhe mit den bordeigenen Sensoren sowie dem OTS gemessen. Das OTS versagte ab einer Höhe von ca. 2,5m, womit in dieser Höhe keine Referenz zur genauen Bestimmung des Fehlers der Höhenbestimmung existiert.

Ziel der Versuche zur Höhenbestimmung ist weniger die Genauigkeitsanalyse, als vielmehr der Machbarkeitsnachweis und die Plausibilität der dargestellten Lösung. Die Herausforderung der Höhenbestimmung liegt in der Selektion des korrekten Low-Cost Sensors, die teilweise signifikante systematische Fehler aufweisen. Dies bedeutet, dass in manchen Situationen zwei von drei Sensoren versagen. Auch in diesen Fällen soll die Datenfusion Namens WF das bestmögliche Ergebnis liefern.

Der Versuch wurde 5-mal durchgeführt und es zeigten sich bei allen Wiederholungen vergleichbare Ergebnisse.

### Auswertung des Versuchs

Die Auswertung der Versuche zur Höhenbestimmung gliedert sich in drei Aspekte. Zunächst wird das Ergebnis der Höhenbestimmung insgesamt auf Plausibilität und qualitative Korrektheit überprüft.

Anschließend wird auf die zwei Schritte des WF-Designs genauer eingegangen.

In Abbildung 8.8 ist die mittels Höhenbestimmung (Total), OTS und Luftdruck (AP) bestimmte Höhe aufgezeichnet. Dem roten Graphen ist auch das Versagen des OTS ab einer Höhe von ca. 2,5m zu entnehmen.

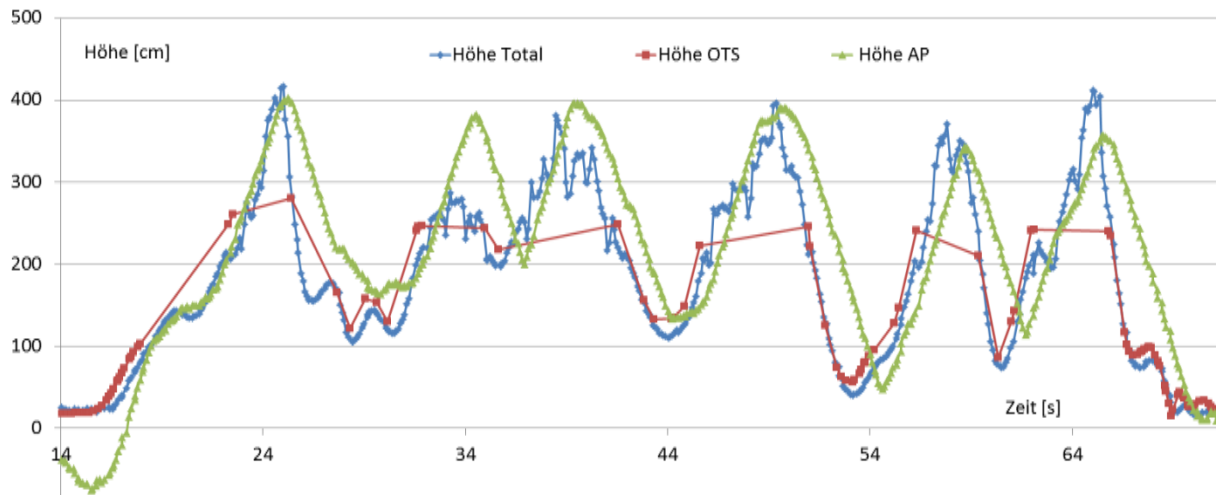


Abbildung 8.8: Zeitlicher Verlauf der Höhe mit Höhenbestimmung, OTS und Luftdruck (AP) gemessen

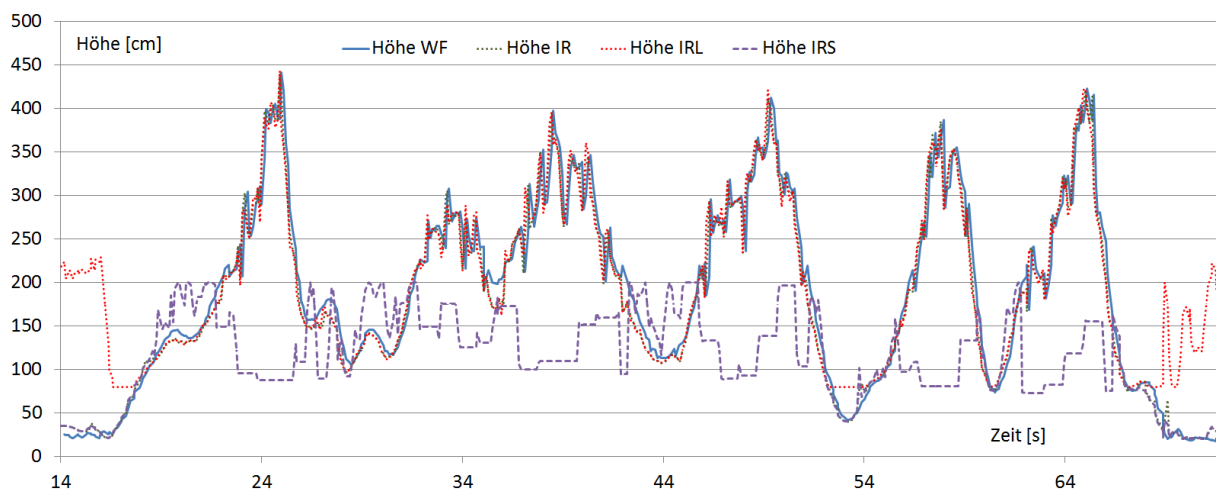


Abbildung 8.9: Zeitlicher Verlauf der Höhe (Vergleich IR-Fusion)

Durch Vergleich von OTS (rot) und Höhenbestimmung (blau) in niedrigeren Höhen und Vergleich von Höhenbestimmung und Luftdrucksensor (grün) in höheren Höhen ist die qualitative Korrektheit der Höhenbestimmung nachgewiesen. Einzige wesentliche Abweichung tritt in Sekunde 34 mit ca. 1m bei einer Höhe von ca. 2,5m auf. Diese Abweichung kann aber auch teilweise von einem Fehler des Luftdrucksensors herrühren. Es kann insgesamt gesagt werden, dass ab einer Höhe von ca. 2m die Genauigkeit der Höhenbestimmung deutlich abnimmt.

Zur genaueren Analyse der Höhenbestimmung werden die beiden Schritte der Fusion, IR-Fusion und Gesamtfusion, genauer beleuchtet. Mit IR-Fusion ist die Fusion der beiden IR-Sensoren gemeint, wobei US, AP und IMU als Referenz dienen. Die Schwierigkeit und Korrektheit der IR-Fusion wird in Abbildung 8.9 deutlich. Den Graphen kann entnommen werden, dass der kurze IR-Sensor (IRS) bei

größeren Höhen und der lange IR-Sensor (IRL) bei niedrigeren Höhen versagt. Das Ergebnis der IR-Fusion (Höhe IR) deckt sich dabei mit dem Ergebnis des WF (Gesamtfusion), womit zusammen mit den vorangegangenen Ergebnissen und den Vergleichen mit OTS und AP die Korrektheit der IR-Fusion nachgewiesen ist.

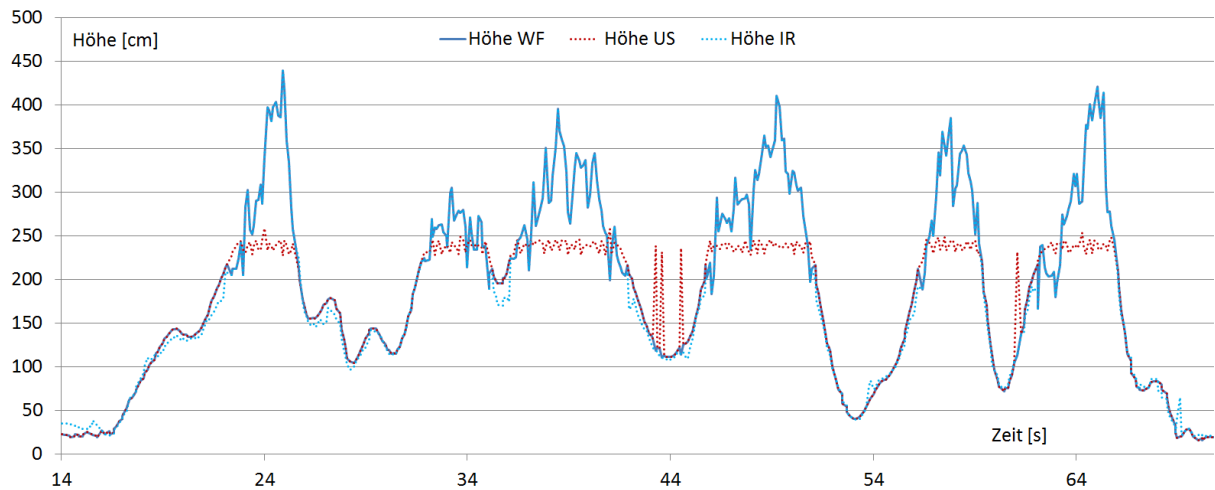


Abbildung 8.10: Zeitlicher Verlauf der Höhe (Vergleich IR-Fusion)

Das Verhalten der Gesamtfusion des WF, wozu US-Sensor und Ergebnis der IR-Fusion unter Verwendung der Referenzsensoren miteinander fusioniert werden, wird durch Abbildung 8.10 veranschaulicht. Aus der Abbildung geht hervor, dass der US-Sensor ab einer Höhe von ca. 2,4m falsche Werte liefert. Somit steht ab dieser Höhe nur noch die Messung des langen IR-Sensors als einzig korrekte Höhe (Abstand zum Boden) zur Verfügung, d.h. zwei von drei Sensoren liefern eine falsche Messung. Aus den Graphen wird ersichtlich, dass der WF in diesen Fällen korrekterweise den Messungen von IRL folgt. Damit ist die Korrektheit des WF zur Höhenbestimmung nachgewiesen.

### Weitere Versuche

Im Rahmen der Bachelorarbeit von Julian Rothe wurde die Höhenbestimmung darüber hinaus gemeinsam ausgiebig untersucht. Die Ergebnisse dieser Versuche finden sich in der Bachelorarbeit [A02] sowie in einer Veröffentlichung auf der UAVvek 2012 [P3].

## 8.4 Lageregelung

### Überblick: Evaluierung Lageregelung

Zur Evaluierung der Lageregelung gehört das Regelverhalten von Pitch-Regler, Roll-Regler und Yaw-Regler. Die Versuche dienen gleichzeitig der Evaluierung der Orientierungsbestimmung bei dynamischen Orientierungsänderungen im Flug.

### Versuchsaufbau und Beschreibung

Zur Evaluierung der Lageregelung wurde der Quadrocopter manuell mit aktivierter Höhenregelung geflogen und die Lage wurde per Fernsteuerung verändert. Die Manipulationen einer Achse bewirkten einen dynamischen Flug des Quadrocopters. In drei verschiedenen Versuchen wurde der Quadrocopter über die Roll-Achse, Pitch-Achse und Yaw-Achse gedreht. Als wahre Referenz zur Orientierung des CQF (IMU) dient die Messung des OTS. Der aktuell gültige Soll-Wert, der von der

Fernsteuerung vorgegeben wird, wurde ebenfalls aufgezeichnet.

## Auswertung des Versuchs

Die Graphen aus Abbildung 8.11 und Abbildung 8.12 beinhalten die Ergebnisse des Versuchs zur Roll-Achse. Aus den Graphen wird ersichtlich, dass der Soll-Wert schnell und stabil eingestellt wird. Eine sprunghafte Sollwertänderung von ca. 20° ist nach ca. 200ms ohne erkennbares Überschwingen erreicht.

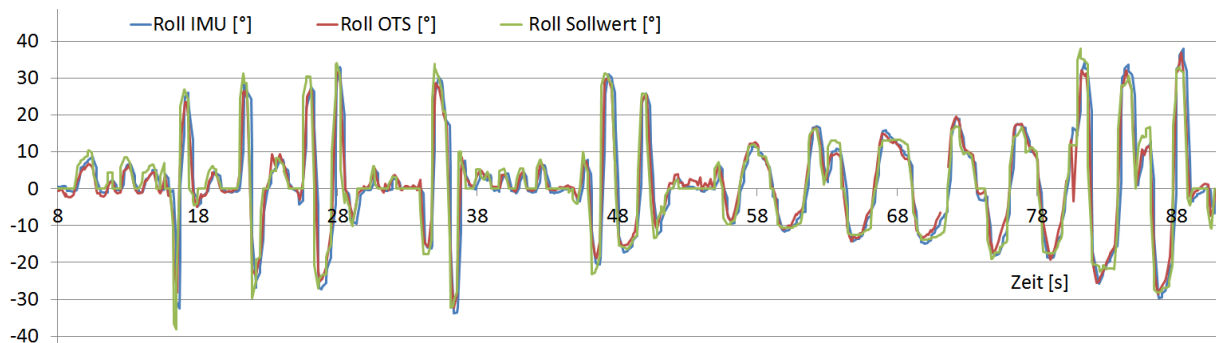


Abbildung 8.11: Zeitlicher Verlauf des Roll-Winkels (dynamischer Flug)

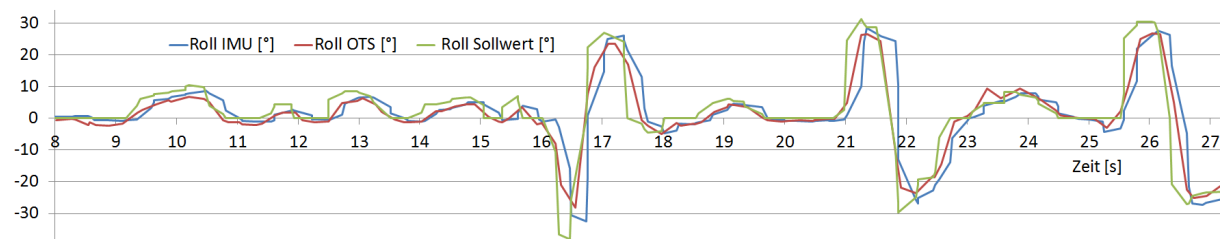


Abbildung 8.12: Verlauf des Roll-Winkels mit größerer zeitlicher Auflösung

Die Versuche zu den beiden anderen Achsen (Pitch, Yaw) zeigen vergleichbare Ergebnisse (Abbildung 11.9, Abbildung 11.10). Damit ist eine stabile und schnelle Lageregelung nachgewiesen.

Bei der Betrachtung des Versuchs zur Yaw-Regelung (Abbildung 11.10) fällt auf, dass bei Drehungen jenseits der 90° die Messungen von OTS und IMU abweichen. Hier liegt ganz offensichtlich ein Fehler des OTS vor, da die Messungen der IMU plausibel erscheinen. Zudem zeigen sich bei dem Versuch größere Abweichungen zwischen OTS und IMU. Ursachen dafür sind, wie bei vorangegangenen Abweichungen bereits vermutet, ein Fehler des mittels OTS bestimmten Yaw-Winkels, eine schlechte Kalibrierung des Magnetsensors sowie magnetische Störungen und andere Fehler bei der Orientierungsbestimmung für Yaw.

Aus den Versuchen zum dynamischen Flug wird des Weiteren ersichtlich, dass die Orientierungsbestimmung bei Dynamik insgesamt korrekt arbeitet. Mit Ausnahme der bereits diskutierten Yaw-Achse sind keine signifikanten Abweichungen zu erkennen. Der um die initiale Abweichung bereinigte Fehler zwischen IMU und OTS beträgt nach 90s dynamischen Flugs 0,75° für Roll (Versuch Roll-Achse). Im Fall des Versuchs zur Pitch-Achse beträgt der Fehler nach 60s 0,38°. Dabei ist zu beachten, dass der Fehler größer erscheinen kann, als er wirklich ist, wenn die Messungen nicht synchron erfolgen. Eine hochgenaue Synchronisation der Messungen ist in diesem Fall aber nicht möglich. Dies ist dem Messaufbau geschuldet. Es dient ein PC als Bodenstation sowie Bluetooth als Medium für die IMU-Messungen und Ethernet für die OTS-Messungen. Dabei ist zu

bedenken, dass Winkeländerungen von  $10^\circ$  pro 100ms auftreten, die Frequenz der Messungen aber lediglich 10Hz beträgt. Insbesondere bei der Betrachtung der Roll-Achse lässt sich erkennen, dass das System auch nach 90s noch in Bewegung war. Dies könnte den ungefähr doppelt so großen Fehler bei Roll im Vergleich zu Pitch erklären.

## 8.5 Höhenregelung

### Versuchsaufbau und Beschreibung

Zur Evaluierung der Höhenregelung wurde der Quadrokopter manuell mit aktivierter Höhenregelung geflogen. Dabei wird die Höhe mittels Weighted Filter bestimmt und zur Regelung der Höhe verwendet. Als Höhensensoren kamen die beiden IR-Sensoren, der US-Sensor, der Luftdrucksensor sowie die IMU zum Einsatz. Simultan wurde die Höhe mit dem OTS gemessen. Diese Messung diente lediglich der Evaluierung der Höhenbestimmung und Höhenregelung. Zu Beginn des Versuchs startet der Quadrokopter auf eine Starthöhe von 80cm und landet am Ende des Experiments automatisch. Während des Versuchs wurde die Soll-Höhe per Telekommando mehrfach variiert.

Der Versuch wurde mit zwei verschiedenen maximalen Sollhöhen durchgeführt. Dabei wurde bei Versuch 1 eine maximale Soll-Höhe von 180cm und bei Versuch 2 eine Soll-Höhe von 200cm verwendet.

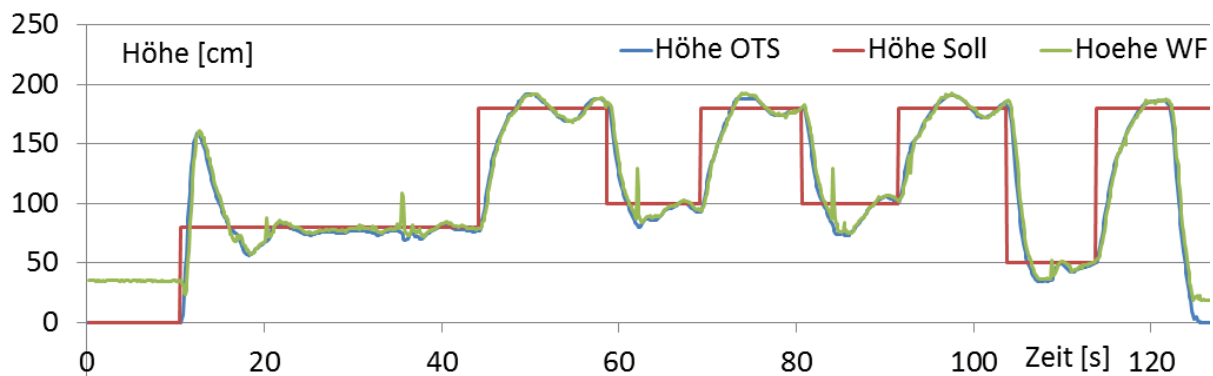


Abbildung 8.13: Zeitlicher Verlauf der Höhe bei dynamischer Höhenregelung (Soll-Höhe bis 180cm)

### Auswertung des Versuchs

Die Versuche zeigen, dass die Soll-Höhe schnell und stabil erreicht wird (Abbildung 8.13, Abbildung 8.14). Eine Höhenänderung von ca. 120cm wird nach ca. 4s erreicht. Das Überschwingen beträgt bei Anstieg ca. 10cm und bei Absinken ca. 20cm. Die autonome Landung funktioniert zuverlässig. Damit ist eine stabile Höhenregelung nachgewiesen.

Das Überschwingen der Höhe beim Start (Abbildung 8.11, Sekunde 10) ist durch einen falschen Parameter für den Schwebeschub zu erklären. Das Verhalten zeigt, dass der Regler in der Lage ist, sich unterschiedlichen Gewichten dynamisch anzupassen. Der Parameter wurde im zweiten Versuch korrigiert (Abbildung 8.12). Ab einer Höhe von ca. 2m versagt das OTS. Diese Aussetzer des OTS sind beim zweiten Versuch zu sehen.

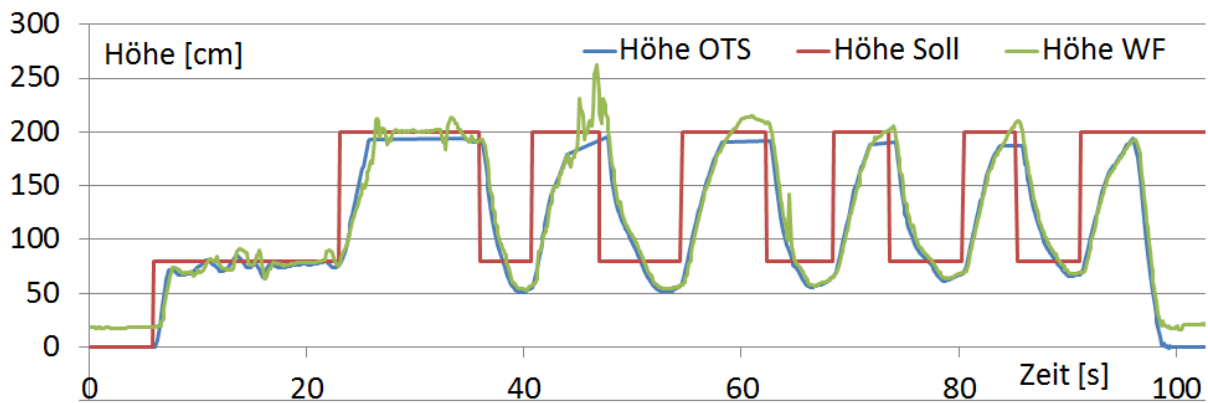


Abbildung 8.14: Zeitlicher Verlauf der Höhe bei dynamischer Höhenregelung (Soll-Höhe bis 200cm)

### Weitere Versuche

Weitere Versuche des Systems zur Höhenregelung wurden auch im Rahmen anderer Arbeiten veröffentlicht [P3][A02].

## 8.6 Kollisionsvermeidung und Abstandsregelung

### 8.6.1 Überblick

Zur Evaluierung der Kollisionsvermeidung und Abstandsregelung wurden im Fluglabor des Lehrstuhls mehr als 100 Experimente durchgeführt. In den anfänglichen Versuchsläufen wurden die Parameter zunächst empirisch optimiert. Anschließend wurden die folgenden drei Versuchsreihen unternommen:

- Stationärer Versuch in einer Ecke
- Stationärer Versuch zwischen zwei Hindernissen
- Dynamischer Versuch mit Wänden und Person

Diese drei Versuchsreihen dienen der Evaluierung der Kollisionsvermeidung. Jede Versuchsreihe besteht aus 10 Experimenten und dauert ca. 3 Minuten. Um vergleichbare Daten zu erhalten, wird aus den gewonnenen Ergebnissen eine repräsentative Zeitspanne von je 60s ausgewählt.

Es wurden diese kurzen Zeitspannen gewählt, um möglichst viele vergleichbare Daten zu erhalten, die nach Möglichkeit unabhängig von weiteren Faktoren wie der Batteriespannung sind. Während aller Experimente agiert der Quadrocopter autonom ohne jegliche Steuersignale. Die Bodenstation diente lediglich dem Empfang der Telemetriedaten und die Fernsteuerung der Sicherheit im Fehlerfall.

### 8.6.2 Stationärer Versuch in einer Ecke

#### Versuchsaufbau und Beschreibung

Beim stationären Versuch in einer Ecke wird das Verhalten der Kollisionsvermeidung zunächst unter den einfachsten Bedingungen betrachtet. Wesentlicher Aspekt der Kollisionsvermeidung ist die Abstandsregelung, die mit diesem Versuch genauer analysiert wird. Um die Abstandsregelung zu aktivieren, wird der Quadrocopter im Fluglabor manuell in eine Ecke geflogen. Dabei befindet sich



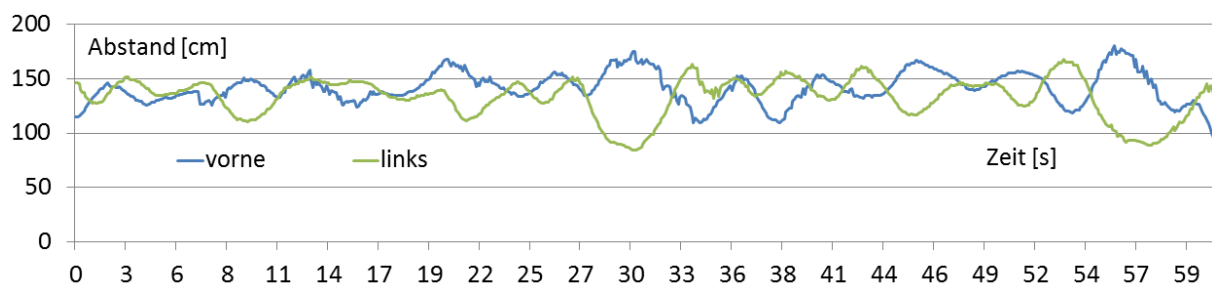
vor dem Quadrokopter sowie an seiner linken Seite eine Wand aus Matten. Die Kollisionsvermeidung wird automatisch beim Unterschreiten des Mindestabstandes aktiviert und hält den Quadrokopter in der Ecke. Fortan operiert der Quadrokopter autonom.

### Auswertung des Versuchs

Die Ergebnisse der stationären Flüge in einer Ecke sind in Tabelle 8.4 zusammengefasst. Der mittlere Abstand zu den Wänden beträgt 144cm (vorne) und 135cm (links). Der Soll-Wert für die Abstandsregelung beträgt 140cm, womit ein Abstand ab Propellerspitze von ca. 1m geregelt wird.

**Tabelle 8.4: Stationärer Flug in einer Ecke**

Ecke Versuch	Mittelwert des Abstandes [cm]		Standardabweichung [cm]	
	vorne	links	vorne	links
1	141,8	133,0	14,6	18,2
2	150,1	138,9	17,2	19,3
3	144,9	140,9	16,6	15,4
4	141,8	137,2	22,5	22,1
5	143,7	134,6	11,4	14,2
6	140,1	135,3	19,7	19,5
7	139,0	135,4	16,3	19,2
8	145,5	139,5	18,7	14,6
9	137,7	128,7	15,3	22,1
10	150,0	131,7	22,3	17,7
Mittelwert	144,1	135,5	17,5	18,2



**Abbildung 8.15: Zeitlicher Verlauf der Abstände beim Flug in einer Ecke**

Die Standardabweichung des Abstandes liegt zwischen 11cm und 23cm. Der Mittelwert der Standardabweichung beträgt ca. 18cm. Abbildung 8.15 stellt den zeitlichen Verlauf der Abstände, die mit Hilfe der Hinderniserkennung bestimmt und zur Abstandsregelung verwendet werden, dar. Die Abstände oszillieren leicht um den Mittelwert und haben einem maximalen Ausschlag (Spitze-Tal-Wert) von ca. 1m. Der Winkel als Stellgröße für die Abstandsregelung verläuft in einem Bereich von +/- 5° (Abbildung 8.16). Diese Daten unterstreichen die niedrige Dämpfung des Systems und das ruhige Regelverhalten. Der niedrigste gemessene Abstand beträgt ca. 85cm und die maximale Amplitude von ca. 1m wird in den meisten Fällen deutlich unterschritten.

Grundsätzlich kann der Regler auch aggressiver eingestellt werden. Damit gehen ein stärkeres Oszillieren aufgrund der verrauschten Sensordaten und somit auch ein höheres Überschwingen einher.

Die geringe Dämpfung liegt an der geringen Reibung (Luft) eines fliegenden Systems. Dies zusammen mit Störungen wie Wind erschwert das Problem der Abstandsregelung. Dabei entsteht Wind bzw. ein Sog durch die Propeller des Quadropters selbst, dessen Luftströmungen von den Wänden beeinflusst werden.

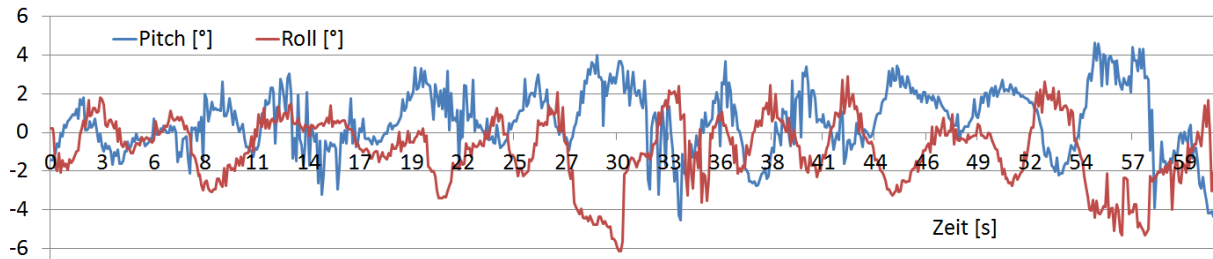


Abbildung 8.16: Zeitlicher Verlauf der IMU-Winkel Roll und Pitch beim Flug in einer Ecke

Durch diese Versuchsreihe konnte somit nachgewiesen werden, dass das System in der Lage ist, autonom in einer Ecke zu fliegen. Dabei kann es Dank der Abstandsregelung seine Position bzw. seinen Abstand zu Wänden halten.

### 8.6.3 Stationärer Versuch zwischen zwei Hindernissen

#### Versuchsaufbau und Beschreibung

Beim stationären Versuch zwischen zwei Hindernissen wurde der Quadroptor an das Ende eines Sackgassen-Tunnels geflogen. Dies bedeutet, dass sich vor dem Quadroptor sowie auf seinen beiden Seiten eine Wand aus Matten befunden hat. Ziel des Experiments ist die Analyse der Abstandsregelung bei einem Flug zwischen zwei Hindernissen. Die Breite des Tunnels betrug 1,8m. Bei einer Spannweite von ca. 60cm blieben dem Quadroptor damit ca. 60cm Platz zu den beiden Seiten.

#### Auswertung des Versuchs

Abbildung 8.17 gibt die gemessenen Abstände zu den Wänden links und rechts vom Quadroptor während eines der Versuche wieder. In Abbildung 8.18 ist der dazugehörige Sollwert für den Roll-Winkel, der Stellwert der Abstandsregelung, zu sehen. Aus den Abbildungen wird das Oszillieren des Systems zwischen den beiden Wänden ersichtlich. Dabei hält das System zu beiden Wänden Abstand.

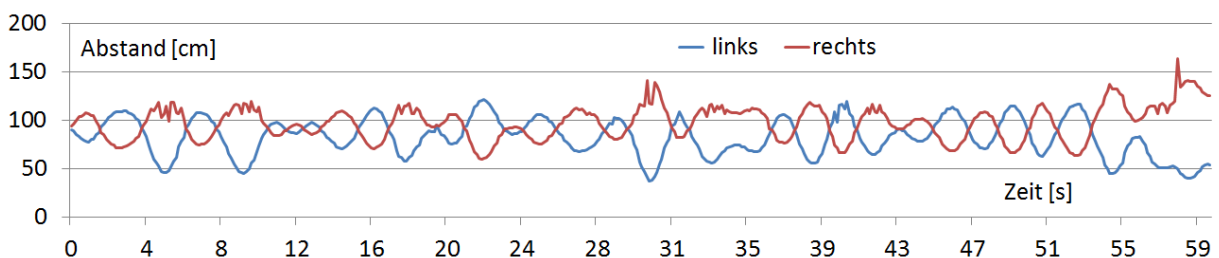


Abbildung 8.17: Zeitlicher Verlauf der Abstände beim Flug zwischen zwei Hindernissen

Der Stellwert liegt weitestgehend im Bereich  $\pm 5^\circ$ . In einigen Fällen unterschreitet der Quadroptor auf der linken Seite einen Abstand von 50cm leicht, wodurch ein Booster aktiviert wird. Dieser Booster erhöht den Stellwert rapide und entfernt den Quadroptor vom Hindernis.

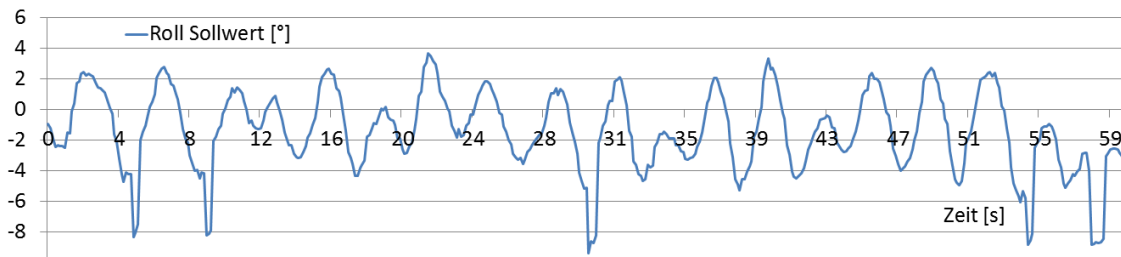


Abbildung 8.18: Zeitlicher Verlauf des Sollwertes für Roll beim Flug zwischen zwei Hindernissen

Alle Booster traten auf der linken Seite auf, obwohl das Verhalten des Quadropters prinzipiell auf beiden Seiten identisch sein sollte. Erklärungen hierzu könnten Fehler in der Orientierung sowie Asymmetrien in der Hardware (Motoren, Propeller, Schwerpunkt) sein.

Tabelle 8.5: Stationärer Flug zwischen zwei Hindernissen

Zwischen Versuch	Mittelwert des Abstandes [cm]		Standardabweichung [cm]	
	rechts	links	rechts	links
1	97,4	82,5	17,4	19,5
2	100,5	78,7	15,0	16,8
3	96,3	81,7	17,1	18,4
4	93,7	84,7	15,5	17,1
5	95,7	82,9	13,5	15,2
6	96,0	83,8	13,4	15,4
7	91,1	88,2	17,3	18,6
8	97,8	81,9	15,2	17,1
9	94,9	84,8	10,9	12,2
10	95,1	84,8	13,0	13,7
Mittelwert	95,9	83,4	14,8	16,4

Tabelle 8.5 fasst die Ergebnisse der 10 Versuche zusammen. Im Vergleich zur Abstandsregelung in einer Ecke (Tabelle 8.4) sind Mittelwert und Varianz bei diesem Versuch niedriger. Der Mittelwert des Abstandes ist niedriger, weil der Quadroptor einen Abstand näher am Hindernis regelt. Dies erfolgt automatisch und ist erforderlich, da sich nun auf beiden Seiten ein Hindernis befindet.

Die niedrigere Standardabweichung erscheint auf den ersten Blick widersprüchlich, da im Tunnel stärkere Turbulenzen herrschen als beim Flug in einer Ecke. Die Erklärung liegt an der Adaption der Reglerparameter, die beim Flug zwischen zwei Hindernissen erfolgt. Da sich in diesem Fall auf beiden Seiten ein Hindernis befindet, darf der Regler nicht aggressiv reagieren, da sich das System andernfalls aufschwingt oder ein Ausweichmanöver auf der einen Seite eine Kollision auf der anderen verursacht. Zudem darf der Regler ohne Einbußen sanfter sein, da er nicht mit hoher Geschwindigkeit auf ein Hindernis zusteuern kann, wenn sich gleichzeitig ein stationäres Hindernis auf der anderen Seite befindet.

Dieser Sonderfall wurde bewusst so gewählt, um in beiden Situationen (Hindernis auf einer bzw. zwei Seiten) eine stabile Abstandsregelung zu ermöglichen. Die dynamische Adaption der Reglerparameter ist durch die Daten der Evaluierung klar nachvollzogen.

## 8.6.4 Dynamischer Versuch mit Wänden und Person

### Versuchsaufbau und Beschreibung

Bei diesem Versuch befindet sich der Quadrokoopter mit einer Person in einem quadratischen Raum. Der Ablauf des Versuchs ist in Abbildung 8.19 veranschaulicht, wobei der Smiley die Person symbolisiert. Da der Quadrokoopter seinen Abstand zu beliebigen Hindernissen regelt, kann er durch bewegliche Hindernisse getrieben werden. Dies wird im folgenden Versuch demonstriert.

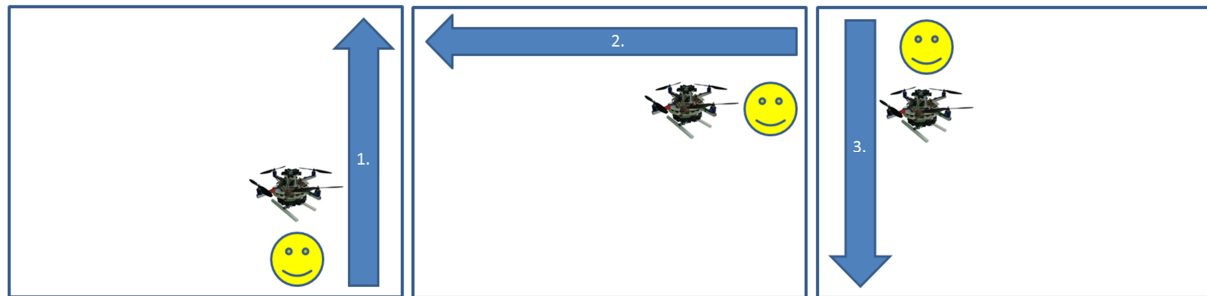


Abbildung 8.19: Ablaufschema des Versuchs mit Wänden und Person (Smiley) bestehend aus 3 Phasen

Zunächst befindet sich die Person hinter dem Quadrokoopter in der rechten unteren Ecke des Raumes. Aus Sicht der Person befindet sich der Quadrokoopter vor ihr. Die Person bewegt sich vorwärts und treibt den Quadrokoopter vor sich her und damit gegen die vordere Wand (1.). Dann stellt sich die Person rechts neben den Quadrokoopter und treibt den Quadrokoopter gegen die linke Wand (2.). Schließlich stellt sich die Person vor den Quadrokoopter und treibt den Quadrokoopter gegen die hintere Wand (3.). Dabei läuft die Person mit einer Geschwindigkeit von ca. 1m/s.

### Auswertung des Versuchs

In Abbildung 8.20 sind die Ergebnisse des 5. Durchlaufs des Versuchs dargestellt. Den Graphen ist zu entnehmen, dass zunächst der Abstand nach vorne passend zum ersten Teil des Versuchs abnimmt, während sich der Abstand rechts (Wand) und hinten (Person) kaum verändert (Sekunde 0 bis 6). Anschließend nimmt der Abstand links ab, da nun der Quadrokoopter gegen die linke Wand getrieben wird (Sekunde 7 - 12). Der Abstand vorne ist in dieser Zeit beinahe konstant. Dies bedeutet, der Quadrokoopter hält seinen Abstand nach vorne, während er von der Person von links nach rechts getrieben wird. Im dritten Schritt nimmt der Abstand hinten ab, während sich die Abstände nach vorne und rechts nahezu konstant verhalten (Sekunde 18 - 23). Nun wird der Quadrokoopter nach hinten getrieben, während sich links eine Wand und vorne die Person befindet.

Neben dem Nachweis eines erfolgreichen Versuchs kann den Graphen darüber hinaus deutlich entnommen werden, dass die Abstandsmessungen bis ca. 2m zuverlässig und darüber hinaus mit zunehmender Entfernung stärker verrauscht sind. Dies veranschaulicht das inhärente Problem der preisgünstigen Sensorik. Dennoch konnte mit diesem Experiment die Fähigkeit des AQopter18, im autonomen Flug Personen im Raum ausweichen zu können, nachgewiesen werden. Dies bedeutet, dass das System in der Lage ist, multiplen Hindernissen aus multiplen Richtungen simultan auszuweichen. Dabei verwendet das System in diesem Fall nur preisgünstige US+IR Sensoren.

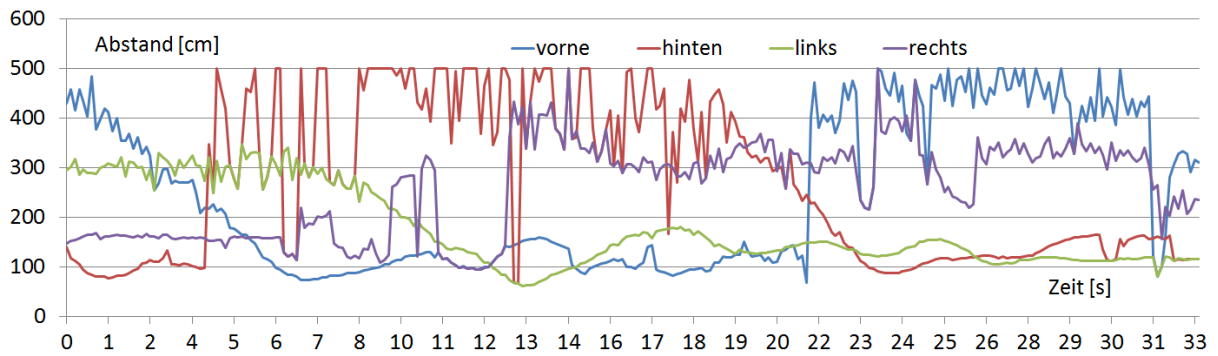


Abbildung 8.20: Vier Abstände beim Versuch mit Wänden und einer Person

### 8.6.5 Weitere Versuche

Die Evaluierung der Kollisionsvermeidung unter Verwendung von IR+US Sensorik wurde auch in der Onlinezeitschrift IEEE Open Access veröffentlicht [P10]. Weitere Resultate zur Kollisionsvermeidung unter alleiniger Verwendung von Ultraschallsensoren zur Hinderniserkennung finden sich in der Diplomarbeit von Thilo Müller [A01] sowie einer Veröffentlichung auf der UAVveek 2012 [P2]. Weitere Experimente zur Evaluierung der Hinderniserkennung mittels Infrarot und Ultraschall finden sich in der Bachelorarbeit von Paul Benz [A10].

## 8.7 Autonomer Flug

### Versuchsaufbau und Beschreibung

Zur Evaluierung des autonomen Fluges fliegt der Quadrokopter ein Nikolaushaus autonom ab (Abbildung 8.21 links). Die dazugehörigen Wegpunkte werden per Telekommando vor dem Flug übertragen und der Quadrokopter agiert während des gesamten Experiments autonom. Zur Positionsbestimmung kommt der OF Sensor zum Einsatz.

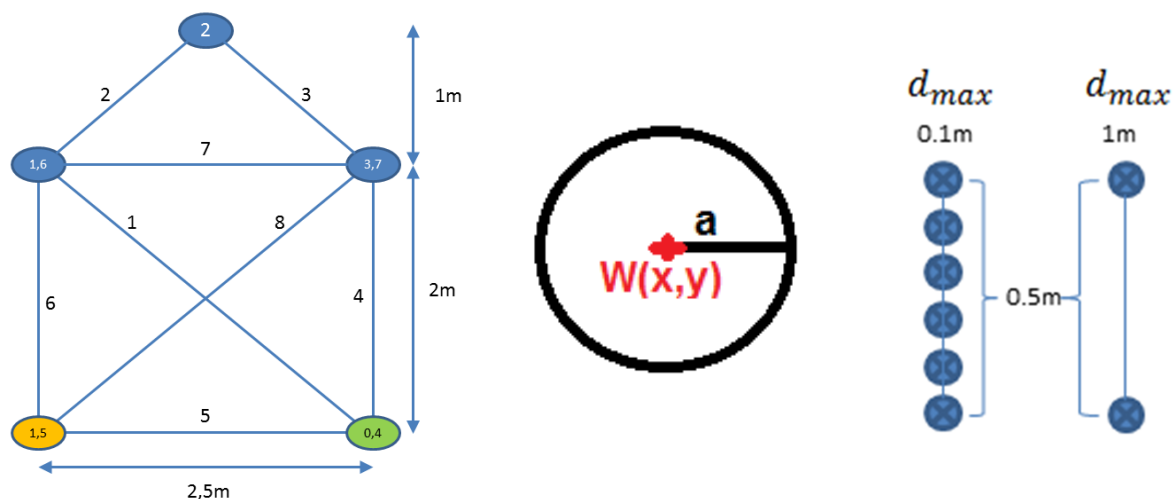


Abbildung 8.21 Links: Wegpunkte als Nikolaushaus mit Start (grün) und Ziel (gelb)

Mittig: Akzeptanz-Schwelle  $a$  um einen Wegpunkt  $W(x,y)$

Rechts: Zwei Pfade mit Wegpunkt Abstand  $d_{max} = 0.1m$  und  $d_{max} = 1m$

Während des Flugs wird die Position mit dem OF Sensor und dem OTS PPT X4 von WorldViz aufgezeichnet [126]. Das OTS dient lediglich der Evaluierung. Es werden mehrere Versuche mit unterschiedlichen Parametern für die Wegpunktsteuerung unternommen. Die Akzeptanz-Schwelle  $a$

gibt an, wie genau ein Wegpunkt  $W(x,y)$  erreicht werden muss (Abbildung 8.21 mittig). Ein Wegpunkt ohne Abweichung zu erreichen ist praktisch unmöglich oder kostet sehr viel Zeit. Durch Einstellen der Akzeptanz-Schwelle  $a$  kann ein Trade-Off zwischen Genauigkeit und Schnelligkeit gefunden werden.

Der Wegpunktabstand  $d_{\max}$  gibt an, wie weit zwei Wegpunkte maximal auseinander liegen dürfen. Liegen zwei Wegpunkte weiter auseinander, als der Wegpunktabstand  $d_{\max}$  beträgt, werden zusätzliche Wegpunkte eingefügt (Abbildung 8.21 rechts). Die Idee dabei ist es, mit Hilfe von  $d_{\max}$  die Genauigkeit des autonomen Flugs zu steigern.

Zur Bewertung des Fluges werden zwei Größen bestimmt. Die Zeit  $t$  ist die Dauer zwischen Start des Experiments und dem Erreichen des letzten Wegpunkts. Die Genauigkeit  $A$  ist eine Größe, die angibt, mit welcher Abweichung das Nikolaushaus abgeflogen wurde. Sie errechnet sich aus dem mittleren Abstand über allen gemessenen Positionen (OTS) und den dazugehörigen nächsten Punkten des Wegpunkte-Pfades.

### Auswertung des Versuchs

Bei vorangegangenen Sichtungsversuchen wurden die besten Ergebnisse mit einer Akzeptanz-Schwelle  $a$  im Bereich von 15cm bis 25cm erreicht. Daher wurden Versuche mit  $a = 15$  cm,  $a = 20$  cm und  $a = 25$ cm unternommen. Beim Wegpunktabstand  $d_{\max}$  wurden die Werte 3m, 1m, 0,5m und 0,01m untersucht. Insgesamt wurden 84 Experimente unternommen.

**Tabelle 8.6: Ergebnisse der Versuche zum autonomen Flug**

$d_{\max}$ [m]	$a$ [cm]	$t$ [s]					$A$ [m]				
3	15	43	65.4	41.4	46.3	42.4	0.27	0.29	0.24	0.28	0.29
	20	41.5	39.6	41.2	40.3	40.5	0.17	0.18	0.29	0.35	0.3
	25	36.3	40.5	36.9	40.6	34.3	0.53	0.27	0.32	0.17	0.22
1	15	80.8	83.8	91.4	75.7	120	0.44	0.34	0.32	0.24	0.31
	20	75.9	50.5	53.5	70.8	53.6	0.18	0.14	0.13	0.3	0.25
	25	66.6	49	48.6	49.3	46.8	0.36	0.18	0.16	0.25	0.25
0.5	15	155	127	110	125	116	0.47	0.19	0.23	0.3	0.13
	20	x	92.4	x	-	-	x	0.18	x	-	-
	25	x	x	-	-	-	x	x	-	-	-
0.01	25	144	143	-	-	-	0.2	0.23	-	-	-

Jede Parameterkombination wurde in einer Versuchsreihe 5 Mal wiederholt. Es zeigte sich, dass das Verhalten des Quadropters beim autonomen Flug ganz entscheidend von den diskutierten Parametern abhängt. Einige Parameter führten zu einem Scheitern des Versuchs, d.h. der Quadropters erreichte den letzten Wegpunkt nicht. Solche Versuche wurden mit einem Kreuz "x" markiert. Wenn Versuche derselben Reihe mehrfach bereits sehr schlechte Ergebnisse lieferten, wurde die jeweilige Reihe nicht mehr abgeschlossen. Dies wurde in der Tabelle mit einem Strich "-" vermerkt.

Die Ergebnisse der Versuche finden sich in Tabelle 8.6. Dabei zeigt sich eine vergleichsweise große Streuung zwischen den Resultaten derselben Versuchsreihe. Allein deshalb ist ein einzelner Versuch wenig aussagekräftig. Dies zeigt, dass das Verhalten beim autonomen Flug auch noch von weiteren Faktoren abhängt, als den hier betrachteten Parametern. Dennoch liefern die Experimente zum autonomen Flug neue wissenschaftliche Erkenntnisse, die sich durch Bildung der Mittelwerte über

die Versuchsreihen besser ablesen lassen (Tabelle 8.7).

**Tabelle 8.7: Zusammengefasste Ergebnisse zum autonomen Flug**

Mittelwert	$d_{\max}$ [m]	a [cm]		
		15	20	25
t [s]	3	47.7	40.62	37.72
	1	90.34	60.82	52.6
	0.5	126.46	x	x
A [m]	3	0.274	0.258	0.302
	1	0.33	0.2	0.24
	0.5	0.264	x	x

Wie zu erwarten, bewirken eine höhere Akzeptanz-Schwelle  $a$  und ein niedrigerer Abstand  $d_{\max}$  einen schnelleren Flug. Gleichzeitig aber bedeuten mehr Wegpunkte in Folge eines niedrigeren Abstandes  $d_{\max}$  zwar zunächst tendenziell eine leichte Genauigkeitssteigerung, treiben aber im Wesentlichen die Versuchszeit in die Höhe. Dabei führen Werte kleiner und gleich einem Wegpunkt Abstand  $d_{\max}$  von 0,5m im Vergleich zu einem Wert von 3m zu einer im Mittel ca. drei Mal längeren Versuchsdauer oder einem Scheitern des Versuchs. Die gewünschte Verbesserung stellt sich nicht dar.

Dieser Umstand ist zunächst unerwartet, da sich das System aus regelungstechnischen Überlegungen konträr verhalten sollte. Die Erklärung für diesen Umstand ist im Sensoraufbau des Systems zu finden. Bei Lageänderungen misst der OF-Sensor Positionsfehler. Häufige, sprunghafte Änderungen, wie diese durch neue Sollwerte erfolgen, führen zu stärkeren Fehlern. Daher sind im hier betrachteten Fall des autonomen Systems, das seine Positionssensoren mitführt, allzu häufige Sollwertänderungen der Positionsregelung bzw. zusätzliche Wegpunkte kontraproduktiv für die Genauigkeit.

Zudem lässt sich feststellen, dass die Akzeptanz-Schwelle  $a$  die Genauigkeit nur bis zu einem bestimmten Grad verbessert. Dies liegt daran, dass ein niedriger Wert eine sehr genaue Positionierung voraussetzt, die aufgrund der Dynamik des Systems und anderer Faktoren nicht umsetzbar ist. Ein zu niedriger Wert für die Akzeptanz-Schwelle  $a$  steigert auch deshalb die Genauigkeit nicht, weil in einem solchen Fall Wegpunkte nicht erreicht und mehrfach angeflogen werden.

Das Verhalten des Systems und die Auswirkungen der Parameter sind auch an Hand von Abbildung 8.22 klar erkennbar. Bei einem hohen  $d_{\max}$  von 3m und einem niedrigen  $a$  von 25cm verläuft der Versuch schnell und in wenigen geradlinigen Bewegungen (Versuch A). Der Versuch 1 dieser Versuchsreihe dauerte 36,3s. Je niedriger  $d_{\max}$  ist, desto weniger glatt verlaufen die Bahnen. Dies sieht auf den ersten Blick zwar für das menschliche Auge weniger genau aus, die Zahlen sagen aber das Gegenteil. Die Genauigkeit von Versuch A beträgt 0,53 m, während die beiden Genauigkeiten für die Versuche C (0,3m) und D (0,2m) deutlich besser ausfallen. Die beste Genauigkeit in diesem Vergleich erzielt Versuch B mit einem Betrag von 0,14m.

Die Erklärung lässt sich auch aus Abbildung 8.22 ablesen. Verpasst der Quadrokopter einen Wegpunkt, so dreht er um und muss den Wegpunkt noch einmal anfliegen. Das sieht auf den Grafiken unschön aus, schlägt sich aber kaum in der berechneten Genauigkeit nieder, sondern hauptsächlich in der Zeit. Die Flugzeit von Versuch D war mit 144s ca. 4 Mal so lang wie die Dauer

von Versuch A. Damit wurde bei Versuch D dieselbe Strecke im Mittel mit einer 4-fach langsameren Geschwindigkeit abgeflogen als bei Versuch A.

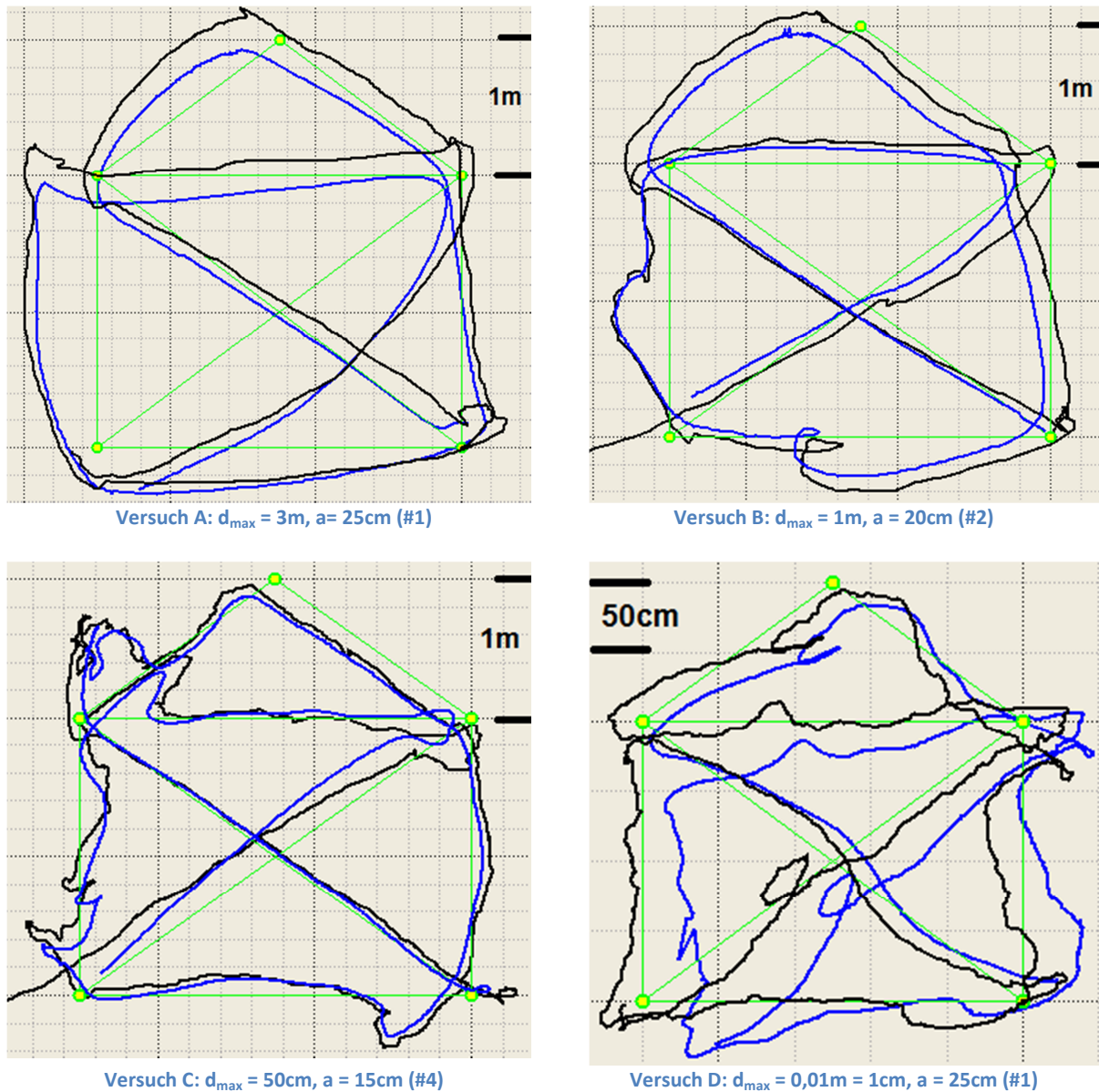


Abbildung 8.22 Pfade von vier autonomen Flügen (grün: Wegpunkte) aufgezeichnet mit dem OTS (blau) und dem OF-Sensor (schwarz) mit unterschiedlichen Parametern. Die Parameter stehen jeweils unter den Bildern und die Nummer des Versuchs der jeweiligen Reihe steht in Klammern dahinter.

Als Ergebnis der Auswertung kann festgehalten werden, dass die Fähigkeit des Quadropters zum autonomen Flug nachgewiesen ist. Darüber hinaus konnte der Einfluss der Akzeptanz-Schwelle  $a$  und des Wegpunktabstandes  $d_{\max}$  demonstriert werden. Als optimale Parameter unter den Anforderungen an Geschwindigkeit und Genauigkeit ergeben sich die Zahlen  $a = 20\text{cm}$  und  $d_{\max} \geq 1\text{m}$ . Der optimale Wert für  $a = 20\text{cm}$  ergibt sich auch aus der Standardabweichung beim Halten der Position, die experimentell  $16\text{cm}$  ergab. Aus diesen Gründen haben niedrigere Werte für die Akzeptanz-Schwelle  $a$  schlechte Ergebnisse erzielt.



## 8.8 Objektsuche

### 8.8.1 Überblick

In insgesamt 72 Experimenten wurde die Objektsuche evaluiert. Ziel der Evaluierung ist der Nachweis der Machbarkeit sowie die Diskussion der Such-Parameter. Die diskutierten Parameter sind der virtuelle Sichtbereich FVOV, der Maskierbereich MA und die Suchstrategie (BFS, DFS). Alle Experimente wurden zweimal durchgeführt.

Es wurden zwei unterschiedliche Versuchsaufbauten untersucht. Beim einfachen Aufbau lagen im Suchbereich an entgegengesetzten Enden zwei Zielobjekte (Ziele, Targets) aus. Beim komplexen Aufbau lagen 8 Zielen aus, die teilweise so nah beieinander lagen, dass sie auf demselben Kamerabild zu sehen waren.

### 8.8.2 Einfacher Versuch zur Suche

#### Versuchsaufbau und Beschreibung

Bei diesem Versuch bestand der Suchbereich aus einem Rechteck der Größe 3m x 2m mit zwei Zielen. Die beiden Ziele (rote Bälle) wurden willkürlich in die linke untere sowie die rechte obere Ecke platziert. Mit der linken unteren Ecke als Nullpunkt entspricht dies den Koordinaten  $P_1 = (0,5m; 0,5m)$  und  $P_2 = (2,4m; 1,4m)$  (Abbildung 8.23).

Für den Maskierbereich wurden die Werte  $MA = 0,1m$ ,  $MA = 0,15m$ ,  $MA = 0,2m$  und  $MA = 0,3m$  untersucht. Als VFOV wurden die Einstellungen  $0,3m \times 0,45m$ ,  $0,4m \times 0,6m$  und  $0,6m \times 0,9m$  untersucht. Dabei beträgt das reale FOV der Kamera ca.  $0,65m \times 0,45m$ .

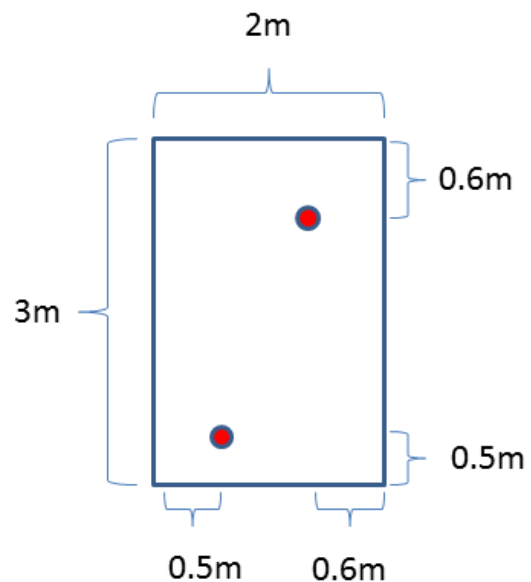


Abbildung 8.23: Objektsuche - Einfacher Versuchsaufbau

#### Auswertung des Versuchs

Die Resultate des Versuchs stehen in Tabelle 8.8. Hierbei steht  $dx$  für den mittleren Fehler in x-Richtung und  $dy$  für den mittleren Fehler in y-Richtung. Die Fehler sind dabei über die gemittelte Abweichung der beiden Ziele zwischen gemessener Position und wahrer Position errechnet. Dabei kann die wahre Position durchaus einen systematischen Fehler von ein paar Zentimetern enthalten. Bei zwei Versuchen wurden beide Ziele übersehen (F, Fehlerkennung).

Da bei diesem Versuchsaufbau bereits bei einem Maskierbereich von 0,2m keine Fehler auftraten, die in Zusammenhang zu MA standen, wurde der Versuch für  $MA = 0,3m$  ausgelassen, da dadurch keinerlei weitere Erkenntnisse gewonnen wären.

Aus den Resultaten in Tabelle 8.8 lässt sich keine eindeutige Präferenz für die Suchstrategie noch für den optimalen Parametersatz ableiten. Jedoch lässt sich eine Aussage über den mittleren Fehler einer Achse treffen, der kleiner als 15cm beträgt. Bei genauerer Betrachtung zeigt sich, dass bei diesem Versuchsaufbau eine Reihe zufälliger Faktoren, wie die Flugbahn in Relation zu den Zielen, einen großen Einfluss haben. Dieser Problematik wird durch den zweiten Versuchsaufbau

entgegengewirkt.

**Tabelle 8.8: Ergebnisse (Mittlere Fehler) des einfachen Versuchs**

Mittlere Fehler [cm]		DFS				BFS			
		Versuch 1		Versuch 2		Versuch 1		Versuch 2	
MA	VFOV	dx	dy	dx	dy	dx	dy	dx	dy
10	30-45	-8	16	-15	19	-24	20	-8	9
	40-60	-11	8	-19	8	-12	26	-19	19
	60-90	F	F	-10	11	-17	15	-14	13
15	30-45	-18	18	-27	7	-15	20	-19	20
	40-60	-28	8	-9	14	-4	19	-14	18
	60-90	-13	15	-3	12	-22	30	-16	12
20	30-45	-15	12	-14	23	-11	15	-12	20
	40-60	-39	4	-4	7	-35	34	-10	17
	60-90	-10	14	-10	17	-8	22	-11	9
30	30-45	-10	12	-	-	-21	29	-	-
	40-60	-15	11	-	-	-6	17	-	-
	60-90	F	F	-	-	-12	13	-	-
<b>Total</b>		<b>-17</b>	<b>12</b>	<b>-12</b>	<b>13</b>	<b>-16</b>	<b>22</b>	<b>-14</b>	<b>15</b>

Aussagekräftiger als die Positionsfehler sind die in Tabelle 8.9 akkumulierten Falscherkennungen, aufgetragen jeweils zu den untersuchten Parametersätzen. Dabei können Ziele fälschlich doppelt erkannt (D: Doppelerkennung) oder übersehen (F: Fehlerkennung) werden.

**Tabelle 8.9: Falschkennungen des einfachen Versuchs: Fehlerkennung (F) und Doppelerkennung (D)**

Detection Failures		DFS		BFS	
		F	D	F	D
VFOV	30-45	0	0	0	3
	40-60	0	1	1	0
	60-90	4	0	2	1
MA	10	2	1	1	3
	15	0	0	1	1
	20	0	0	1	0
	30	2	0	0	0

Hieraus lassen sich ein paar klare Aussagen treffen. Ein VFOV von 0,4m x 0,6m und größer führt zu Fehlerkennungen, da nicht mehr der gesamte Suchbereich von der Kamera erfasst wird. Ein geeignetes VFOV von 0,3m x 0,45m führt dagegen zu keinen Fehlerkennungen.

Ein zu niedrig eingestellter Maskierbereich MA führt zu Doppelerkennungen. Die untere Schranke für MA orientiert sich dabei am Positionsfehler, der ca. 15cm beträgt. Beim Vergleich der Suchstrategien fällt auf, dass bei der DFS zwei Parametersätze ohne Fehlerkennungen den Versuch bestehen,

während dies bei keinem Parametersatz der BFS der Fall ist. Daraus lässt sich bereits eine Präferenz für die DFS herleiten.

### 8.8.3 Komplexer Versuch zur Suche

#### Versuchsaufbau und Beschreibung

Der komplexe Versuchsaufbau wurde mit Hilfe der Erfahrungen aus dem einfachen Versuchsaufbau erstellt (Abbildung 8.24). Dabei wurden 8 Ziele im gesamten Suchbereich verteilt bzw. in einem Fall nahe beieinander platziert, um den Einfluss des Maskierbereichs genauer analysieren zu können. Tabelle 8.10 listet die Positionen der Ziele auf.

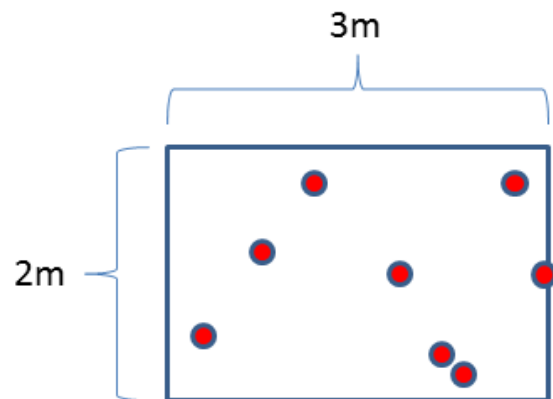


Abbildung 8.24: Objektsuche - Komplexer Versuchsaufbau

Der Suchbereich wurde ebenfalls verändert, um den Einfluss der Suchstrategie klarer herausarbeiten zu können. Mit einem Suchbereich von 2m x 3m, der im Vergleich zum einfachen

Versuchsaufbau mehr in die Breite geht, wurde probiert, der BFS entgegen zu kommen. Damit sollte die Hypothese, das schlechtere Abschneiden der BFS liegt am Suchbereich, negiert werden.

Tabelle 8.10: Positionen der 8 Ziele (Komplexer Versuchsaufbau)

Target Positions [cm]					
	X	Y		X	Y
<b>Position 1</b>	60	20	<b>Position 5</b>	100	180
<b>Position 2</b>	180	280	<b>Position 6</b>	20	240
<b>Position 3</b>	120	60	<b>Position 7</b>	40	220
<b>Position 4</b>	180	100	<b>Position 8</b>	100	300

#### Auswertung des Versuchs

Tabelle 8.11 listet die mittleren Positionsfehler pro Achse sowie die über beide Achsen bestimmten euklidischen Positionsfehler (E) auf. Der mittlere euklidische Positionsfehler beträgt für die Tiefensuche (DFS) 16cm und für die Breitensuche (BFS) 20cm. Daraus lässt sich bereits ablesen, dass DFS genauer als BFS ist.

Deutlicher wird dieser Unterschied bei Betrachtung der Falscherkennungen (Tabelle 8.12). Die DFS liefert 10 Falscherkennungen in 9 Experimenten, während die BFS bei denselben Einstellungen auf 20 Falscherkennungen kommt. Bei der BFS gibt es keinen Parametersatz, der ohne Falscherkennungen auskommt, während es vier Parametersätze bei der DFS gibt, die genau die 8 Ziele korrekt detektiert haben. Aufgrund dieser Diskrepanz wurde für die BFS ein VFOV von 25cm x 35cm zusätzlich erprobt. Dies führte zu noch schlechteren Resultaten, wie der Tabelle zu entnehmen ist.

Die Ergebnisse zusammengefasst, lässt sich aussagen, dass die untersuchten Parameter entscheidend für das Resultat der Objektsuche sind. Einfluss auf die Resultate, insbesondere die Genauigkeit, haben zwar noch weitere Faktoren, ohne optimale Suchparameter versagt die Objektsuche jedoch oder liefert schlechte Ergebnisse.

Tabelle 8.11: Ergebnisse (Mittlere Fehler) des komplexen Versuchs

Mittlere Fehler [cm]		DFS			BFS				
		Mittelwert			Mittelwert			Max	
MA	VFOV	dx	dy	E	X	Y	E	X	Y
15	25-35	-			-19	8	21	32	24
	30-45	-17	16	24	-22	17	30	23	31
	40-60	-9	5	12	-13	9	18	23	17
	60-90	-6	9	11	-12	13	19	18	26
20	25-35	-			-17	2	22	26	23
	30-45	-7	6	13	-17	17	25	28	32
	40-60	-23	0	24	-23	19	31	35	31
	60-90	-4	5	8	-7	4	10	13	10
30	25-35	-			-14	7	17	27	12
	30-45	-12	-5	13	-15	9	18	20	19
	40-60	-6	13	17	-17	11	21	24	21
	60-90	-16	15	22	-9	5	13	16	13
<b>Average:</b>		<b>-11</b>	<b>7</b>	<b>16</b>	<b>-15</b>	<b>10</b>	<b>20</b>		

Tabelle 8.12: Falschkennungen des komplexen Versuchs: Fehlerkennung (F) und Doppelerkennung (D)

Detection Failures		DFS		BFS	
MA	VFOV	M	D	M	D
15	25-35	Skipped		2	4
	30-45	0	2	2	0
	40-60	0	0	1	0
	60-90	2	0	4	0
20	25-35	Skipped		3	2
	30-45	0	0	1	0
	40-60	0	0	2	0
	60-90	2	0	4	0
30	25-35	Skipped		2	0
	30-45	0	0	1	0
	40-60	1	0	1	0
	60-90	3	0	4	0

Aus der Verteilung der Falscherkennungen lassen sich die optimalen Parametersätze ebenfalls ablesen (Abbildung 8.25). Bei einem MA von 15cm bzw. einem MA von 30cm kommt es in jedem Fall zu Falscherkennungen. Dies liegt daran, dass bei einem zu niedrigen Maskierbereich aufgrund der Positionsfehler des Systems Doppelerkennungen auftreten. Bei einem zu hohen Maskierbereich verdeckt eines der beiden nahe beieinander liegenden Ziele das andere. Dabei ist zu bedenken, dass die nahen Ziele auf jeder Achse lediglich 20cm auseinanderliegen, womit ein MA von 30cm schon

theoretisch zu Fehlern führen muss.

Für einen MA von 20cm treten Falscherkennungen bei einem VFOV von 60cm x 90cm auf, da in diesem Fall nicht mehr der gesamte Suchbereich abgeflogen wird. Bei einem MA von 20cm und einem FVOV von 30cm x 45 cm sowie 40cm x 60cm erkennt die DFS alle 8 Ziele stets korrekt.

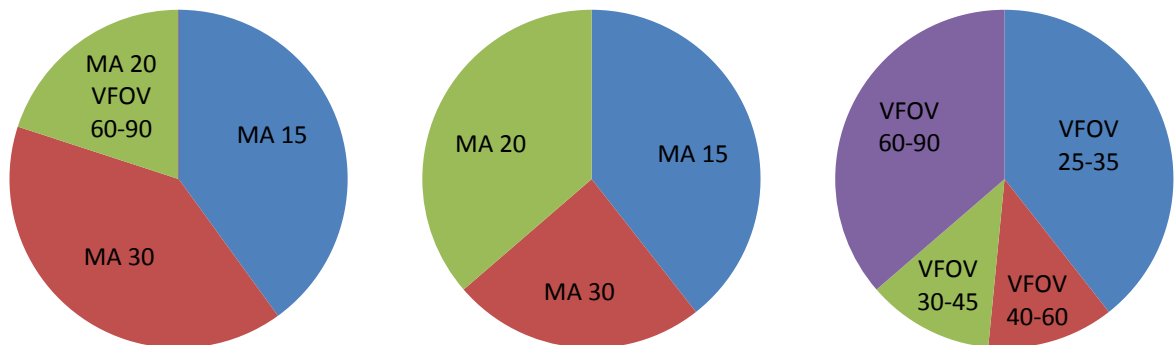


Abbildung 8.25: Verteilung der Falscherkennungen für DFS (links) und BFS nach Maskierbereich (mittig) und VFOV (rechts)

In Abbildung 8.26 und Abbildung 8.27 sind ein Foto des komplexen Versuchsaufbaus sowie ein GUI-Screenshot mit einer Darstellung der Ergebnisse der Objektsuche gegenübergestellt.

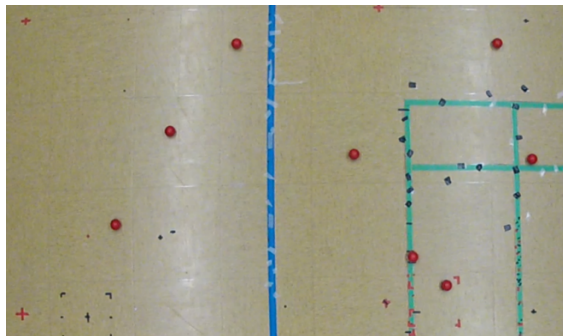


Abbildung 8.26: Foto des komplexen Versuchsaufbaus (Objektsuche)

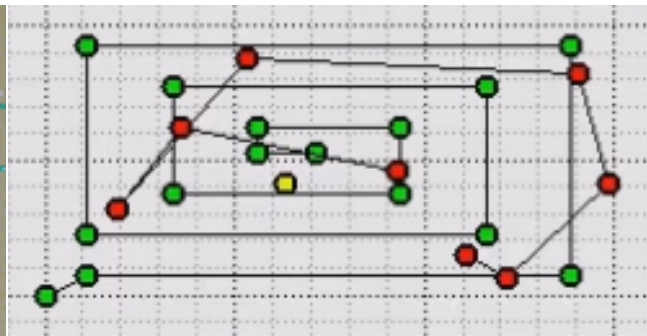


Abbildung 8.27: GUI Screenshot der Ergebnisse der Objektsuche mit MA = 20cm und VFOV = 0,3m x 0,45m. Dabei sind gefundene Ziele als rote Punkte, Wegpunkte als grüne Punkte und die Position des Quadropters als gelber Punkt dargestellt

Abschließend lässt sich festhalten, dass die BFS ungeeignet für das betrachtete Problem ist, wohingegen die DFS das Problem gut löst. Die Erklärung hierzu ist dieselbe wie beim bereits diskutierten Problem zum autonomen Flug. Viele Wegpunkte in kurzen Abständen führen zu Positionsfehlern. Besser ist es, möglichst in langen, gleichgroßen Bahnen mit wenigen Wegpunkten größere Bereiche abzudecken. Die Resultate der Objektsuche wurden 2013 auf der UAV World vorgestellt [P6].

## 8.9 Gerichteter Flug

### 8.9.1 Überblick

Zur Evaluierung des gerichteten Flugs und der dazugehörigen Datenfusion aus OF und Fourier Tracking zur Positionsbestimmung wurden insgesamt 70 Experimente unternommen. Neben zwei

Versuchsreihen zum gerichteten Fliegen, bestehend aus einem einzelnen Wegpunkt und einer Wegpunktliste (multiple Rotationen mit anschließenden Translationen), wird auch die Leistungsfähigkeit der Datenfusion beim Flug unter schlechten Lichtverhältnissen analysiert.

## 8.9.2 Gerichteter Wegpunktflug

### Versuchsaufbau und Beschreibung

Bei diesem Versuch wurde die Position des Quadropters beim Durchführen des gerichteten Fluges bestehend aus einem einzelnen Punkt aufgezeichnet. Der dabei bestimmte Fehler errechnet sich aus der Differenz zwischen OTS und on-Board Datenfusion (kurz EST).

Ausgehend von der Startposition  $P_0 = (0m, 0m)$  wurden in je 5 Versuchen die beiden Zielpositionen  $P_1 = (0m; 1,5m)$  und  $P_2 = (1,5m; 1,5m)$  gerichtet angefliegen.

### Auswertung des Versuchs

Die Positionsfehler beim Erreichen der Zielposition sind in Tabelle 8.13 angegeben.  $E_x$  und  $E_y$  sind dabei die Fehler in x- und y-Richtung.  $\Delta F_p$  ist der euklidische Gesamtfehler. Aus den Daten wird ersichtlich, dass die Positionsfehler mit einem Mittelwert von ca. 19cm im Bereich von 12cm bis 27cm liegen. Damit ist der Fehler nach relativ kurzer Zeit schon beachtlich. Dabei gibt es eine ganze Reihe an Faktoren, die einen solchen Fehler verursachen, wie ein falscher Skalierungsfaktor bei der Positionsbestimmung und ein Ausrichtungsfehler der beiden Koordinatensysteme für OTS und Datenfusion (EST).

**Tabelle 8.13: Positionsfehler beim gerichteten Fliegen (Wegpunktflug)**

Positionsfehler	$P_1$			$P_2$		
	$E_x$ [cm]	$E_y$ [cm]	$\Delta F_p$ [cm]	$E_x$ [cm]	$E_y$ [cm]	$\Delta F_p$ [cm]
Versuch 1	16.6	5.9	17.6	-15.4	-6.0	16.5
Versuch 2	-13.1	-20.0	23.9	-6.1	-3.8	7.2
Versuch 3	-22.0	-15.5	26.9	-26.3	-0.7	26.3
Versuch 4	-8.4	-17.1	19.1	-10.8	-13.2	17.1
Versuch 5	-11.6	-17.4	20.9	-11.9	-4.2	12.6
Mittelwert	-7.7	-12.82	21.68	-14.1	-5.58	15.94

Qualitativ zeigten alle Versuche ein vergleichbares Verhalten, womit die Fähigkeit des gerichteten Fliegens nachgewiesen ist. Um einen detaillierten Einblick in das Systemverhalten zu ermöglichen, werden im Weiteren zwei Versuche exemplarisch betrachtet.

Die Graphen (Abbildung 8.28, Abbildung 8.29) zeigen, dass die Rotation ca. 2s dauert und der Zielpunkt nach ca. 5s erreicht ist. Die lila Linie stellt den Yaw-Winkel dar und zeigt das mustergültige Einschwingverhalten des erforderlichen Winkels zum gerichteten Fliegen. An der grünen Linie ist zu erkennen, dass das Fourier Tracking während der Drehung aktiv ist. Nach der Rotation erfolgt die Translation mit einem Überschwinger von ca. 30 – 50 cm.

Der mechanische Aufbau des Systems war ca. 2° über die Pitch-Achse fehlausgerichtet. Dies erklärt

die Positionsabweichung beim Starten in x-Richtung von ca. 25cm (Abbildung 8.29).

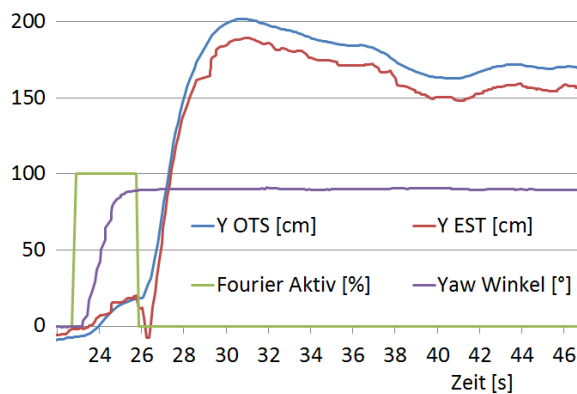


Abbildung 8.28: P<sub>1</sub> Versuch 5 (einfacher gerichteter Flug)

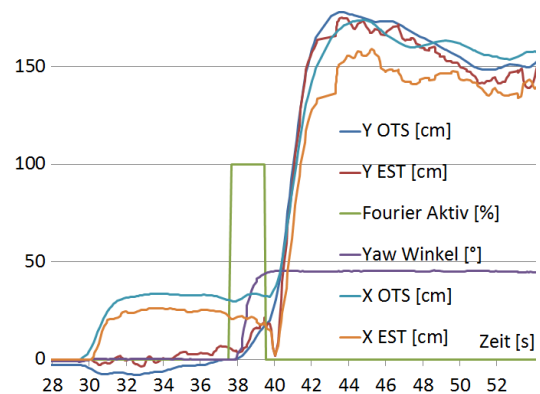


Abbildung 8.29: P<sub>2</sub> Versuch 4 (einfacher gerichteter Flug)

### 8.9.3 Gerichteter Wegpunktlistenflug

#### Versuchsaufbau und Beschreibung

Bei diesem Versuch führt der Quadrokopter einen gerichteten Flug bestehend aus einer Wegpunktliste autonom durch. Im Vergleich zum vorherigen Versuch führt der Quadrokopter nun multiple Rotationen (Yaw-Drehungen) und Translationen durch. Alle Translationen werden dabei gerichtet durchgeführt, d.h. das „Vorderteil“ des Quadrokopters zeigt in Flugrichtung. Als Wegpunktliste kommen ein Quadrat bestehend aus 4 Wegpunkten (Abbildung 8.30) und das Nikolaushaus (Abbildung 8.21 links) bestehend aus 8 Wegpunkten zum Einsatz, wobei jeder Versuch 5mal wiederholt wird.

Zur Evaluierung (Bestimmung des Fehlers) werden die mittels OTS und on-Board Datenfusion bestimmten Positionen und die Dauer des Versuchs bis zum Erreichen des letzten Wegpunktes aufgezeichnet. Dabei dient das OTS wiederum nur der Evaluierung.

#### Auswertung des Versuchs

Tabelle 8.14 fasst die Ergebnisse des Versuchs zusammen.  $E_x$  und  $E_y$  sind dabei die Fehler in x- und y-Richtung.  $\Delta F_p$  ist der euklidische Gesamtfehler. Der Positionsfehler am Ende des Versuchs liegt im Bereich von 12 – 34cm bei einem Ausreißer von 55cm. Der Mittelwert des Positionsfehlers für das Quadrat aus 4 Wegpunkten beträgt ca. 28cm und für das Nikolaushaus ca. 31cm. Die mittlere Versuchsdauer für das Quadrat beträgt 26s, während diese für das Nikolaushaus mit 78s mehr als doppelt so lange ist. Somit kann festgestellt werden, dass der Fehler mit der Zeit und der Anzahl der Wegpunkte leicht ansteigt und sich gleichzeitig nicht proportional dazu verhält. Die beiden Ausreißer (nach oben und unten) beim Flug des Nikolaushaus herausgerechnet, ergibt einen mittleren Fehler von ca. 29cm, der nur noch minimal über dem Fehler beim Quadrat liegt.

Der große Fehler von 55cm erforderte eine genauere Analyse der Daten. Die Auswertung der Daten ergab, dass der Fehler während einer schnellen Bewegung auftrat, die offensichtlich nicht korrekt erkannt wurde. Da sich die C270 als ungeeignet für das FT erwies, wird die C920 als Kamera verwendet. Eine bessere Kamera würde die Fehler weiter reduzieren.

Tabelle 8.14: Positionsfehler beim gerichteten Fliegen (Wegpunktlistenflug)

Positionsfehler	Quadrat				Nikolaushaus			
	$E_x$ [cm]	$E_y$ [cm]	t [s]	$\Delta F_p$ [cm]	$E_x$ [cm]	$E_y$ [cm]	t [s]	$\Delta F_p$ [cm]
Versuch 1	11,5	-21,8	25	24,7	-31.5	-11.2	75	33.4
Versuch 2	11,6	-22,6	25	25,4	21.1	10.5	55	23.5
Versuch 3	-13,6	-18,2	20	22,8	28.3	-11.5	140	30.6
Versuch 4	8,7	-32,9	35	34,0	-7.4	10.5	70	12.9
Versuch 5	18,8	-25,4	25	31,6	-53.3	-13.,4	50	55.0
Mittelwert	7,4	-24,8	26	27,7	-8,6	-3,0	78	31,0

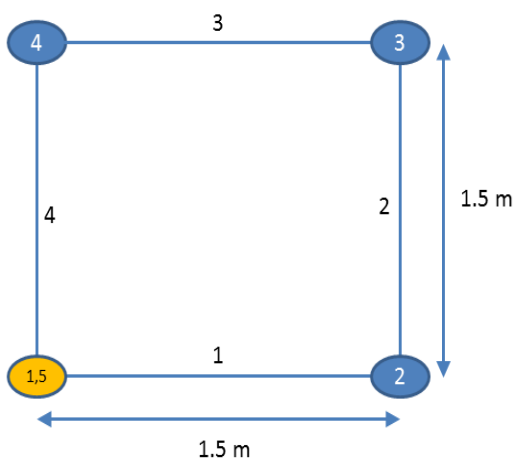


Abbildung 8.30: Wegpunkte als Quadrat, Start und Ziel sind gelb markiert

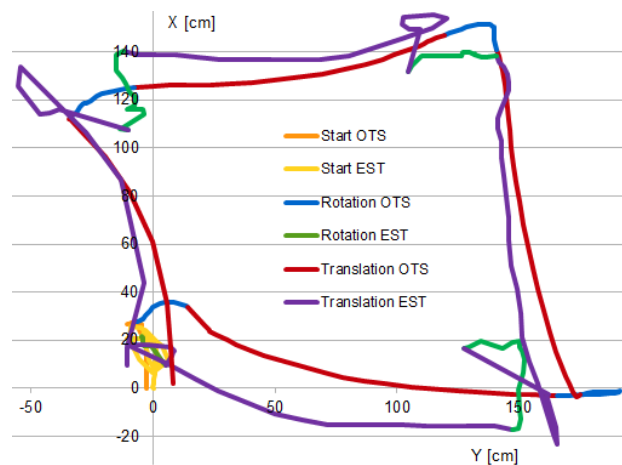


Abbildung 8.31: Pfade beim gerichteten Flug (Versuch 2) aufgezeichnet mit OTS und on-Board Datenfusion (EST) während der einzelnen Phasen Start, Rotation und Translation

Zur genaueren Analyse des Systemverhaltens werden die Ergebnisse einiger Versuche exemplarisch betrachtet. Die Abbildungen (Abbildung 8.31 und Abbildung 8.32) zeigen je einen Flug der beiden Wegpunktlisten mit einer Unterteilung in die verschiedenen Phasen des Flugs. Von Interesse ist dabei insbesondere die Trennung zwischen Rotation und Translation. Es ist zu erkennen, dass während der Rotation Positionsabweichungen und Positionsfehler entstehen. Diese werden nach der Rotation korrigiert.

Die Erklärung dazu liegt im implementierten Konzept für das gerichtete Fliegen. Die Messungen des Fourier Trackings gehen aus Synchronisationsgründen nur einmal nach der Drehung als Korrektur ein. Diese Korrektur ist in den Daten klar zu erkennen.

Zusammenfassend hat die Evaluierung den Nachweis erbracht, dass der AQopterI8 zum autonomen, gerichteten Fliegen fähig ist. Dabei steigt der Positionsfehler mit der Anzahl der Wegpunkte und der Gesamtlaufzeit an. Der Fehler verhält sich jedoch nicht linear. Doppelt so viele Versuche und eine mehr als doppelt so lange Laufzeit haben zu einem Fehleranstieg von ca. 10% geführt. Der Fehlerzuwachs ist dabei prinzipiell inhärent und rührt aus der Methodik der iterativen relativen Positionsbestimmung her. Dabei machen schnelle, fehlerhaft erkannte Bewegungen einen großen



Anteil des Positionsfehlers aus.

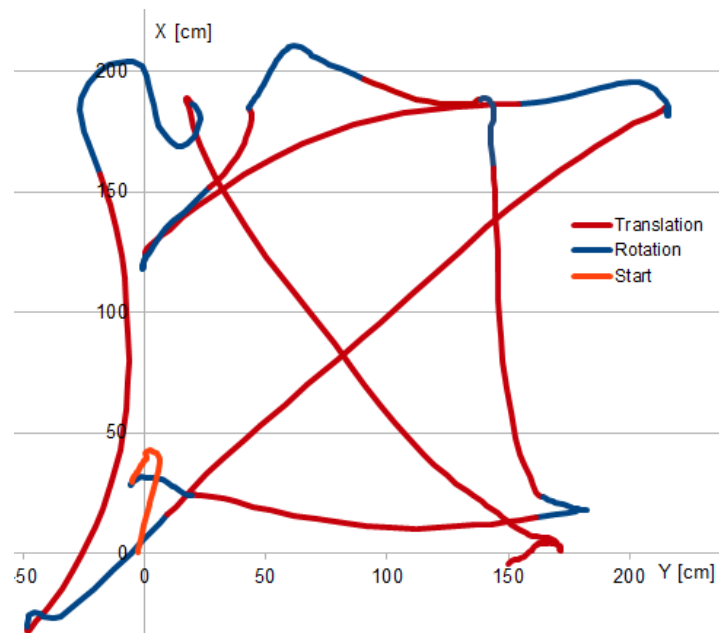


Abbildung 8.32: Pfade beim gerichteten Flug (Versuch 2) aufgezeichnet mit OTS während der einzelnen Phasen Start, Rotation und Translation

#### 8.9.4 Flug bei schlechten Lichtverhältnissen

##### Versuchsaufbau und Beschreibung

Neben der Befähigung zum gerichteten Fliegen ermöglicht das Fourier Tracking eine Positionsbestimmung unter schlechten Lichtverhältnissen, bei denen der OF Sensor versagt. Damit ist der Quadrocopter in der Lage, seine Position auch unter wechselnden und teilweise sehr schlechten Lichtverhältnissen zu bestimmen und zu regeln. Dies wird mit dem folgenden Versuchsaufbau im Flug evaluiert.

Die vier untersuchten Lichtverhältnisse sind in Abbildung 8.33 veranschaulicht. Im Fluglabor befinden sich vier Reihen Deckenleuchten, die per Schalter einzeln an- und ausgeschaltet werden können. Bei „guten“ Lichtverhältnissen sind alle Deckenleuchten an. Bei „schlechten“ Lichtverhältnissen ist die Deckenleuchte genau über dem Quadrocopter aus, eine zweite Deckenleuchte daneben sowie die beiden übrigen Deckenleuchten des Raumes sind an. Bei „sehr schlechten“ Lichtverhältnissen sind nur die beiden entferntesten Deckenleuchten an, wodurch im Versuchsbereich bzw. an dessen Rand ein Helligkeitsübergang von hell zu dunkel entsteht. Bei „total schlechten“ Lichtverhältnissen sind alle Deckenleuchten aus. Das Fenster ist bei allen Versuchen mit einem Vorhang abgedunkelt, wobei durch ein paar Schlitze Tageslicht ins Labor fällt.

Während des Versuchs hält der Quadrocopter seine Position mit Hilfe des OF Sensors und des Fourier Tracking. Als wahre Referenz wird die Position zusätzlich per OTS aufgezeichnet. Ziel des Experiments ist die Evaluierung der dynamisch gewichteten Datenfusion aus OF Sensor und Fourier Tracking zur Positionsbestimmung und Positionsregelung.

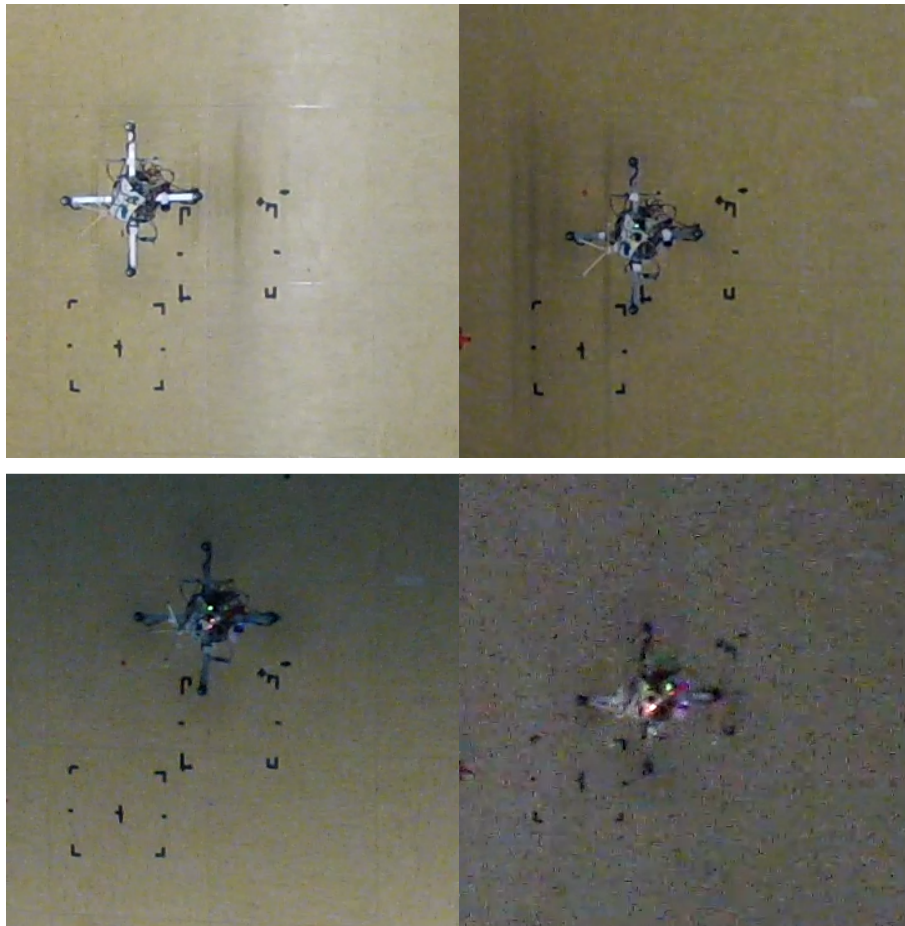


Abbildung 8.33: Untersuchte wechselnde Lichtverhältnisse (oben links: gut; oben rechts: schlecht; unten links: sehr schlecht; unten rechts: total schlecht)

Es wurden zu allen vier Lichtverhältnissen je 5 Versuche über ca. 120s durchgeführt, von denen 60s ausgewertet werden

### Auswertung des Versuchs

Die Ergebnisse des Versuchs aus Tabelle 8.15 und Tabelle 8.16 werden durch Abbildung 8.34 veranschaulicht.  $E_x$  und  $E_y$  sind die Positionsfehler in x- und y-Richtung am Ende des Versuchs.  $F$  ist der daraus bestimmte Gesamtfehler.  $\bar{P}$  ist der gemittelte Positionsfehler aller Gesamtfehler  $F$ . Der Positionsfehler unter guten Lichtverhältnissen ist signifikant kleiner als unter den schlechteren Lichtverhältnissen. Damit ist der Einfluss der Lichtverhältnisse auf den Positionsfehler gezeigt.

Auffällig und eher unerwartet ist das Ergebnis, dass der Positionsfehler unter schlechten Lichtverhältnissen größer ist als unter sehr schlechten und total schlechten Lichtverhältnissen. Obwohl also theoretisch weniger Licht zur Verfügung steht, ist der Fehler größer. Die Erklärung hierfür könnte das häufige Aktivieren und Deaktivieren des Fourier Trackings sein sowie die Datenfusion mit dem OF Sensor. Durch eine schlechte Synchronisation oder ein Versagen des OF Sensors können Fehler das Gesamtergebnis im Vergleich zum Ergebnis des Fourier Trackings verschlechtern. Die gesparte Rechenleistung kommt bei diesem Vergleich nicht zum Tragen.

Das Problem der Synchronisation der Messungen wird dadurch verschärft, dass die Messungen auf zwei unterschiedlichen Rechnern (UC3A, LP180) erfolgen. Eine explizite Synchronisation der Messungen könnte Abhilfe schaffen. Die Ursache des Problems könnte aber auch an nicht optimalen

Parametern für die Gewichtung und Datenfusion liegen.

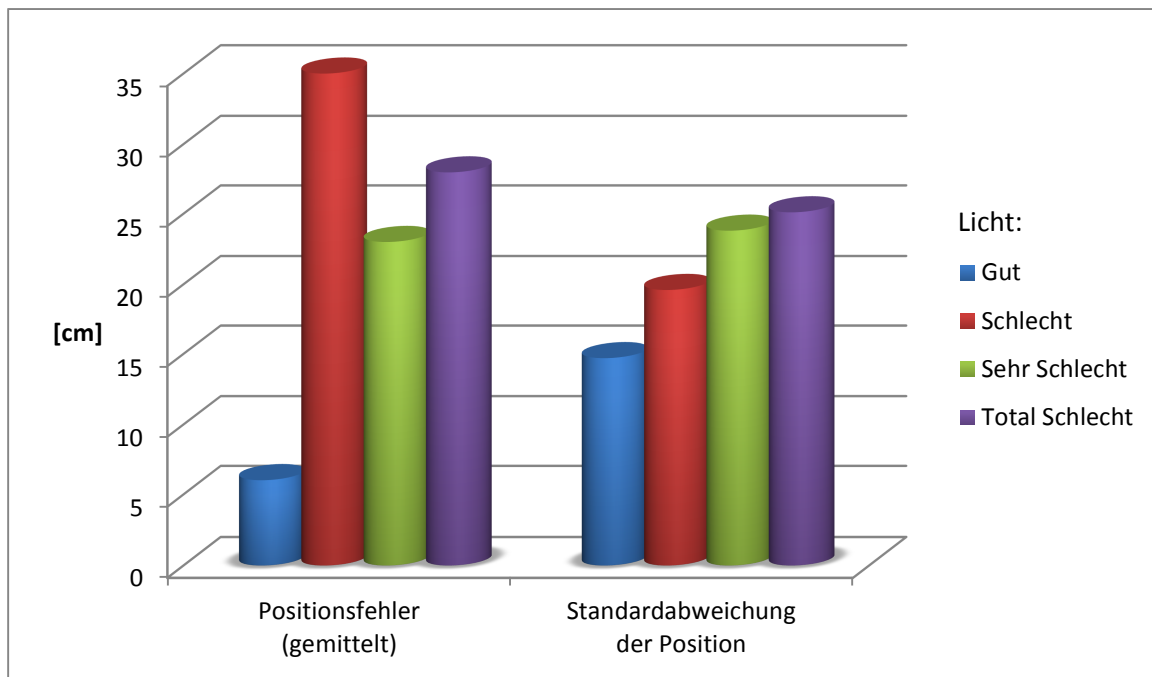


Abbildung 8.34: Einfluss der Lichtverhältnisse auf Positionsfehler (gemittelt) und Standardabweichung der Position

Tabelle 8.15: Positionsfehler bei verschiedenen Lichtverhältnissen

Fehler nach Licht [cm]	Versuch 1			Versuch 2			Versuch 3			Versuch 4			Versuch 5			$\bar{P}$
	$E_x$	$E_y$	F	$E_x$	$E_y$	F	$E_x$	$E_y$	F	$E_x$	$E_y$	F	$E_x$	$E_y$	F	
Gut	-1	-13	13	-3	-3	4	1	-2	2	-5	0	5	2	5	6	6
Schlecht	-47	-44	64	-31	-6	31	-48	-32	58	-28	-32	43	-2	-15	15	35
Sehr Schlecht	X -8	X -17	X 18	-11	-8	14	-4	-67	67	-12	-21	24	-6	-11	13	23
Total Schlecht	8	-35	36	2	-8	8	X -8	X 19	X 21	-16	13	20	-43	-67	80	28

Tabelle 8.16: Standardabweichung der Position bei verschiedenen Lichtverhältnissen

Standardabweichung nach Licht [cm]	Versuch 1		Versuch 2		Versuch 3		Versuch 4		Versuch 5		Mean
	X	Y	X	Y	X	Y	X	Y	X	Y	
Gut	6	19	16	18	12	13	21	18	11	13	14.7
Schlecht	19	29	12	14	23	24	18	20	26	15	19.6
Sehr Schlecht	20	26	23	20	24	46	15	25	16	23	23.8
Total Schlecht	28	36	19	19	24	22	28	22	24	29	25.1

Trotz des größeren Positionsfehlers lieferte das Gesamtsystem unter schlechten Lichtverhältnissen bessere Resultate als unter sehr schlechten und total schlechten Lichtverhältnissen. Dies hat die folgenden zwei Gründe. In Tabelle 8.15 sind solche Versuche mit einem Kreuz „x“ gekennzeichnet, die beim ersten Anlauf nicht abgeschlossen werden konnten, weil die Positionsbestimmung versagte, so dass der Quadropter den Versuchsbereich verlies. Diese Versuche wurden dann wiederholt. Dies trat einmal unter sehr schlechten und einmal unter total schlechten Lichtverhältnissen auf.

Darüber hinaus bekam der Quadropter mit Verschärfung der Lichtverhältnisse mehr Schwierigkeiten, seine Position ruhig zu regeln. Dies wird an Hand der Standardabweichung der Position ersichtlich (Abbildung 8.34). Die Erklärung hierzu liegt in der Wiederholrate der Sensorik und dem Verhalten der Regelung. Der OF Sensor liefert neue Messungen mit einer Wiederholrate von 100Hz, während das Fourier Tracking (FT) lediglich 10 Hz schafft. Daher wird das Regelverhalten bei Verlust des OF Sensors schlechter. Um das Problem abzuschwächen, müssen die Reglerparameter bei Verwendung des Fourier Tracking geändert werden. Da die Sensorik jedoch dynamisch wechselt, würde das System auch dynamisch zwischen verschiedenen Reglerparametern springen. Empirische Versuche ergaben, dass auch dies zu Instabilitäten führt. Daher ist ein Trade-Off mit leichter Anpassung der Reglerparameter beim Wechsel zwischen OF und FT implementiert.

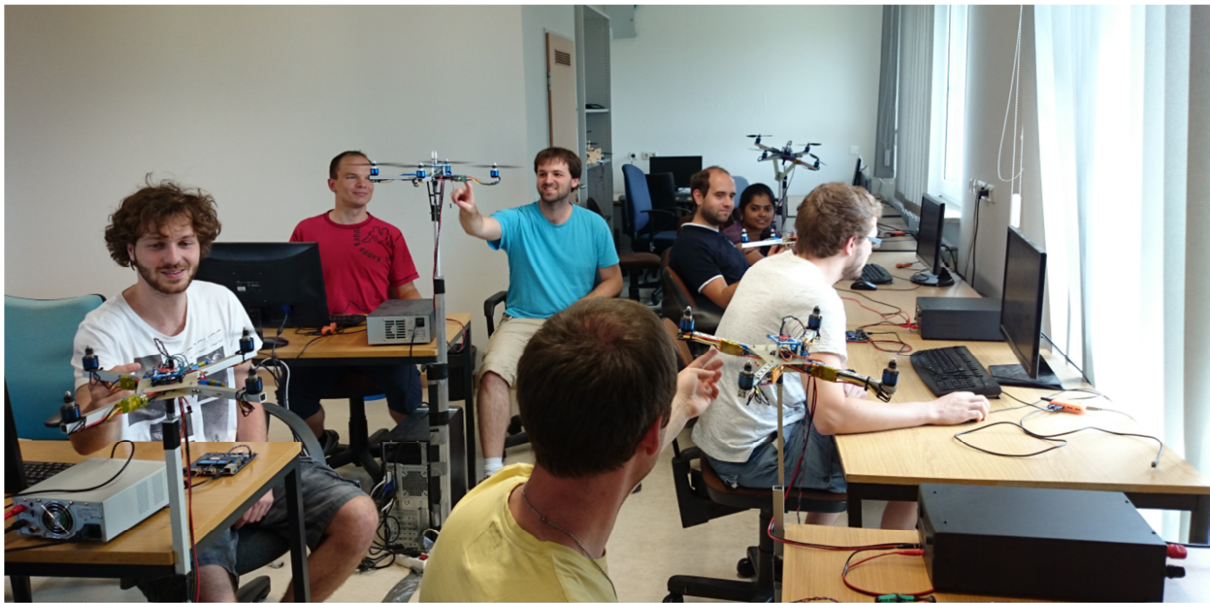
Diese und weitere Ergebnisse zum gerichteten Flug wurden in der Zeitschrift IJAI (International Journal of Artificial Intelligence and Applications, zu Deutsch: Internationale Zeitschrift für künstliche Intelligenz und Anwendungen) veröffentlicht [P8].





*We are what we repeatedly do.  
Excellence, then, is not an act, but a habit.*  
[Will Durant, Aristoteles]

## 9 Quadrocopter in der Lehre



## 9.1 Lehre Heute

Die naturwissenschaftlichen und technischen Studiengänge an Universitäten werden heute häufig sehr theoretisch unterrichtet. Theoretische Fächer, Vorlesungen und Übungen, bei denen Übungsblätter bearbeitet werden, machen den Hauptanteil des Studiums aus und werden ergänzt durch Praktika und Projektarbeit. Entsprechend machen die praktischen Übungen den deutlich geringeren Anteil aus. Dies führt dazu, dass der Ausbildung an den Hochschulen häufig der Praxisbezug fehlt. Darüber klagen viele Unternehmer [127].

Während also die theoretischen Kenntnisse vieler Hochschulabsolventen häufig hervorragend sind, fällt den Berufsanfänger oftmals die Anwendung dieses Wissens schwer. Dies ist aus vielen Beispielen bekannt und nicht zuletzt den Gesprächen mit Studenten entnommen. Die klassischen Vorlesungsfächer, die den Hauptteil des Studiums ausmachen, bestehen aus zwei bis vier Schulstunden Vorlesung in der Woche sowie meist zwei Schulstunden Übung. Die Vorlesungen sind in der Regel Frontalunterricht, häufig ohne Beteiligung der Zuhörer bzw. Studenten. In den Übungen werden Übungsblätter vorgerechnet, ebenfalls oft in Form von Frontalunterricht. Im Beruf sitzt heute aber kaum noch jemand mit Stift und Papier vor den Fragestellungen aus den Übungsblättern, sondern am PC. Dabei treten zunächst ganz andere Probleme auf, die in der Theorie gar nicht behandelt wurden. Hinzu kommt, dass die Anwendung von gelerntem Wissen in vielen Fällen nicht weniger schwierig ist als die Theorie selbst. Dieser Transfer wird heute aber eher unzureichend unterrichtet. Das führt dazu, dass die meisten Studenten sich das Nötige selbst beibringen oder ihr Studium ohne ausreichende Kenntnisse auf diesem Gebiet absolvieren.

Wo und inwiefern genau Theorie und Praxis auseinanderklaffen, ist noch eine weitere, wichtige Erfahrung, die vielen Studenten gar nicht oder nur unzureichend unterrichtet wird. Je nach beruflichem Werdegang sind praktische Fähigkeiten vordergründig auch gar nicht erforderlich. Das mangelnde Verständnis der Praxis, d.h. der Wirklichkeit, kann dann aber zu weitreichenden Fehleinschätzungen beisteuern. Denn letztlich unterliegt die Welt den wirklichen Gesetzmäßigkeiten, wovon die Theorie stets nur eine Abstraktion darstellt.

## 9.2 Konzept der Lehre

Als Ansatzpunkt zur Verbesserung dieser Problematik wurde im Rahmen der Dissertation und in Zusammenarbeit mit Prof. Montenegro mit dem Luft- und Raumfahrtlabor eine Veranstaltung mit mehr Praxisbezug entwickelt. Das Luft- und Raumfahrtlabor ist eine Veranstaltung bestehend aus Laborübungen und Vorlesungen und geht über ein Semester. Der Schwerpunkt und die Besonderheit der Veranstaltung sind die Laborübungen, die nun exemplarisch beschrieben werden. Als Beispiel dient dabei die Fassung aus dem Sommersemester 2015. Dabei sei darauf hingewiesen, dass sich die Inhalte jedes Semester leicht ändern, wobei grundsätzliche Teile erhalten bleiben.

Im Sommersemester 2015 bestanden die Laborübungen aus 10 wöchentlichen Übungen eingeteilt in 5 Blöcke. Der erste Block besteht aus zwei einführenden Veranstaltungen, in denen eine Einführung in CAD sowie Mikrocontrollerprogrammierung gegeben wird. In der CAD Übung zeichnen die Studenten ein 3D Modell des Quadropters am PC und drucken dieses anschließend mit dem 3D Drucker des Lehrstuhls aus. Während die CAD Übung einen motivierenden, einführenden Charakter hat, werden in der einführenden Mikrocontrollerübung die Grundlagen für die weiteren Laborübungen geschaffen. Dabei haben die Studenten vorab schon Grundkenntnisse in C/C++ sowie



im Mikrocontrollerprogrammieren.

Hauptteil des Labors ist das Entwickeln einer eigenen Software für einen Quadrocopter, bestehend aus Funktionen zur Signalverarbeitung, Regelung, Steuerung und Kommunikation. Damit die Studenten diese anspruchsvolle Aufgabe überhaupt in der vorgegebenen Zeit erledigen können, wird den Studenten mit einem selbstentwickelten und leicht konfigurierbaren Framework eine Hilfe gegeben. Das Framework enthält dabei fertige Treiber und Bibliotheken, die das Arbeiten erleichtern und beschleunigen sollen.

Der zweite Block besteht aus einer Übung zur IMU und Lagebestimmung. In dieser Übung soll die Lage des Quadrocopters aus den Daten der IMU und unter Verwendung von Quaternionen bestimmt werden.

Der dritte Block hat das Thema Kommunikation zur Grundlage und besteht aus je einer Übung zur Telemetrie und zur Telekommandierung. In diesen beiden Übungen wird mittels Qt eine Bodenstation entwickelt, mit der Telemetriedaten empfangen und Telekommandos verschickt werden können.

Thema der vierten Übung ist die Signalverarbeitung zur Orientierungsbestimmung, d.h. Datenfusion bzw. Filtern der IMU Daten zur Lagebestimmung. Dazu werden in den beiden Übungen ein Kalman Filter und ein KFI (Kalman Filter für Informatiker) von den Studenten implementiert. Der KFI ist eine von Prof. Montenegro entwickelte vereinfachte Form des Kalman Filters und verhält sich vergleichbar einem komplementären Filter.

Alle beschriebenen Übungen zielen schließlich auf den letzten, fünften Block, in dem die Studenten mit Hilfe der QCS (Lehraufbauten) die Steuerung und Regelung eines Quadrocopters programmieren, parametrisieren und optimieren. Dieser Block ist gegliedert in drei Übungen. Thema der ersten Übung ist die Kippregelung über eine Achse (1 DOF Roll bzw. Pitch). In der zweiten Übung werden die Yaw-Regelung (1DOF) und die Superposition (Mehrgrößenregelung, 2DOF Roll + Yaw) thematisiert. In der dritten Übung wird schließlich auf die 3 DOF Regelung sowie deren Steuerung eingegangen. Diese Reihenfolge ist sehr wichtig, da eine unmittelbare 3 DOF Regelung nicht nur schwierig in der Umsetzung und Parametrisierung ist, sondern ein hohes Unfallrisiko für Material und Personen mit sich bringt.

## 9.3 Evaluierung der Lehre

### 9.3.1 Umfrage und Fragebogen

Zur Evaluierung der Lehre wurde eine Online-Umfrage bestehend aus 20 Fragen durchgeführt. Ziel der Umfrage war es, zu erfahren, wie die Studenten das Luft- und Raumfahrtlabor, insbesondere die Laborübungen zum Thema Quadrocopter, in Relation zu den sonstigen, meist theoretischen Übungen bewerten. Dabei sollte untersucht werden, wieviel Sie gelernt bzw. tiefergehend verstanden haben. Zudem sollte untersucht werden, ob Sie motiviert waren und Spaß am Arbeiten mit den Quadrocopter-Aufbauten hatten. Da es sehr schwierig ist, dies, d.h. das Gelernte etc., zu bestimmen, wurde in erster Hinsicht mit der Umfrage untersucht, was die Studenten darüber dachten, gelernt etc. zu haben. Dies allein beinhaltet schon eine gewichtige Aussage.

Das Luft- und Raumfahrtlabor bestand nicht jedes Semester nur aus Übungen zum Thema

Quadrokooper. Daher wird in der Umfrage explizit auf die Quadrokooper-Übungen Bezug genommen.

Wenn nicht anders erwähnt, standen die in Tabelle 9.1 aufgelisteten 5 Antwortmöglichkeiten der Spalte „A“ zur Auswahl. Der Zahlenwert in der letzten Spalte dient der Darstellung der Ergebnisse sowie der rechnerischen Auswertung.

**Tabelle 9.1: Antwortmöglichkeiten des Fragebogens und Zahlenwert (Online-Umfrage)**

Antwortmöglichkeiten				Zahlenwert
A	B	C	D	
voll zutreffend	Ja	SS 2011	Weiblich	1
eher zutreffend	Nein	SS 2012	Männlich	2
neutral		SS 2013		3
weniger zutreffend		SS 2014		4
nicht zutreffend				5

Der Fragebogen gliedert sich in die folgenden 3 Blöcke:

### **Block 1: Vergleich Quadrokooper-Übungen mit Übungen anderer Fächer**

- Frage 1: Durch die Übungen der anderen Fächer meines Studiums habe ich im Durchschnitt viel gelernt.
- Frage 2: Durch die Quadrokooper-Übungen habe ich im Durchschnitt viel gelernt.
- Frage 3: In den Übungen in meinem Studium (andere Fächer) konnte ich gelerntes Wissen direkt anwenden und dadurch tiefergehend begreifen.
- Frage 4: In den Quadrokooper-Übungen konnte ich gelerntes Wissen direkt anwenden und dadurch tiefergehend begreifen.
- Frage 5: Die Übungen in meinem Studium (andere Fächer) haben mir Spaß gemacht.
- Frage 6: Die Quadrokooper-Übungen haben mir Spaß gemacht.
- Frage 7: Durch die Übungen in meinem Studium (andere Fächer) wurde ich motiviert zu lernen.
- Frage 8: Durch die Quadrokooper-Übungen wurde ich motiviert zu lernen.
- Frage 9: Die Übungen im Studium (andere Fächer) haben mir im weiteren Verlauf des Studiums geholfen.
- Frage 10: Die Quadrokooper-Übungen haben mir im weiteren Verlauf des Studiums geholfen.
- Frage 11: Bist du bereits berufstätig?
- Frage 12: Das Gelernte aus den Übungen im Studium (andere Fächer) hat mir im Beruf geholfen.
- Frage 13: Das Gelernte aus den Quadrokooper-Übungen hat mir im Beruf geholfen.

Die Antwortmöglichkeiten für Frage 11 waren „Ja“ und „Nein“ (Antwortmöglichkeiten „B“). Die Fragen 12 und 13 wurden nur eingeblendet, wenn Frage 11 mit „Ja“ beantwortet wurde.

### **Block 2: Tiefergehende Fragen zu den Quadrokooper-Übungen**

- Frage 14: Das Programmieren eines Quadrokoopers war für mich eine sinnvolle Übung, um meine Kenntnisse im Bereich Embedded Programming (Mikrocontrollerprogrammierung) und Regelung zu verbessern.
- Frage 15: Das Programmieren eines Quadrokoopers halte ich für eine sinnvolle Hilfe in meiner beruflichen Entwicklung, da es ein spielerisches Erlernen technischer Grundlagen darstellt.
- Frage 16: Hast du schon andere eingebettete Systeme (Mikrocontroller) programmiert, außer Quadrokooper?

Frage 17: Hat dir das Programmieren eines Quadropters mehr Spaß gemacht als das Programmieren anderer eingebetteter Systeme?

Für Frage 16 gab es wiederum nur die beiden Antwortmöglichkeiten „Ja“ und „Nein“ (Antwortmöglichkeiten „B“). Wurde Frage 16 mit „Ja“ beantwortet, so wurde Frage 17 gestellt, sonst nicht.

### Block 3: Allgemeine Fragen

Frage 18: In welchem Semester hast du die Veranstaltung Luft- und Raumfahrtlabor besucht?

Frage 19: Welches Geschlecht hast du?

Frage 20: Hast du Luft- und Raumfahrtlabor bestanden?

Zu Fragen 18 standen die Antwortmöglichkeiten „C“ zur Auswahl. Frage 19 hatte die Antwortmöglichkeiten „D“ und Frage 20 hatte die Antwortmöglichkeiten „B“ (Tabelle 9.1).

### Weitere Hintergründe zur Umfrage

Der Link zum Fragebogen wurde per E-Mail an alle Teilnehmer des Labors der Jahrgänge 2011 bis 2014 verschickt. In diesem Zeitraum haben insgesamt ca. 150. Studenten teilgenommen. Darüber hinaus wurden die Studenten noch über andere Wege wie z.B. Facebook von der Umfrage unterrichtet. Da einige Studenten bereits zum Zeitpunkt der Umfrage nicht mehr an der Universität Würzburg eingeschrieben waren, war es nicht möglich, alle zu erreichen. Trotz dieser Schwierigkeiten haben 61 Personen die Umfrage ausgefüllt. Dies entspricht einer Beteiligung von ca. 41%.

### 9.3.2 Ergebnisse der Umfrage

Die Ergebnisse der Umfrage aus Tabelle 9.2 sind im Anhang (S. 192 - 195) grafisch aufbereitet dargestellt. Es fällt auf, dass nicht alle Fragen von jedem Teilnehmer der Umfrage beantwortet wurden. Für die Auswertung der Umfrage kann dies aber insgesamt vernachlässigt werden.

Tabelle 9.2: Anzahl der gegebenen Antworten zu den Fragen 1 bis 20

	Frage									
Antwort	1	2	3	4	5	6	7	8	9	10
1	10	37	6	30	2	26	6	9	11	18
2	44	18	25	26	17	22	20	29	37	21
3	6	2	20	3	30	10	23	17	12	13
4	1	3	8	1	10	2	11	5	1	8
5	0	0	0	0	1	1	1	1	0	1
Total	61	60	59	60	60	61	61	61	61	61
	Frage									
Antwort	11	12	13	14	15	16	17	18	19	20
1	11	2	4	48	37	43	13	17	4	56
2	47	6	2	11	21	18	15	12	53	1
3	0	1	3	2	3	0	13	17	0	0
4	0	1	1	0	0	0	1	14	0	0
5	0	0	0	0	0	0	0	0	0	0
Total	58	10	10	61	61	61	42	60	57	57

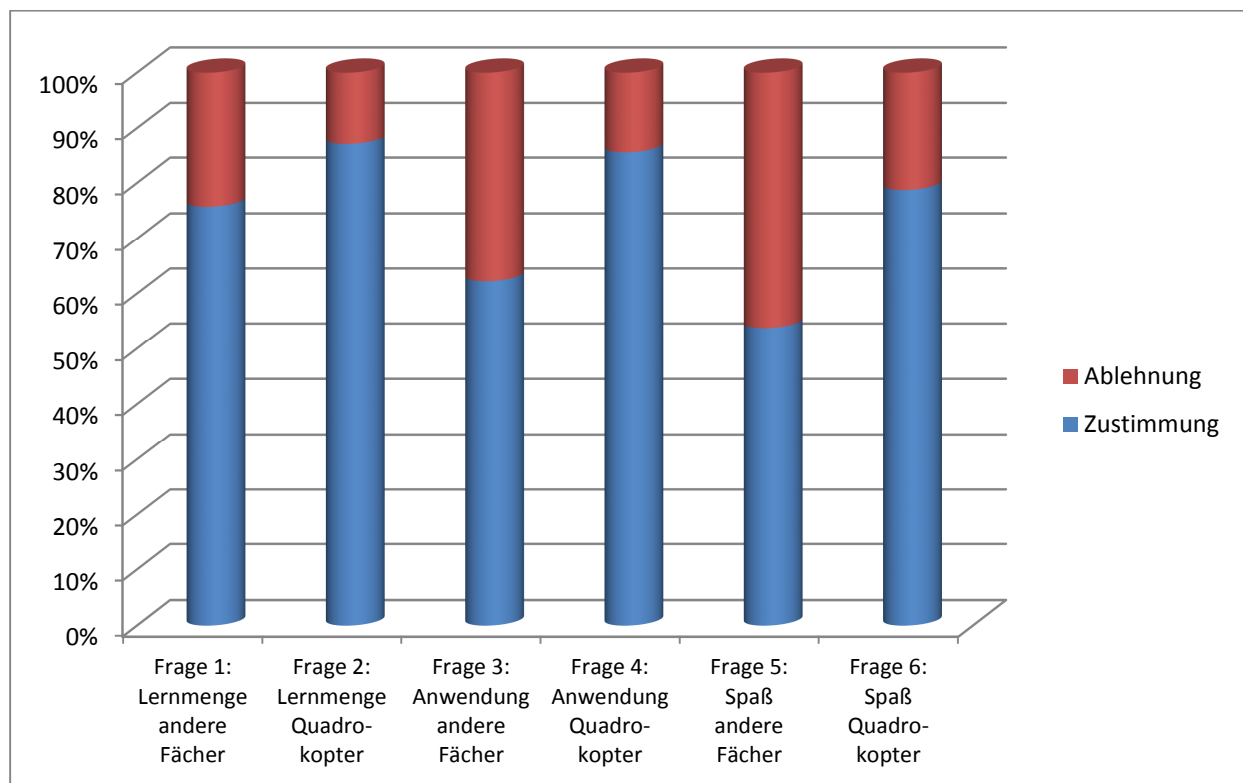
### 9.3.3 Auswertung der Umfrage

In Tabelle 9.3 ist der rechnerische Schnitt als Mittelwert der zu den Antworten korrespondierenden Zahlenwerte ausgerechnet. Je niedriger der Wert, desto größer ist die mittlere Zustimmung.

**Tabelle 9.3: Numerischer Schnitt aus dem Zahlenwert der gegebenen Antworten zu den Fragen 1 bis 20**

Frage:	1	2	3	4	5	6	7	8	9	10
Schnitt:	1,97	1,52	2,51	1,58	2,85	1,85	2,69	2,34	2,05	2,23
Frage:	11	12	13	14	15	16	17	18	19	20
Schnitt:	1,81	2,10	2,10	1,25	1,44	1,30	2,05	2,47	1,93	1,02

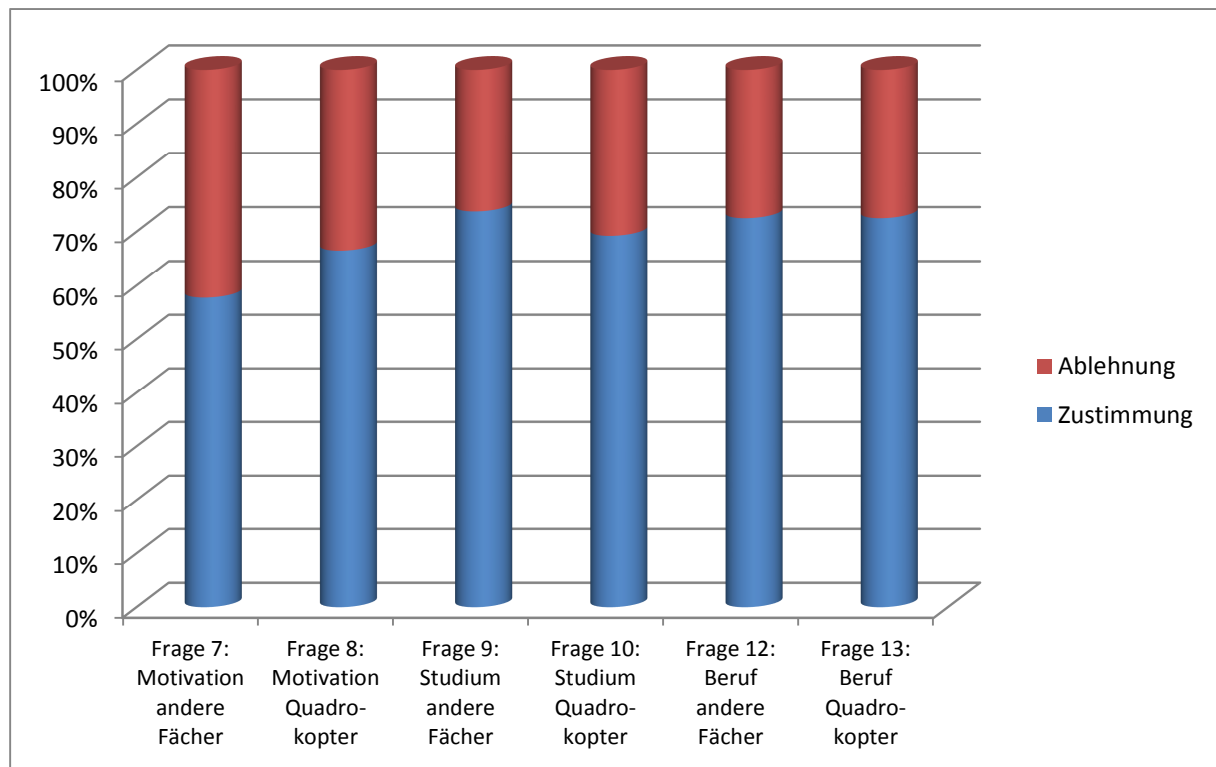
Aus den gewonnenen Daten lässt sich eine relative Zustimmung aller Befragten ermitteln. Dazu wird der Schnitt in Relation zum theoretischen Maximum (Zustimmung aller) und Minimum (Verneinung aller) gesetzt. Diese relative Zustimmung ist für die Fragen zum Vergleich von Quadropter-Übungen mit den anderen Fächern durch Abbildung 9.1 und Abbildung 9.2 grafisch aufbereitet dargestellt: Dabei entsprechen der blaue Balken der relativen Zustimmung und der rote Balken der relativen Ablehnung.



**Abbildung 9.1: Relative Zustimmung zu den vergleichenden Fragen 1-6 Themen: Lernmenge (1,2), Anwendung (3,4) und Spaß (5,6)**

Den Daten ist zu entnehmen, dass die Studenten aussagen, in den Quadropter-Übungen mehr gelernt zu haben (1,2). Gleichzeitig sagen die Studenten deutlich aus, dass Sie in den Quadropter-Übungen besser als im Durchschnitt der anderen Fächer gelerntes Wissen anwenden und tiefergehend begreifen konnten (3, 4). Vor allem aber hatten Sie im Durchschnitt deutlich mehr Spaß

in den Quadrokopter-Übungen als in den Übungen anderer Fächer (5,6).



**Abbildung 9.2: Relative Zustimmung zu den vergleichenden Fragen 7-10, 12 und 13**  
Themen: Motivation (7,8), Studium (9,10) und Beruf (12,13)

In den Quadrokopter-Übungen war die Motivation zu lernen etwas höher als in den anderen Fächern (7,8). Für das weitere Studium haben die Quadrokopter-Übungen nach Aussage der Studenten etwas weniger geholfen als die Übungen anderer Fächer (9,10). Laut der Aussage einiger Umfrageteilnehmer liegt dies an den weiteren, meist theoretischen Inhalten des Studiums, die eine geringe Schnittmenge mit den Quadrokopter-Übungen haben. Beim Nutzen im Beruf hält es sich die Waage (12,13), wobei die anderen Fächer einen wesentlich größeren Umfang des Studiums ausmachen. Diese Frage besitzt aber wegen der lediglich 10 Antworten eine geringe Aussagekraft.

Bei der Auswertung der tiefergehenden Fragen zeigt sich ein noch klareres Bild. In Abbildung 9.3 ist die relative Zustimmung zu den Fragen 14, 15 und 17 dargestellt. Allen drei Aussagen wurde mit großer Mehrheit zugestimmt. Damit wurden die Thesen aus Frage 14, 15 und 17 klar bejaht.

An der Umfrage haben 53 männliche und 4 weibliche Studenten teilgenommen. Dieses Verhältnis könnte repräsentativ für die Laborübungen sein. 56 Teilnehmer der Umfrage gaben an, das Labor bestanden zu haben. Einer gab an, es nicht bestanden zu haben. Dies ist plausibel, ebenso wie die gleichmäßige Verteilung der Jahrgänge. Es gaben 17 Teilnehmer an, im SS 2011 die Laborübungen besucht zu haben. Analog gaben 12, 17 und 14 Teilnehmer an, im SS 2012, SS 2013 und SS 2014 die Laborübungen absolviert zu haben. Die Daten ermöglichen eine weitere Auswertung der Umfrage, auf die an dieser Stelle aufgrund der Klarheit der Ergebnisse verzichtet wird.

Zusammenfassend bewertet die Umfrage die Quadrokopter-Übungen sehr positiv. Die Evaluierung hat somit ihr Ziel erreicht, den besonderen Wert der Quadrokopter-Übungen für die Lehre mit

empirischen Daten zu belegen.

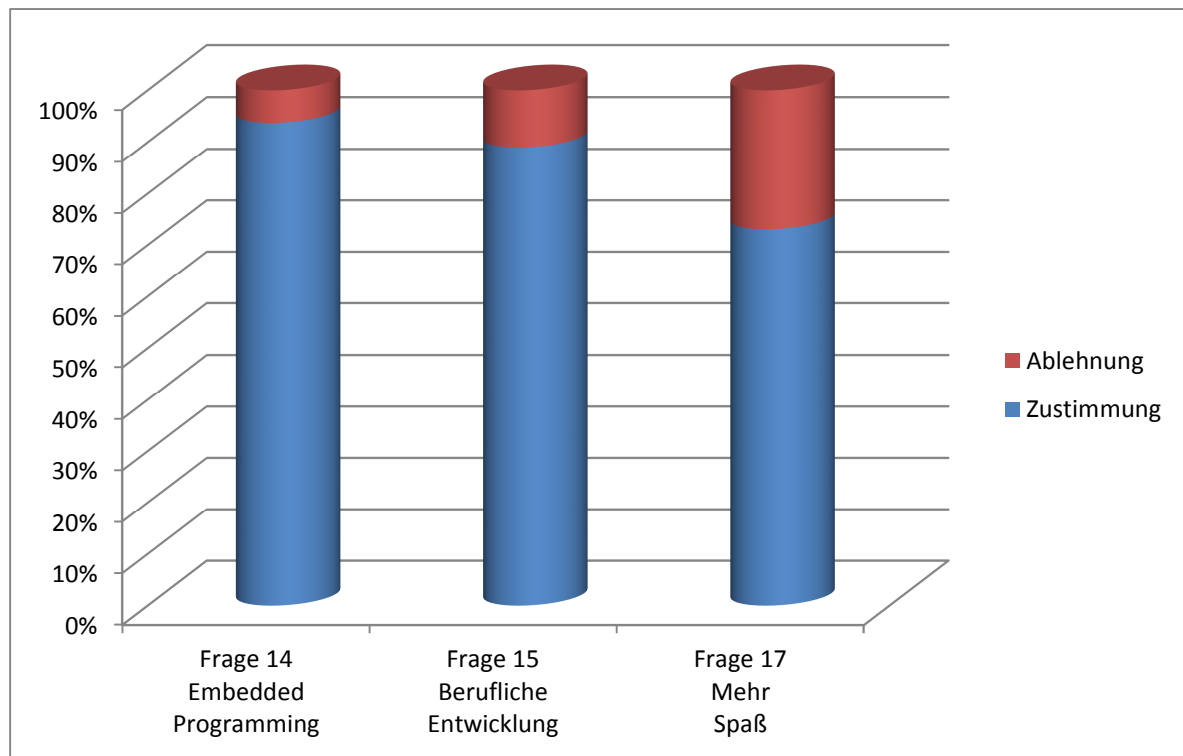


Abbildung 9.3: Auswertung der tiefergehenden Fragen 14, 15 und 17

Abschließend sei an dieser Stelle noch eine Aussage eines Teilnehmers der Summer School 2015 nach den Quadropter-Übungen wiedergegeben: „I am in my 1st year master studies and this was the best exercise I ever had!“







*The future belongs to those who believe in the beauty of their dreams.*  
[Eleanor Roosevelt]

## 10 Diskussion und Ausblick



Grafik: Jan Balzer

## 10.1 Diskussion

Die Evaluierung und Bewertung des Systems und seiner vielfältigen Aspekte zeigt, dass die gemachte Anforderung an die Positionsgenauigkeit (ca. 1m) unter den gewünschten Kosten (ca. 3.000 €) erfüllt wurde. Ebenfalls wurde die Anforderung an die Genauigkeit der Lokalisierung von Objekten (ca. 50cm) erfüllt. Damit sind die quantifizierten Anforderungen erfüllt.

Alle weiteren qualitativen Anforderungen nach Einfachheit, Verlässlichkeit, Modularität, Erweiterbarkeit und Flexibilität konnten durch das Konzept sowie dessen Realisierung in Hard- und Software nachgewiesen werden. Nicht zuletzt sei die Forderung nach Autonomie zu nennen, die durch die Evaluierung nachgewiesen werden konnte. Dabei ergaben sich einige Vor- und Nachteile sowie Verbesserungsmöglichkeiten des Systems und seiner Subsysteme, auf die im Weiteren eingegangen wird.

Die Evaluierung der Orientierungsbestimmung ergab, dass die Orientierung über eine lange Zeit ohne große Fehler bestimmt werden kann, aber nicht völlig fehler- und rauschfrei ist. Die Fehler sind im unteren einstelligen Gradbereich anzusiedeln. Diese geringen Fehler bleiben aber gerade für ein autonomes System nicht ohne Auswirkungen, da Sie eine Positionsänderung bewirken und damit Probleme für die Positionsregelung und Kollisionsvermeidung bedeuten. Die verwendete Datenfusion hat in der Evaluierung ihre Fähigkeit der Fehlerkompensation bewiesen, ihre Schwäche bleibt aber das Rauschverhalten und die fehlende dynamische Bias-Korrektur. Das Rauschverhalten ist zudem abhängig von Vibrationen, die wiederum von der Sensoraufhängung, den Beinen, Propellern und Lagern der Motoren stammen können. Hier besteht zusätzliches Optimierungspotential, das gleichzeitig eine Kostensteigerung nach sich ziehen kann. Dagegen besteht der Vorteil der entwickelten Orientierungsbestimmung in der Einfachheit und der Flexibilität, womit die Lösung optional Accelerometer und Magnetometer verwenden kann. Damit können gerade Verschlechterungen der Datenfusion durch Störungen im Magnetfeld kategorisch vermieden werden. Zudem wirken sich magnetische Störungen stets nur auf den weniger kritischen Yaw-Winkel aus, was ein großer Vorteil sein kann.

Die Experimente zur Höhenbestimmung zeigen, dass die Höhe selbst für den Fall, dass zwei von drei Hauptsensoren falsche Werte lieferten, korrekt bestimmt werden kann. Der Fehler der Höhenbestimmung hängt von der Höhe ab und ist für die hier betrachteten üblichen Flughöhen in Innenräumen (bis ca. 2m) vernachlässigbar. Da die Lösung mittels WF keine Rauschunterdrückung bewirkt, kann die Höhenbestimmung mit Hilfe eines einfachen KF verbessert werden. Ob die dazu notwendige Rechenzeit zur Verfügung steht, wäre noch zu untersuchen. Durch Verwendung eines teureren Lasersensors anstatt der IR-Sensoren kann die Höhenbestimmung in Hinblick auf Genauigkeit und Messbereich zusätzlich verbessert werden.

Die Versuche zur Lage- und Höhenregelung zeigten wenig Handlungsbedarf. PID-Regler zur Lageregelung sind eine weit verbreitete Lösung und haben sich 1000-fach bewährt. Ein Nachteil der Lageregelung ist jedoch die RPY-Darstellung. Durch einen Quaternionen-Regler könnte eine weitere Verbesserung erreicht werden.

Die Evaluierung der Kollisionsvermeidung ergab, dass der AQopter18 in der Lage ist Hindernissen wie Wänden und Personen auszuweichen. Dabei kann der Quadrocopter den Abstand zu Hindernissen regeln und damit auch in engen Tunneln (Breite 1,8m) fliegen. Der evaluierte Ansatz unter Ver-

wendung von IR und US Sensorik kann mit Hilfe der übrigen OD Sensorik weiter verbessert werden, um auch das Erkennen von Fenstern, Türen und kleinen Objekten zu ermöglichen.

Die innovativen Arbeiten und Resultate zum autonomen Flug, der Objektsuche und dem gerichteten Flug zeigten neue Lösungswege und Zusammenhänge für technische Fragen und Probleme auf. Schließlich wurde das QCS mit Hilfe eines online Fragebogens didaktisch analysiert und es offenbarte sich dessen Wert für die Lehre.

## 10.2 Ausblick

Zum Ausblick dieser ca. 5 jährigen Arbeit gehört eine ganze Reihe an Aspekten, wobei hier nur auf das Wesentliche eingegangen wird. Diese wesentlichen Ausblicke sind der Förderpreis, die geplante Firmengründung mit Produktentwicklung sowie die zukünftige Verwendung zum Zwecke der Lehre und Forschung.

Die Aktivitäten im Rahmen des Projektes „Rettungshelfer mit Propellern“, das 2014 mit dem Förderpreis der Universität und der Mainfränkischen Wirtschaft ausgezeichnet wurde, sind noch nicht abgeschlossen (S. 188, Abbildung 10.3). Zu diesem Teilprojekt des AQopter18 gehört die Entwicklung und Integration eines Manipulators zur Verwendung im Feuerwehreinsatz. Wie in der Arbeit bereits beschrieben, wurde der Manipulator aufgebaut, geflogen und rudimentär programmiert. Im manuellen Flug konnte bereits ein Tennisball in einen Papierkorb versenkt werden. Zum Greifen von Objekten im Flug und für den autonomen Betrieb ist das System noch weiterzuentwickeln. Die Arbeiten daran werden in den nächsten Wochen und Monaten weiter geführt.

Die innovativen Ergebnisse der Arbeit zeigten, dass für die gefundenen Lösungen eine wirtschaftliche Nachfrage existiert. Daher sind im Anschluss an die Arbeit eine Firmengründung sowie eine Produktentwicklung vorgesehen. Dabei stehen drei Produkte als Ergebnisse dieser Arbeit auf der Agenda.

Das erste Produkt ist der bereits genannte Quadrocopter mit Manipulator als Ergebnis des Förderpreises. Das zweite Produkt ist der Kollisionsvermeidungsassistent, der als Modullösung in kommerzielle Quadrocopter und Community-Projekten integriert werden kann (Abbildung 10.2, S. 188). Hier bestehen Kontakte zu Firmen, die bereits Interesse an einer solchen Lösung gezeigt haben. Eine Kooperation mit einer Firma als Pilotprojekt ist ebenfalls auf dem Wege.

Das dritte Produkt ist das QCS (Abbildung 10.1, S. 188). Dieses wurde bereits an die FH Brandenburg und die FH Würzburg-Schweinfurt verkauft und wird dort eingesetzt. Darüber hinaus wird ein System von der TU Hamburg im Rahmen einer Kooperation verwendet. An der Universität Würzburg kommt das QCS im Luft- und Raumfahrtlabor (Studiengang: Luft- und Raumfahrtinformatik) sowie im Quadrotor Labor der Summer School Aerospace Information Technology zum Einsatz.

Eine Weiterentwicklung als Messdrohne für Aquakulturen liegt ebenso im Fokus zukünftiger Aktivitäten wie die Erkundung von Bergwerken, großen Schornsteinen und anderen Tunnelsystemen. Schließlich werden das System, seine Komponenten sowie die damit gewonnenen Erkenntnisse, Erfahrungen und Ergebnisse noch in weiteren und zukünftigen Tätigkeiten des Lehrstuhls verwendet. Dazu gehören Promotionen und weiterführende Projekte wie das Robex-Projekt [128] zur

Erforschung robotischer Systeme in extremen Umweltbedingungen (Abbildung 10.4).

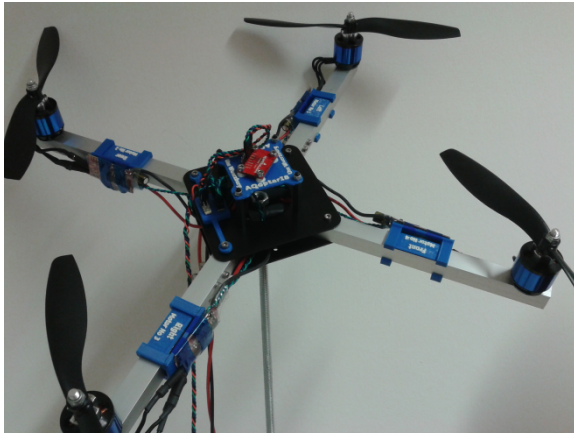


Abbildung 10.1: QCS

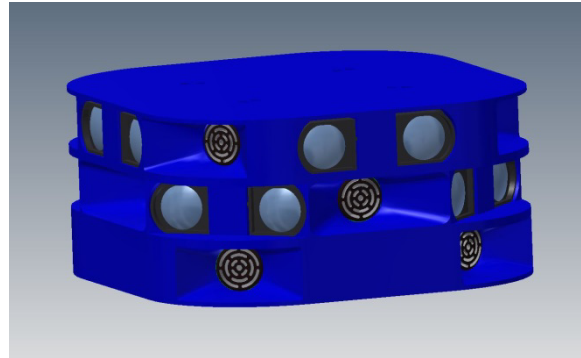


Abbildung 10.2: Kollisionsvermeidungsassistent  
(CAD Zeichnung Sascha Dechend)



Abbildung 10.3: Rettungshelfer mit Propellern  
(Grafik Jan Balzer)



Abbildung 10.4: Erster Testaufbau mit ausgewählter  
AQopter18-Sensorik in der Arktis (Juli 2014, Foto: AWI)

# 11 Anhang

## 11.1 Weitere Graphen zur Evaluierung des Systems

### 11.1.1 Resultate zur Orientierungsbestimmung (Stationärer Flug, S. 144)

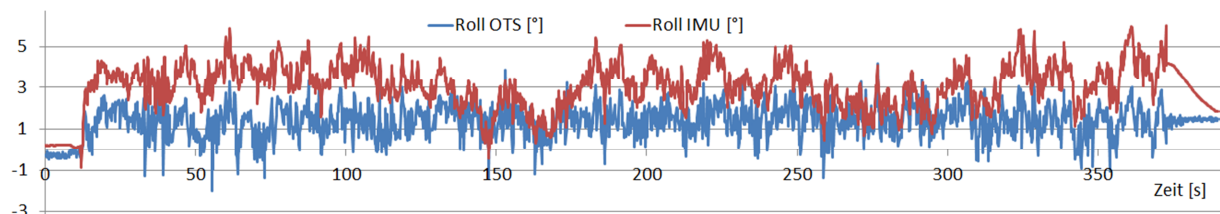


Abbildung 11.1: Zeitlicher Verlauf des Roll-Winkels (stationärer Flug, Dauer ca. 6 Minuten)

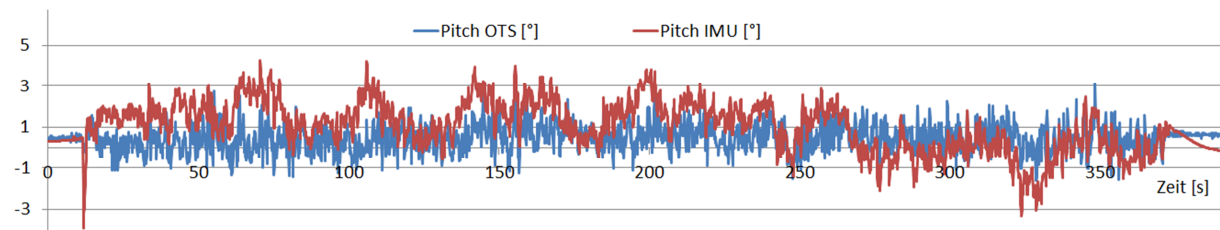


Abbildung 11.2: Zeitlicher Verlauf des Pitch-Winkels (stationärer Flug, Dauer ca. 6 Minuten)

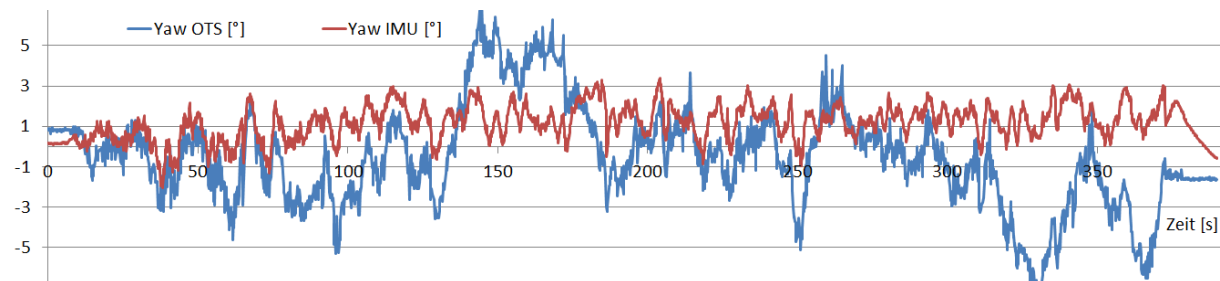


Abbildung 11.3: Zeitlicher Verlauf des Yaw-Winkels (stationärer Flug, Dauer ca. 6 Minuten)

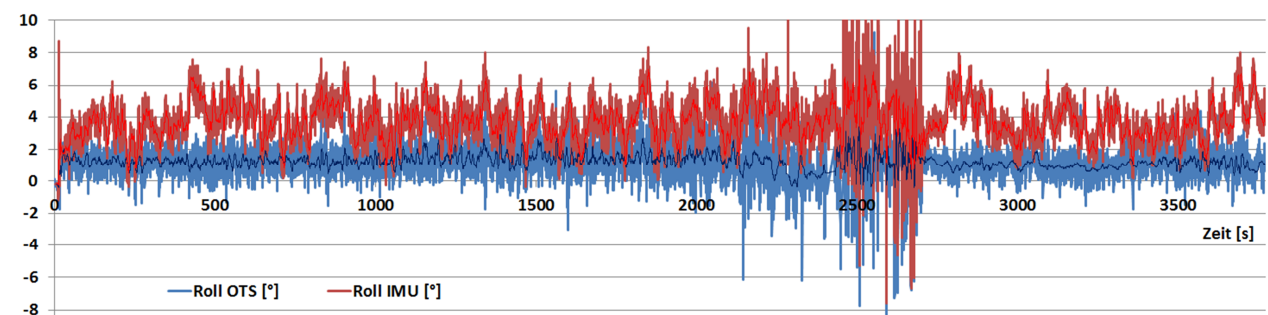


Abbildung 11.4: Zeitlicher Verlauf des Roll-Winkels (stationärer Flug, Dauer ca. 1 Stunde)

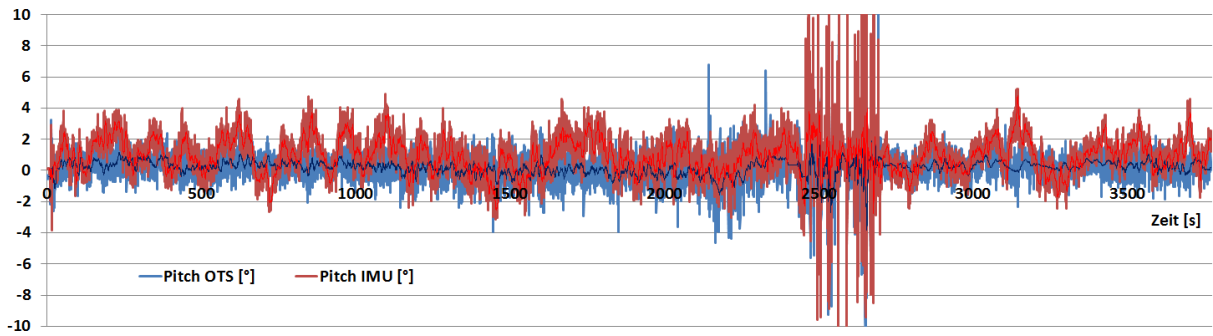


Abbildung 11.5: Zeitlicher Verlauf des Pitch-Winkels (stationärer Flug, Dauer ca. 1 Stunde)

Anmerkung zu Abbildung 11.4 und Abbildung 11.5: Zum Zeitpunkt 2500s wurde in die Steuerung eingegriffen. Daher rühren die Ausschläge, die vom OTS nicht detektiert wurden, weil der Quadrocopter sich außerhalb des Sichtfeldes befand, da die mittels OF Sensor bestimmte Position gedriftet ist. Der Quadrocopter wurde dann manuell zurück ins Zentrum geflogen und die Position per Hand zurückgesetzt.

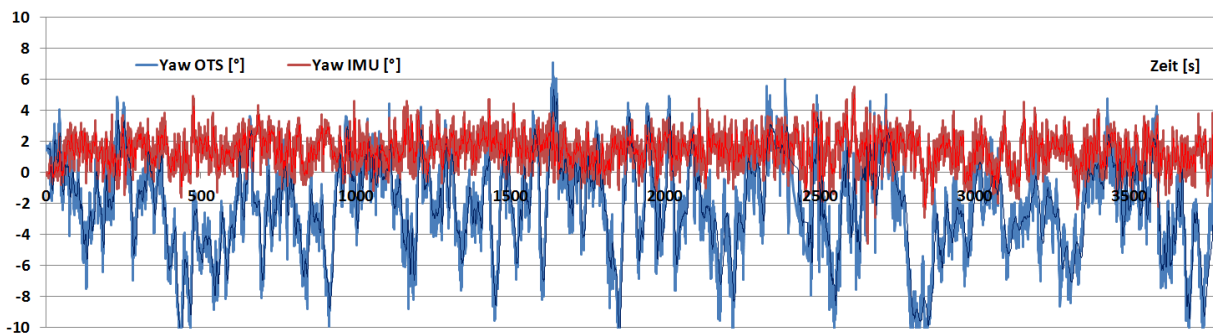


Abbildung 11.6: Zeitlicher Verlauf des Yaw-Winkels (stationärer Flug, Dauer ca. 1 Stunde)

### 11.1.2 Resultate zur Orientierungsbestimmung (Stationärer Flug mit Fehlereinspeisung, S. 146)

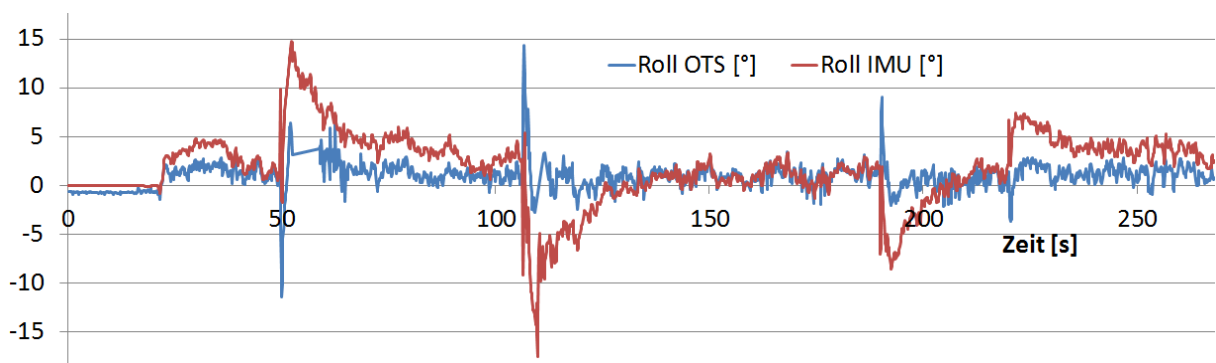


Abbildung 11.7: Zeitlicher Verlauf des Roll-Winkels (stationärer Flug, Fehlereinspeisung)

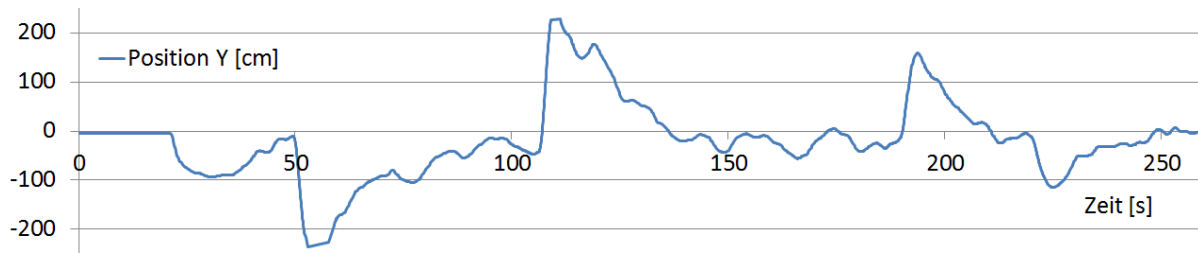


Abbildung 11.8: Zeitlicher Verlauf der y-Position mit OTS gemessen (stationärer Flug, Fehlereinspeisung)

### 11.1.3 Resultate zur Lageregelung (S. 149)

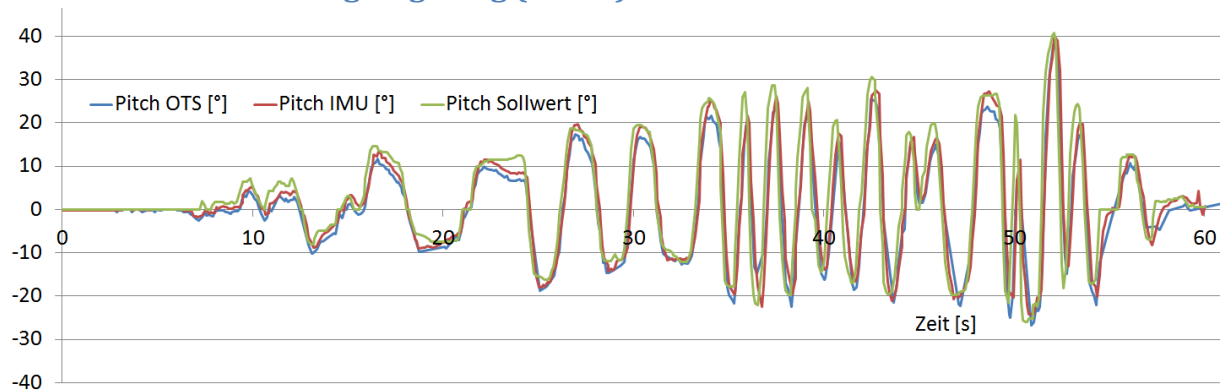


Abbildung 11.9: Zeitlicher Verlauf des Pitch-Winkels (dynamischer Flug)

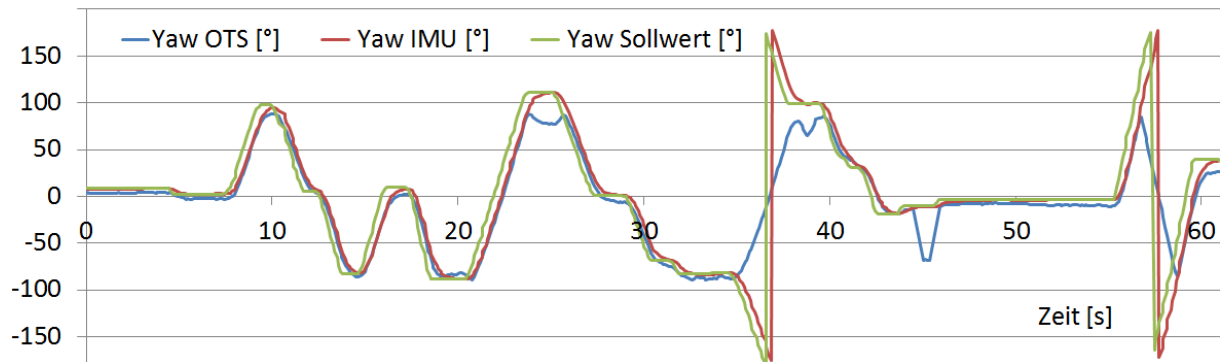


Abbildung 11.10: Zeitlicher Verlauf des Yaw-Winkels (dynamischer Flug)

### 11.1.4 Weitere Ergebnisse zur Umfrage (S. 179)

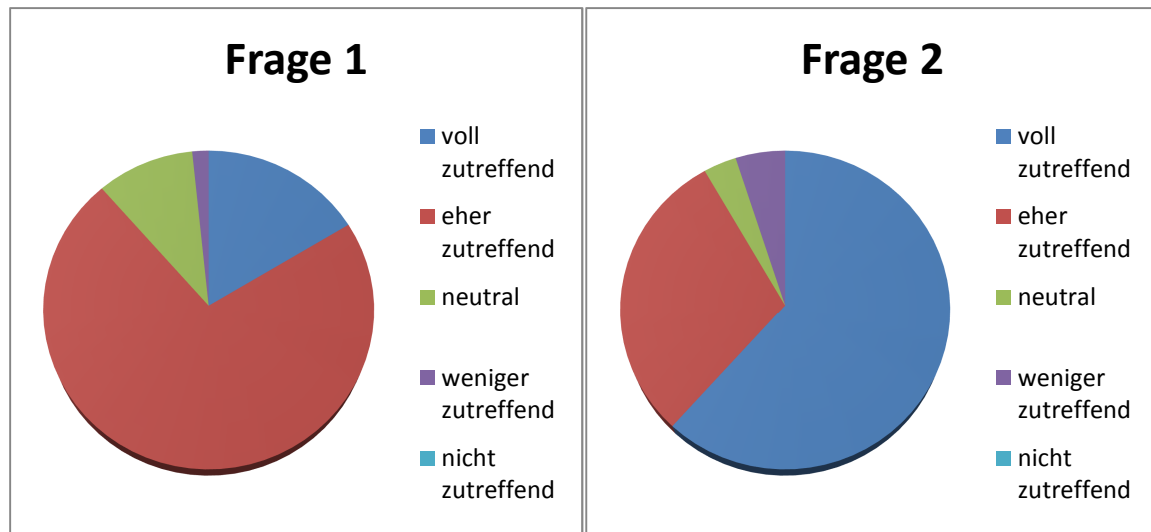


Abbildung 11.11: Verteilung der Antworten auf Frage 1 (andere) und 2 (Quadropter-Übungen) zur Lernmenge

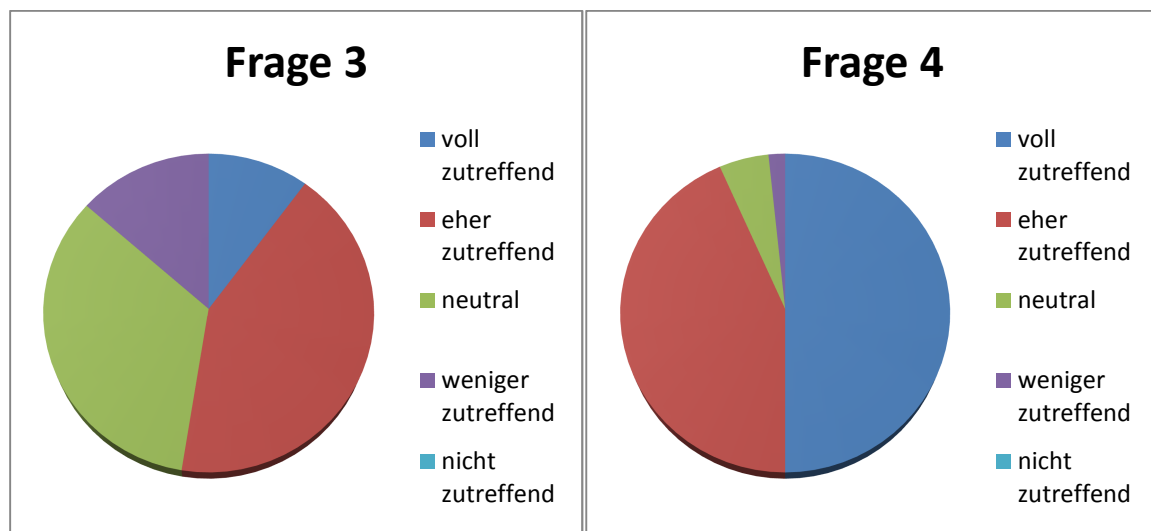


Abbildung 11.12: Verteilung der Antworten auf Frage 3 (andere) und 4 (Quadropter-Übungen) zur Anwendung des gelernten Wissens



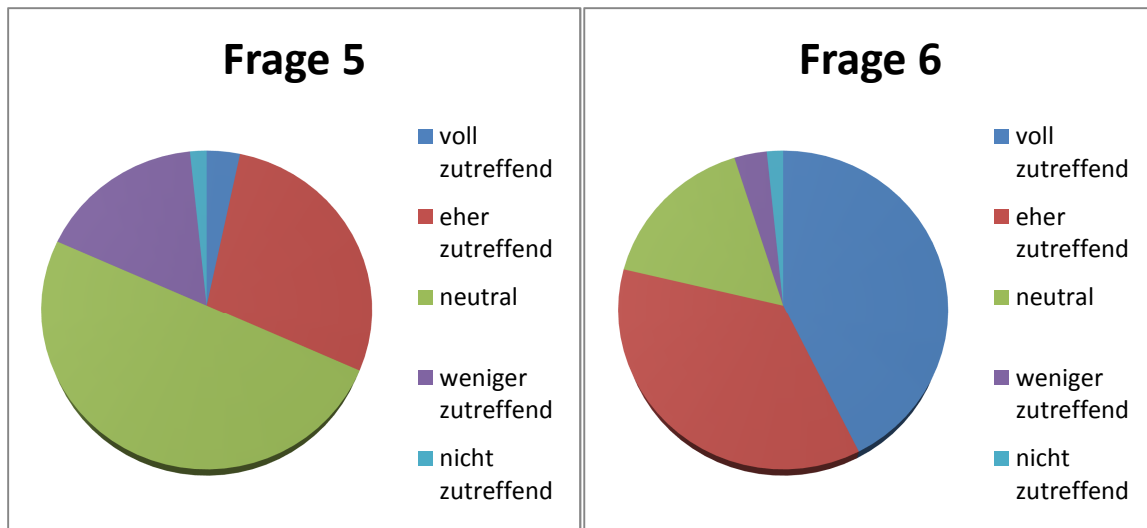


Abbildung 11.13: Verteilung der Antworten auf Frage 5 (andere) und 6 (Quadropter-Übungen) zum Spaß

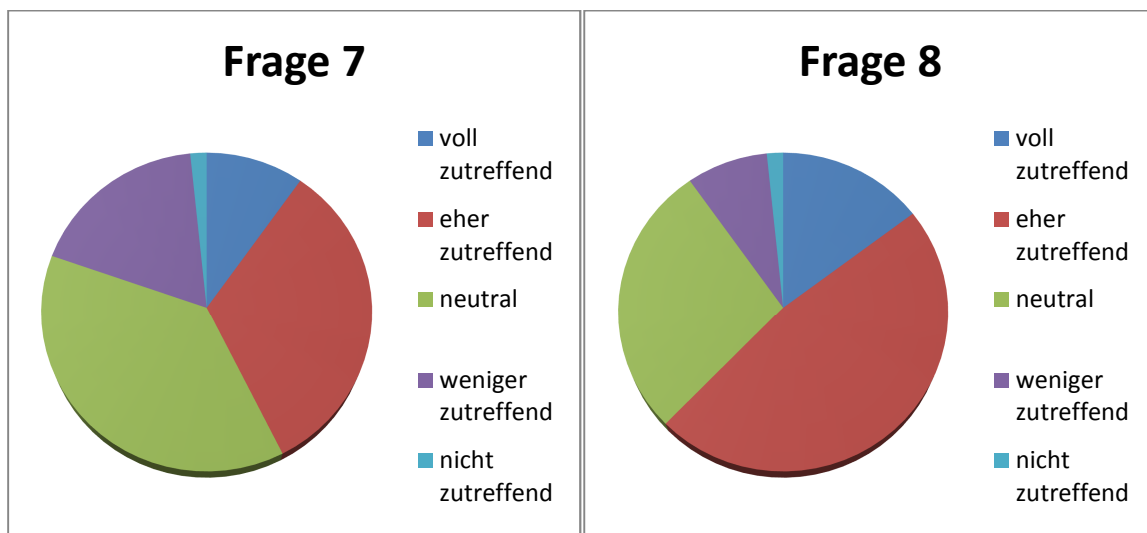


Abbildung 11.14: Verteilung der Antworten auf Frage 7 (andere) und 8 (Quadropter-Übungen) zur Lernmotivation

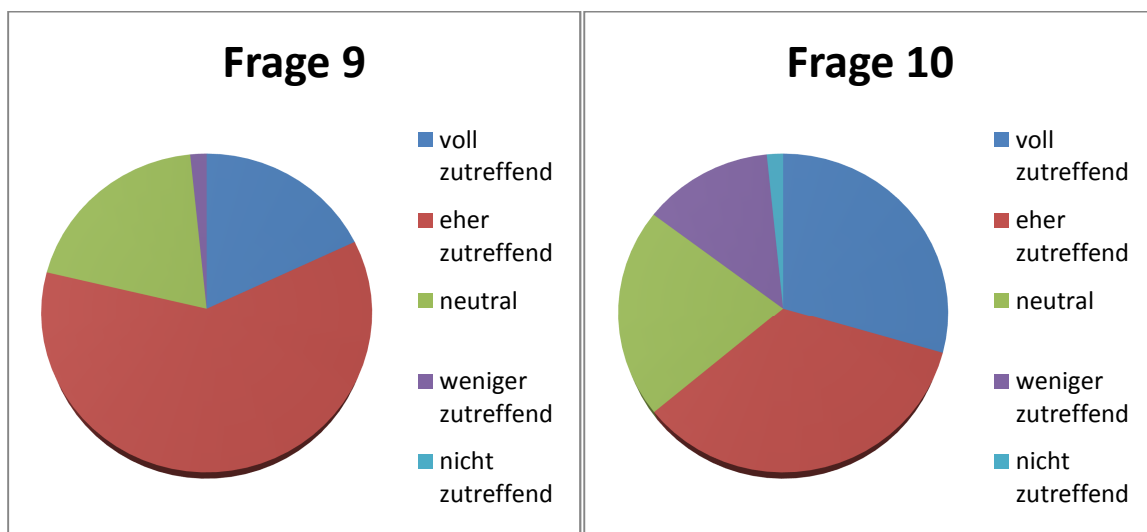


Abbildung 11.15: Verteilung der Antworten auf Frage 9 (andere) und 10 (Quadropter-Übungen) zur Hilfe im Studium

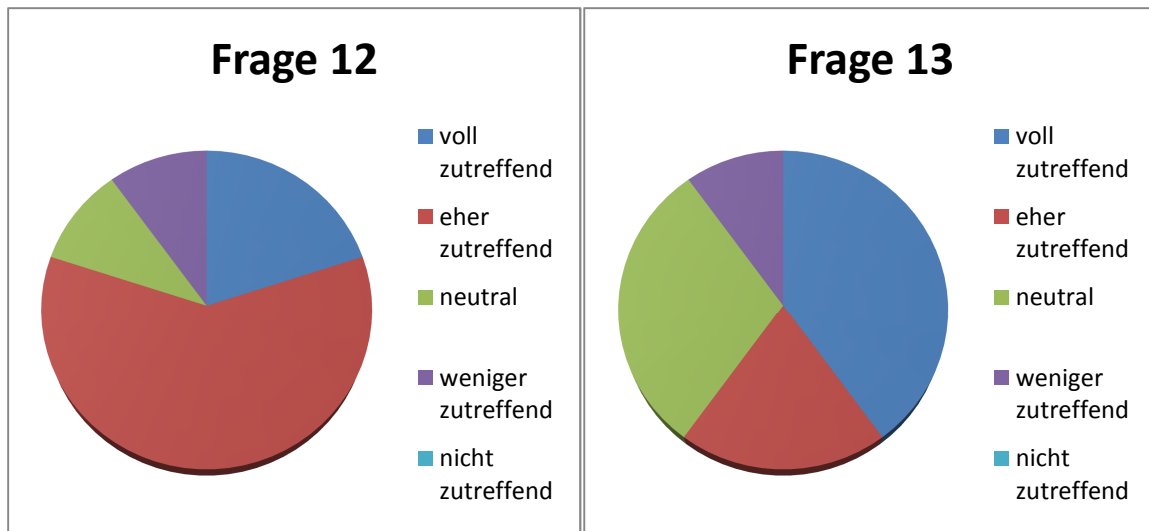


Abbildung 11.16: Verteilung der Antworten auf Frage 12 (andere) und 13 (Quadropter-Übungen) zur Hilfe im Beruf

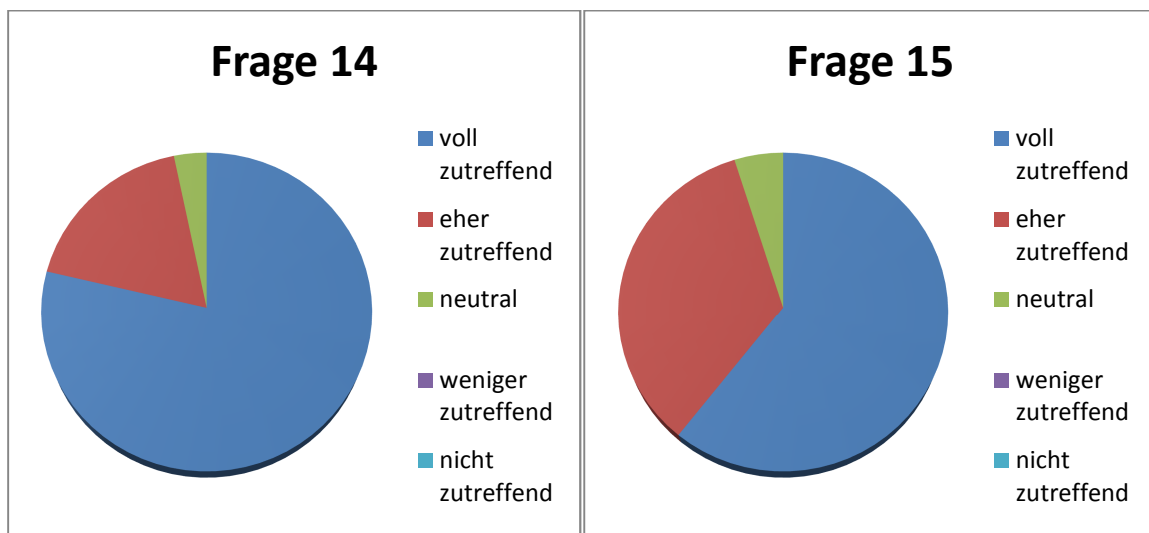


Abbildung 11.17: Verteilung der Antworten auf Frage 14 (Embedded) und 15 (Beruf)

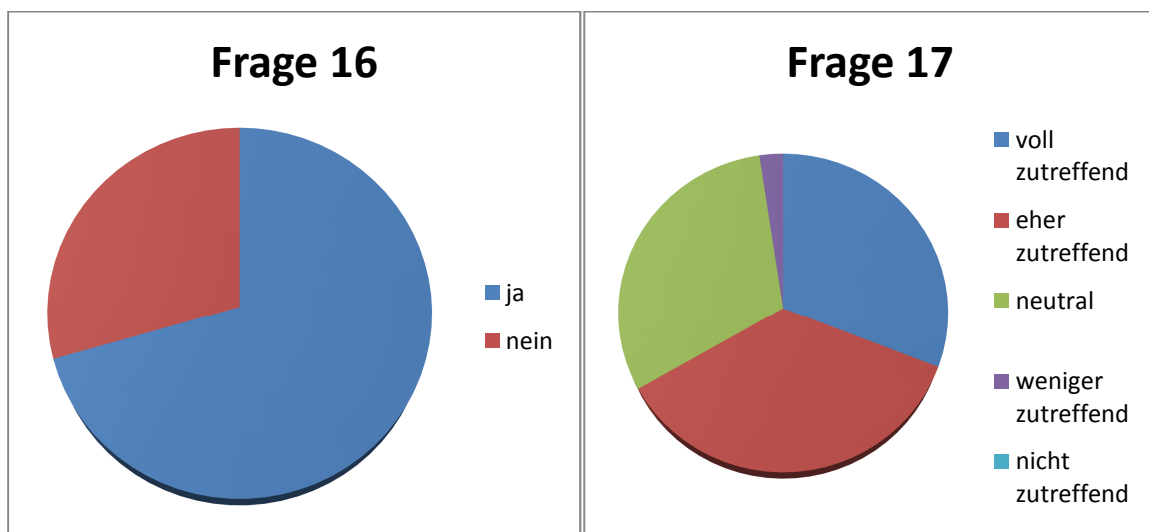


Abbildung 11.18: Verteilung der Antworten zu Frage 16 und 17 (Spaß Mikrocontroller)

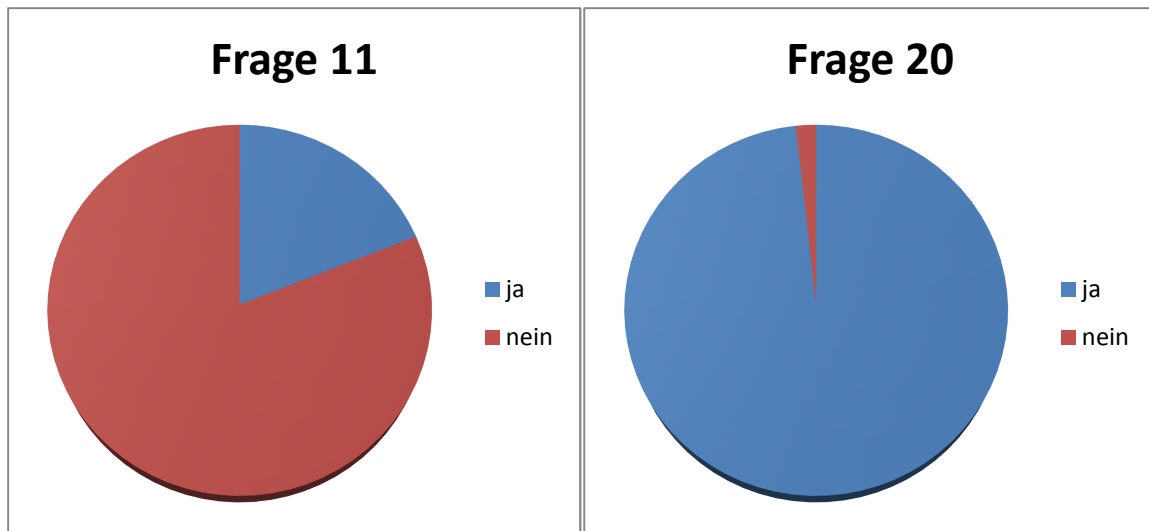


Abbildung 11.19: Verteilung der Antworten zu Frage 11 (Berufstätigkeit) und 20 (Bestanden)

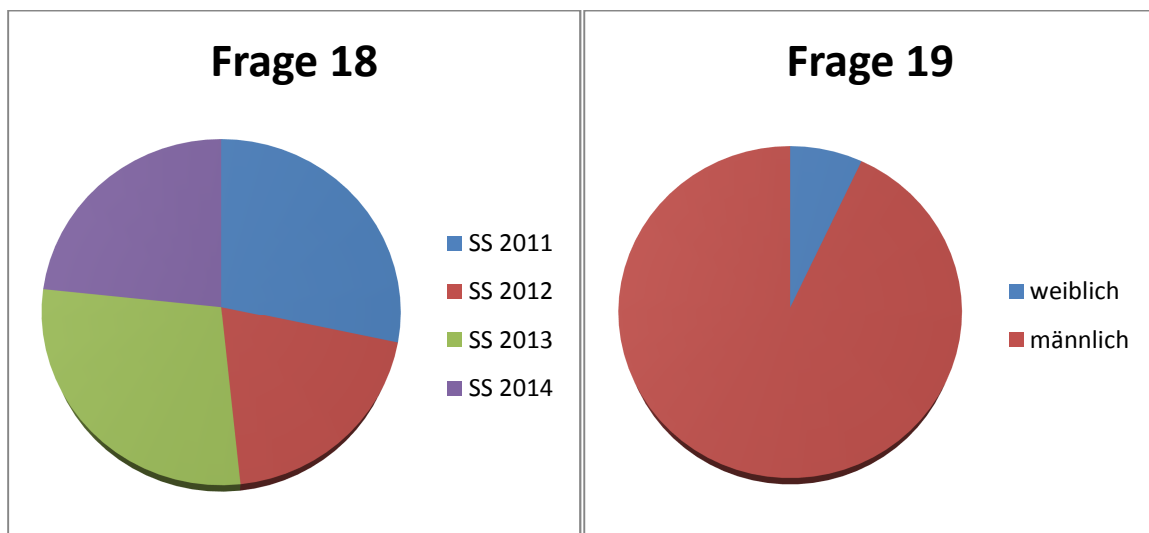


Abbildung 11.20: Verteilung der Antworten zu Frage 18 (Semester) und 19 (Geschlecht)

## 11.2 Abkürzungsverzeichnis

Die Nomenklatur befindet sich auf Seite 16.

## 11.3 Zitate und Kapitelbilder

Zur kreativen Unterhaltung der Arbeit werden alle Kapitel eingeführt mit einem historischen Zitat und einem Bild, deren Zusammenhang sowie deren Aufklärung und Interpretation dem Leser überlassen ist. Zum allgemeinen Verständnis trägt folgender Hinweis zur Abbildung auf Seite 7 bei:

Kernaspekt jeder Einleitung ist die Motivation. Das Zitat kann interpretiert werden als die fundamentale Motivation der Menschheit und jedes Menschen: Die Wiederkehr ins Paradies, den Garten Eden. Das Bild passend dazu zeigt einen Quadrokopter, der nach einem Apfel aus dem Garten Eden greift, wie der Wissenschaftler nach neuen Erkenntnissen strebt. In diesem Sinne viel Freude

beim Interpretieren der Bilder und Zitate.

## 11.4 Historie des AQopterI8-Projekts

Das folgende Kapitel gibt eine chronologische Beschreibung einiger ausgewählter Meilensteine zur Forschung und Entwicklung des AQopterI8-Systems von seinen Anfängen bis zum heutigen Stand. Dabei ist eine genaue Datierung und Abfolge der Entwicklungen schwierig, da oftmals an vielen Baustellen simultan gearbeitet wurde. Die Reihenfolge geht somit nach Abschluss der Hauptarbeiten.

### 11.4.1 Erste Versuche und Jungfernflug

Um Schäden an Mensch und Maschine zu vermeiden, wurde die Software für den Quadrocopter zunächst ausgiebig an einem QCS entwickelt und optimiert. Im Sommer 2011 war dann der Jungfernflug des ersten selbstentwickelten Quadrocopters des Lehrstuhls Informationstechnik für Luft- und Raumfahrt (Abbildung 11.1). Der erste Flug war noch sehr „holprig“, denn der Regler war alles andere als optimal eingestellt. Aber er erfüllte seine Pflicht und - aus Sicherheitsgründen zunächst noch an einer Kordel befestigt - flog. Im Anschluss wurde gegrillt.



Abbildung 11.21: Junger QCS, Jungfernflug und Junggesellengrillen (2011)

### 11.4.2 Kollisionsvermeidung mit Ultraschall

Einige Monate später flog der Quadrocopter schon sehr viel besser. Er konnte nun seine Höhe mittels Ultraschallsensoren regeln. Dies war auch dringend nötig für die Arbeiten zur Kollisionsvermeidung mittels Ultraschall. An diesem Thema forschte und entwickelte Thilo Müller im Rahmen seiner Diplomarbeit, indem er seitlich Ultraschallsensoren anbrachte (Abbildung 11.22). Die Arbeiten daran waren Ende des Jahres fertig. Damit konnte der Quadrocopter erstmals autonom fliegen und Kollisionen mit Hindernissen vermeiden. Personen konnten mit Ultraschall jedoch nicht zuverlässig erkannt werden.



Abbildung 11.22: Flug mit AQopterI8 in High School [Foto: MainPost] Stand 12/2011

### 11.4.3 Höhenregelung mit Sensorfusion

Probleme der Höhenregelung unter alleiniger Verwendung von Ultraschall konnten durch die Verwendung von weiteren Sensoren wie Infrarot und Luftdruck gelöst werden. Julian Rothe leistete im Rahmen seiner Bachelorarbeit einen Beitrag dazu. Der Quadrocopter konnte nun autonom starten, landen und unterschiedliche Höhen anfliegen. Ein Treppenflug wurde ebenfalls möglich. Diese Entwicklungen waren im Frühsommer 2012 abgeschlossen. In Abbildung 11.23 ist ein Foto des damaligen Quadrocopters sowie der Bodenplatte zu sehen.

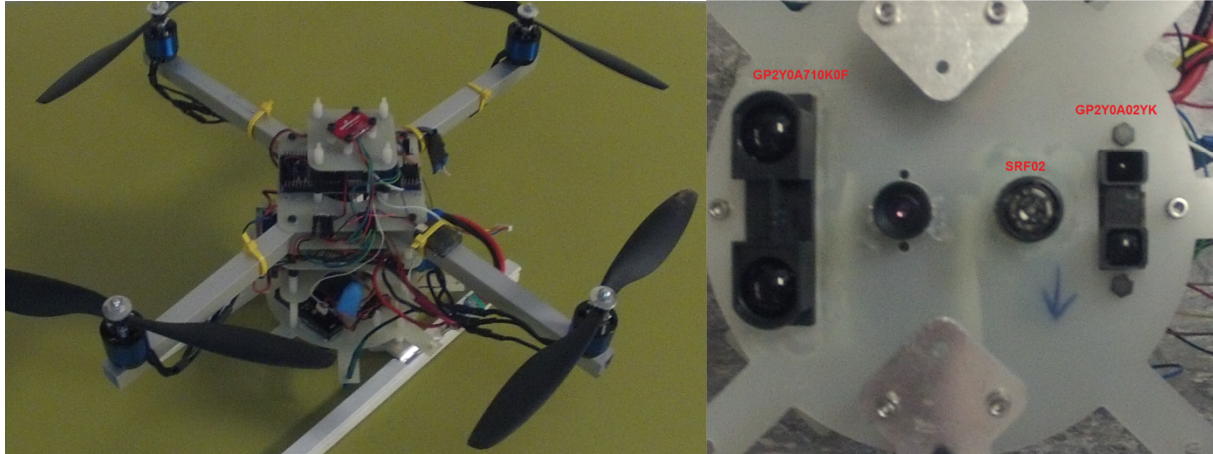


Abbildung 11.23: AQopter18 Stand 7/2012 (links) und dazugehörige Bodenplatte (rechts)

### 11.4.4 Autonomer Flug mit optischem Flusssensor

Mit den bisherigen Ansätzen konnte die Position des Quadrocopters nur über den Abstand zu Hindernissen gesteuert werden. Das autonome Abfliegen eines größeren, leeren Raums war damit nicht möglich. Eine Möglichkeit zur Positionsbestimmung bietet der optische Fluss. In seiner Bachelorarbeit experimentierte Michael Strohmeier mit Lösungen zur Positionsbestimmung mittels optischen Flusses und integrierte den ADNS-3080 in den AQopter18 (Abbildung 11.24). Damit konnte eine Positionsregelung implementiert werden, die das Halten und Anfliegen von Positionen ermöglichte.



Abbildung 11.24: AQopter18 Stand 9/2012 [Foto aus A04]

### 11.4.5 Kardanische Aufhängung

Kippbewegungen des Quadropters führten zu Fehlern der Sensoren für Hinderniserkennung, Höhenbestimmung und Positionsbestimmung (OF). Zur Verbesserung wurde eine kardanische Aufhängung entwickelt, die die Bodenplatte in der Waagerechten hält. In Abbildung 11.25 ist ein Foto des Quadropters zum Stand März 2013 zu sehen.

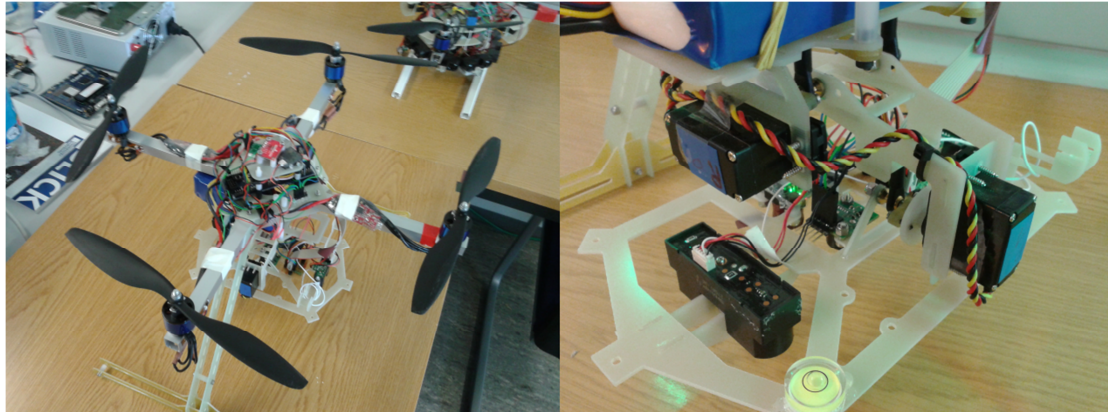


Abbildung 11.25: AQuadropter18 (links) und dazugehörige Bodenplatte (rechts), Stand ca. 3/2013

### 11.4.6 Objektsuche

Mit der Integration des LP 180 und einer nach unten gerichteten Kamera (C 290) waren im Mai 2015 die Voraussetzungen zur Objektsuche geschaffen (Abbildung 11.26). Die dazugehörige Objekterkennung entwickelte Christian Reul in seiner Bachelorarbeit. Somit war der Quadropters bis zum Oktober 2015 in der Lage autonom in einem Raum Objekte zu suchen und deren Anzahl und Position zu bestimmen.

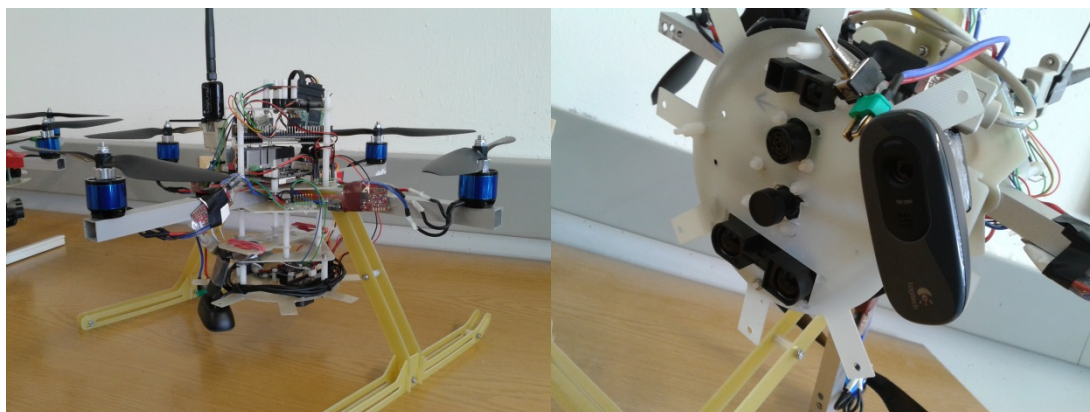


Abbildung 11.26: AQuadropter18 (links) und dazugehörige Bodenplatte (rechts), Stand ca. 4/2013 – 11/2013

### 11.4.7 Kollisionsvermeidung mit Ultraschall und Infrarot

Die Hinderniserkennung und Kollisionsvermeidung konnte durch Verwendung von Infrarotsensoren deutlich verbessert werden (Abbildung 11.27). Dazu mussten wesentliche Teile neu entwickelt werden. Dies war ein langwieriger Prozess der im Wesentlichen im Zeitraum Februar 2013 bis Mai 2015 stattfand. Paul Benz leistete im Rahmen seiner Bachelorarbeit einen Beitrag dazu. Am Ende konnte der Quadropters Wänden und Personen ausweichen sowie in einem Tunnel fliegen.



Abbildung 11.27: AQuadropter Stand ca. 4/2013

#### 11.4.8 Stereo-optische Hinderniserkennung

Abbildung 11.28 zeigt den AQuadropter mit zwei Sensoren zur stereo-optischen Abstandsmessung. Das Verfahren zur Abstandsbestimmung aus zwei Kamerabildern wurde von Helge Lauterbach im September 2013 implementiert und in den AQuadropter integriert.

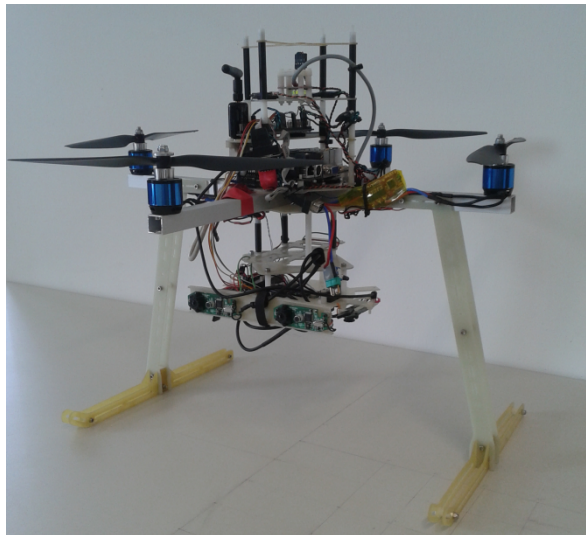


Abbildung 11.28: AQuadropter Stand ca. 9/2013

#### 11.4.9 Gerichteter Flug

Der OF Sensor bringt die beiden Nachteile der starken Abhängigkeit von guten Lichtverhältnissen und der Beschränkung auf Translationen mit sich. Rotationen führen zu Fehlern. Um diese Problematik zu überwinden, wurde von Eric Reinthal mit dem Fourier Tracking (März 2014) eine alternative Lösung zur Bestimmung der Position geschaffen. Damit wurde der autonome, gerichtete Flug möglich.

#### 11.4.10 Auszeichnung mit Universitätsförderpreis

Der Autor dieser Arbeit wurde für das Projekt „Rettungshelfer mit Propellern“ im April 2014 mit dem Universitätsförderpreis der Mainfränkischen Wirtschaft ausgezeichnet (Abbildung 11.29). Ein mögliches Anwendungsszenario ist der Feuerwehreinsatz. Das bisherige System wurde mit Hilfe des Förderpreises um eine entscheidende Komponente, einen Manipulator bzw. einen Roboterarm, erweitert, um es zu ermöglichen, Gegenstände zu bewegen und in die Umgebung einzugreifen. Die Idee dabei ist es, dass der Quadrokopter lernen soll, Gegenstände aufzunehmen und zu

transportieren, Knöpfe zu drücken oder auch Türen zu öffnen, um eines Tages (im Schwarm) Verletzte zu bergen, ohne dass sich auch nur ein Mensch in Gefahr bringen muss.

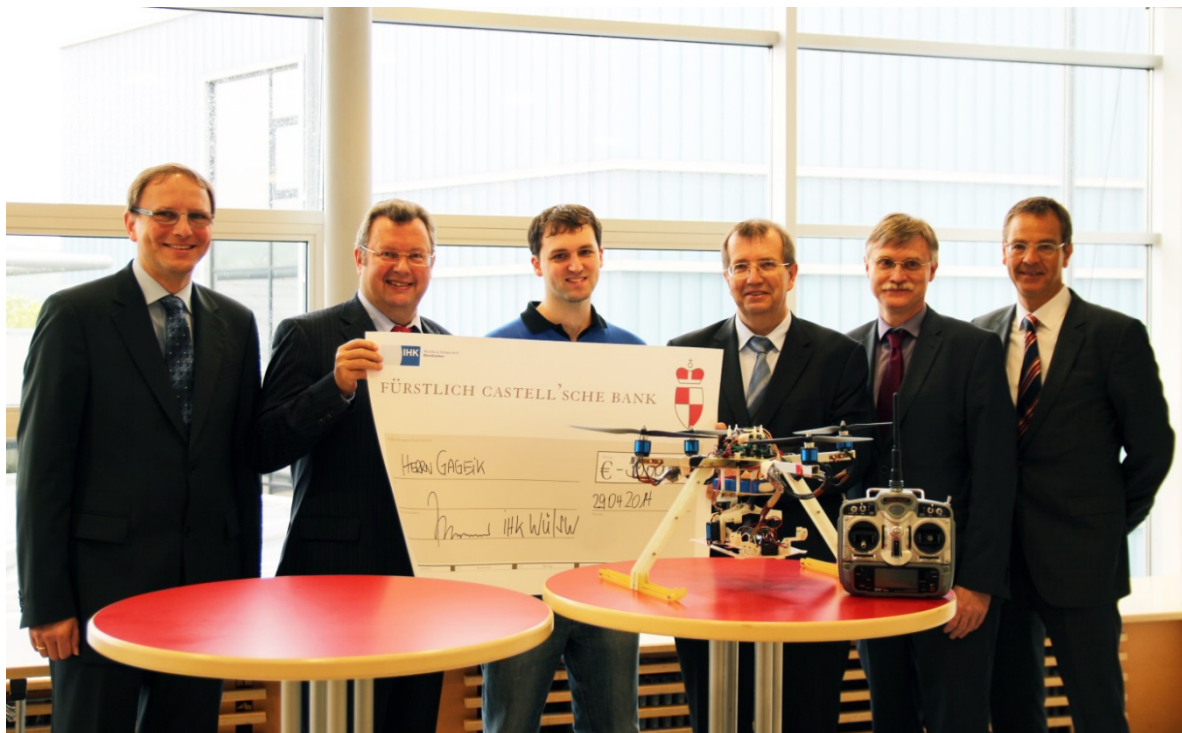


Abbildung 11.29: Checkübergabe 4/2014

Main Post v. 30.4.2014, S.17

## Rettungshelfer mit Propeller

IHK Würzburg-Schweinfurt vergibt Universitätsförderpreis 2014 an den Forscher Nils Gageik

Von unserem Mitarbeiter  
**BERNHARD RAUH**

**WÜRZBURG** Seit einigen Jahren erfreuen sich Quadrocopter einer steigenden Beliebtheit. Diese rucksackgroßen und mit vier Propellern ausgestatteten Fluggeräte nutzen nicht nur Firmen für Luftbeobachtungen. Auch immer mehr Privatleute setzen diese nunmehr erschwinglich gewordenen Helfer ein, um beispielsweise schadhafte Stellen auf Dächern aufzuspüren oder Sportereignisse von oben zu verfolgen.

Doch die möglichen Einsatzbereiche dieser sehr wendigen Geräte können noch viel größer sein. Und für eine solche Idee hat nun Nils Gageik den Universitätsförderpreis der IHK Würzburg-Schweinfurt in Höhe von 35 000 Euro erhalten. Gageik arbeitet an einem Forschungsprojekt des Lehrstuhls für Informationstechnik für Luft- und Raumfahrt in Würzburg und sucht nach Möglichkeiten, einen Quadrocopter als „helfende Hand“ für spezielle Arbeiten der Feuerwehr einzusetzen: „Ausgehend von unserer bisherigen Arbeit

möchten wir unser System mit Hilfe des Förderpreises nun um eine entscheidende Komponente, wie zum Beispiel einen Roboterarm, erweitern, um es dem fliegenden Lebensretter zu ermöglichen, Gegenstände zu bewegen und in seine Umgebung einzugreifen.“ So ist daran gedacht, die fliegende Drohne in brennende Gebäude zu schicken, um Verletzte zu suchen oder Atemmasken, Schutzanzüge und Medikamente in Gebiete zu transportieren, die durch giftige Gase oder radioaktive Strahlung verseucht sind. Auch einsturzgefährdete Innenräume wie Tunnel und Höhlen könnten mit einem speziell ausgestatteten Quadrocopter begutachtet werden.

Eine besondere Bedeutung sieht Gageik in der Selbstständigkeit des Systems. „Wir legen den Schwerpunkt auf die Entwicklung autonomer Drohnen zur Innenraumerkundung in unbekannter Umgebung, ohne Verwendung externer Referenzsysteme wie zum Beispiel GPS.“ Das heißt, die Drohne soll selbstständig und zuverlässig Arbeiten in für Menschen schwer zugänglichen

Bereichen durchführen und sich dabei mittels Infrarot-, Radar- und Ultraschall-Signalen orientieren.

Nils Gageik konnte sich mit seinem Projekt gegenüber 13 Mitbewerbern durchsetzen und erhielt von IHK-Präsident Dieter Pfister und in Anwesenheit von Universitätspräsident Prof. Dr. Alfred Forchel, Uni-Kanzler Dr. Uwe Klug und dem

Schriftführer des Universitätsbundes Dr. Alfons Ledermann den Scheck, der die weitere Finanzierung seiner Forschungen sichert. Der Universitäts-Förderpreis speist sich aus dem Kapital einer Stiftung, die die IHK vor 32 Jahren im Universitätsbund ins Leben gerufen hat. Bis heute konnten 89 Projekte von insgesamt knapp 900 000 Euro profitieren.

Fliegender Helfer: Preisträger Nils Gageik, mit Quadrocopter. FOTO: B. RAUH

Abbildung 11.30: Zeitungsartikel 4/2014

Das Projekt heißt „Rettungshelfer mit Propellern“, da der Fokus dieses Vorhabens im Eingriff in die Umgebung liegt: Der Quadrocopter stellt quasi eine weitere helfende Hand dem Menschen zur



Verfügung (Abbildung 11.30).

#### 11.4.11 Lidar-basierte Kollisionsvermeidung

Seit dem September 2015 kann der AQopter18 auch mit Hilfe des Lidar-Sensors Kollisionen durch Abstandsregelung vermeiden. Robin Lelebici leistete im Rahmen seiner Bachelorarbeit dazu einen Beitrag und entwickelte Teile der Hinderniserkennung weiter. In Abbildung 11.31 ist der AQopter18 mit Lidar zu sehen.

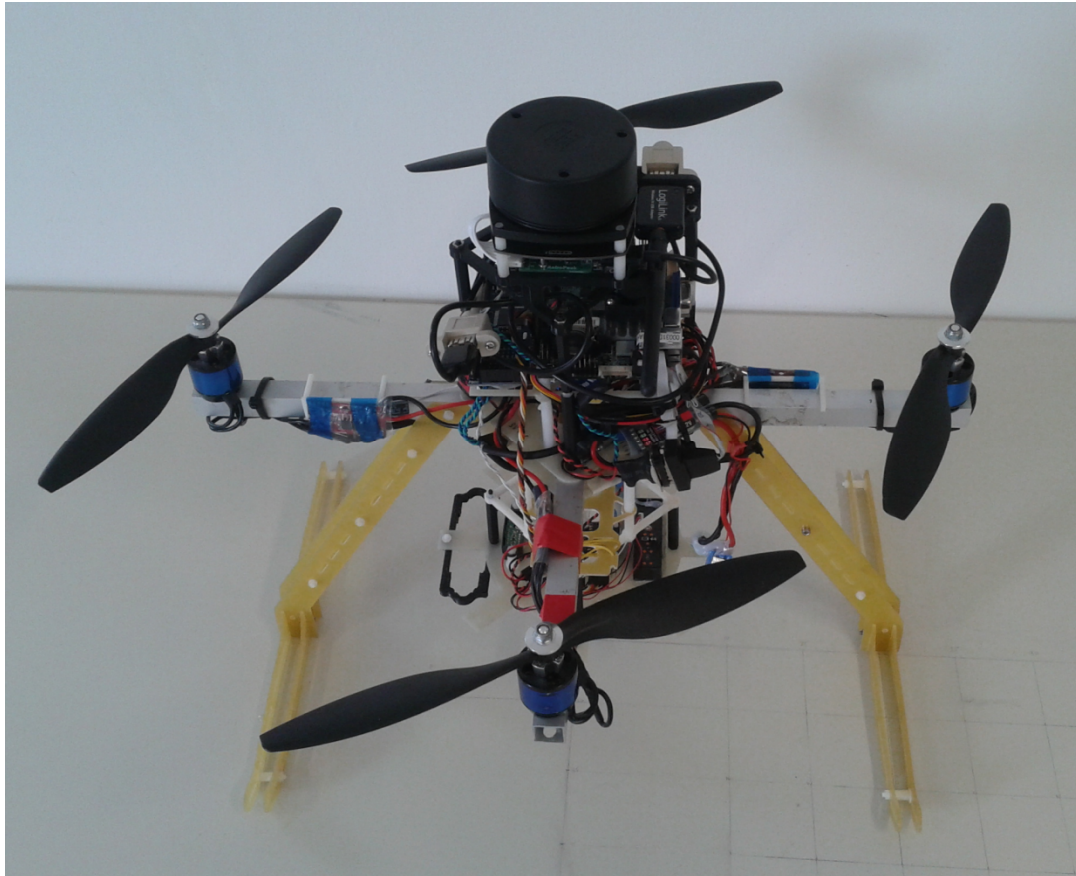


Abbildung 11.31: AQopter18 mit Lidar

#### 11.4.12 AQopter18 Heute

Heute (9/2015) existieren die in Abbildung 11.32 zu sehenden 10 Quadrokopter als Prototypen des AQopter18. Dies sind von vorne nach hinten:

- 1 Quadrokopter mit GPS und Bodenplatte
- 1 Quadrokopter mit kardanischer Aufhängung der Bodenplatte
- 3 Quadrokopter mit LP-180, davon
  - 1 Quadrokopter mit aller Sensorik der Arbeit (ohne Lidar)
  - 1 Quadrokopter mit SV-System
  - 1 Quadrokopter mit Lidar (RP Lidar)
- 2 Quadrokopter mit US + IR Sensoren zur Kollisionsvermeidung
- 3 Quadrokopter mit Manipulator (je 1 Quadrokopter pro Design des Manipulators)



Abbildung 11.32: AQuadcopter8 Flotte besetzt Seminarraum (9/2015)

## 11.5 Arbeiten am AQopterI8-Projekt

Im Rahmen des AQopterI8-Projekts wurden von mir während meiner Tätigkeit am Lehrstuhl 37 studentische Arbeiten (Tabelle 11.1, Tabelle 11.2) und diverse studentische Hilfskräfte betreut, die viele Teilaspekte des Projekts bearbeiteten und untersuchten. Die 37 studentischen Arbeiten teilen sich auf in 2 Masterarbeiten, 3 Diplomarbeiten, 26 Bachelorarbeiten und 6 Praktika. Viele dieser Arbeiten hatten Anteil am Gesamtprojekt.

Zur Betreuung der studentischen Arbeiten gehörten die Ausarbeitung der Aufgabenstellung und Fragestellung der Arbeiten sowie die Begleitung der Arbeit. Dies inkludiert bzw. impliziert oftmals auch das Konzept oder Teile dessen. Zur Betreuung gehörte auch die Hilfestellung bei Problemen aller Art, insbesondere bei der Evaluierung des fliegenden Systems sowie Hilfe bei der Integration. Somit sind viele Aspekte gemeinsam im Team bearbeitet wurden, so dass die verschiedenen Systeme und Arbeiten des Gesamtsystems ineinander greifen. Größtenteils sind die Arbeiten online abrufbar.

**Tabelle 11.1: Von mir betreute Abschlussarbeiten des AQopterI8-Projekts**

Verweis	Datum	Art	Thema	Autor
A01	12 / 2011	D	Implementierung und Evaluierung eines Systems zur Hinderniserkennung und Kollisionsvermeidung für Indoor-Quadrocopter	Thilo Müller
A02	07 / 2012	B	<a href="#">Implementierung und Evaluierung einer Höhenregelung für einen Quadrocopter</a>	Julian Rothe
A03	08 / 2012	B	Implementation and Evaluation of a Differential GPS System	Christian Hofmann
A04	09 / 2012	B	<a href="#">Implementierung und Evaluierung einer Positionsregelung unter Verwendung des optischen Flusses</a>	Michael Strohmeier
A05	12 / 2012	B	<a href="#">Intelligentes Mapping für Indoor-Quadrocopter</a>	Norbert Schmitt
A06	12 / 2012	B	<a href="#">Implementierung einer Datenfusion von Inertial- und Magnetsensor für einen Quadrocopter</a>	Lukas Iffländer
A07	01 / 2013	D	Entwicklung eines autonomen Prallluftschiffs	Johannes Wischert
A08	02 / 2013	D	<a href="#">Design und Implementierung eines konfigurierbaren Controllers zur Ansteuerung bürstenloser Motoren und Regelung von Echtzeitsystemen</a>	Erik Dilger
A09	04 / 2013	B	<a href="#">Implementierung und Evaluierung einer Objekterkennung für einen Quadrocopter</a>	Christian Reul
A10	04 / 2013	B	<a href="#">Implementierung und Evaluierung eines Systems zur Hinderniserkennung und Kollisionsvermeidung für Indoor-Quadrocopter</a>	Paul Benz
A11	07 / 2013	B	<a href="#">Implementierung und Evaluierung einer Positionserfassung durch Phasenkomparation elektromagnetischer Wellen</a>	Bastian Scholz
A12	07 / 2013	B	<a href="#">Implementierung und Evaluierung einer radarbasierten Geschwindigkeitsbestimmung für einen Quadrocopter</a>	Peter Hein

A13	08 / 2013	B	<a href="#">Implementierung und Evaluierung verschiedener Algorithmen zur autonomen Suche eines Quadrokters</a>	Paul Barth
A14	08 / 2013	B	<a href="#">Implementierung und Evaluierung eines kartenbasierten Lokalisationsverfahrens für einen autonomen Quadrokopter</a>	Damian Rothkegel
A15	09 / 2013	M	<a href="#">Design and Implementation of a 6DOF Control System for an Autonomous Quadrocopter</a>	Alexander Lebedev
A16	09 / 2013	B	<a href="#">Stereo-Optische Abstandsmessung für einen autonomen Quadrocopter</a>	Helge Lauterbach
A17	09 / 2013	B	<a href="#">Punktgenaues Starten und Landen eines Quadrokters mittels optischer Sensorik</a>	Matthias Büttner
A18	09 / 2013	B	<a href="#">Entwicklung einer ad-hoc W-Lan Gui-basierten Steuerung von multiplen Quadrokoptern</a>	Joshua Smolka
A19	09 / 2013	B	<a href="#">Implementierung und Evaluierung einer Branddetektion für einen autonomen Quadrocopter</a>	Oliver Kleinke
A20	10 / 2013	M	<a href="#">Design and Implementation of a Failsafe Solution for Quadrocopters</a>	Tobias Roos
A21	01 / 2014	B	<a href="#">Optimierung einer magnetischen Orientierungskompensation durch Magnetfeldfehlererkennung</a>	Matthias Bethge
A22	03 / 2014	B	<a href="#">Softwaredesign und Regelung eines autonomen Prallluftschiffes</a>	Felix Schwieger
A23	03 / 2014	B	<a href="#">Positionsbestimmung eines autonomen Quadrokters durch Bildverarbeitung</a>	Eric Reinthal
A24	03 / 2014	B	Autonomer Quadrocopter zur Feuersuche	Sascha Dechend
A25	10 / 2014	B	<a href="#">Implementierung einer HMD-basierten Telemetrie-Anzeige für einen Quadrokopter</a>	Deniz Caliskan
A26	11 / 2014	B	<a href="#">Implementierung und Evaluierung einer radarbasierten Abstandsmessung für einen Quadrocopter</a>	Ferdinand Seidel
A27	12 / 2014	B	Miniatur-Steuergerät für einen Quadrocopter	Hasan Tuncay
A28	12 / 2014	B	GPS-basierte Steuerung eines autonomen Quadrokters	Andrea Tüchert
A29	05 / 2015	B	<a href="#">Kooperative Steuerung multipler Quadrokopter mittels optischem Tracking</a>	Arthur Scharf
A30	05 / 2015	B	<a href="#">Radarbasierte Hinderniserkennung für einen autonomen Quadrocopter</a>	Tim Jonas Horst
A31	09/2015	B	Laserscanner basierte Hinderniserkennung für einen autonomen Quadrocopter	Robin Lelebici

**Tabelle 11.2: Von mir betreute Praktika zum AQopter18-Projekt**

Name	Thema
Florian Kempf	Erstellung einer Messstation zur Bestimmung der Übertragungsfunktion eines Quadropters
Erik Dilger	Entwicklung und Implementierung eines flexiblen Kommunikationsprotokolls für einen Quadrocopter
Alexander Lebedev	System-Modellierung eines Quadropters
Christian Reul	Intelligente Objekterkennung und –suche für einen autonomen Quadrocopter
Maximillian Becker	Smartphone-Steuerung für einen Quadrocopter
Paul Benz	Kollisionsvermeidung für Indoor-Quadrocopter

## 11.6 Wissenschaftliche Veröffentlichungen

Tabelle 11.3 ist eine Auflistung der von mir im Rahmen des AQopter18-Projekts gemachten wissenschaftlichen Veröffentlichungen (Paper), während meiner Zeit als wissenschaftlicher Mitarbeiter des Lehrstuhls Informatik 8. Einige Teile dieser Dissertation beruhen auf diesen Arbeiten oder greifen auf die Ergebnisse zurück.

**Tabelle 11.3: Eigene Wissenschaftliche Veröffentlichungen im Rahmen des AQopter18-Projekts**

Datum	Verweis	Ort	Art	Titel
01 / 2012	P1	INTED	Konferenz	<a href="#">Avionic Control Systems for Education And Development</a>
11 / 2012	P2	UAVveek	Workshop	<a href="#">Obstacle Detection and Collision Avoidance using Ultrasonic Distance Sensors for Autonomous Quadrocopter</a>
11 / 2012	P3	UAVveek	Workshop	<a href="#">Data Fusion Principles for Height Control and Autonomous Landing of a Quadrocopter</a>
05 / 2013	P4	IJAIA Vol.4 Nr.3	Journal	<a href="#">Waypoint Flight Parameter Comparison of an Autonomous UAV</a>
10 / 2013	P5	Intech IJARS	Journal	<a href="#">An Autonomous UAV with Optical Flow Sensor for Positioning and Navigation</a>
11 / 2013	P6	UAV World	Konferenz	<a href="#">Autonomous Quadrocopter for Search, Count and Localization of Objects</a>
05 / 2014	P7	IJAIA Vol.5 Nr.4	Journal	<a href="#">A Review on Distributed Control Of Cooperating Mini UAVs</a>
09 / 2014	P8	IJAIA Vol.5 Nr.5	Journal	<a href="#">Complementary Vision Based Data Fusion for Robust Positioning and Directed Flight of an Autonomous Quadrocopter</a>
03 / 2015	P9	SPIE EI Vol. 9411	Konferenz	<a href="#">Concept for practical exercises for studying autonomous flying robots in a university environment: part II</a>
05 / 2015	P10	IEEE Access	Journal	<a href="#">Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors</a>

## 11.7 Beiträge in Medien zum AQopterI8

In Tabelle 11.4 sind 10 Beiträge in Medien zum AQopterI8-Projekt aufgelistet. Hierbei handelt es sich um 4 Zeitungsartikel, 4 Fernsehbeiträge und 2 Radiointerviews. Die Liste erhebt keinen Anspruch auf Vollständigkeit.

**Tabelle 11.4: Medienbeiträge über das AQopterI8-Projekt**

Titel	Art	Autor(en)	Quelle und Datum
<a href="#">Fliegende Lebensretter</a>	Zeitungsartikel	<a href="#">Micheale Schneider</a>	<a href="#">Fränkischer Tag</a> <sup>⌵</sup> Ausgabe 8.8.2012
Lebensretter mit Propellern	Zeitungsartikel	N. Bodenbach, T. Obermeier, T. Mueller	<a href="#">Main Post</a> <sup>⌵</sup> , Juni 2012
Uni Würzburg entwickelt Rettungs-Drohne	Fernsehbeitrag	A. Lehnert	<a href="#">Sat.1 Bayern</a> <sup>⌵</sup> , 17:30 24.08.2012
<a href="#">Drohnen als Lebensretter einsetzen</a>	Zeitungsartikel	Sylvia Schubart-Arand	<a href="#">Main Post</a> <sup>⌵</sup> , März 2014
Franken kompakt	Fernsehbeitrag	unbekannt	<a href="#">BR</a> <sup>⌵</sup> , 17:30 29.04.2014
Uni-Förderpreis	Radiointerview	Jürgen Gläser	<a href="#">BR1</a> <sup>⌵</sup> , 12:05 29.04.2014 Mittags in Mainfranken
Universitäts-Förderpreis	Radiointerview	Tanja Oppelt	<a href="#">BR2</a> <sup>⌵</sup> , 13:30 29.04.2014 regionalZeit
<a href="#">Rettungshelfer mit Propellern</a>	Zeitungsartikel	Bernhard Rauh	<a href="#">Main Post</a> <sup>⌵</sup> , 29.04.2014
<a href="#">Förderpreis für fliegende Retter</a>	Fernsehbeitrag	unbekannt	<a href="#">TV Touring</a> <sup>⌵</sup> , 30.04.2014
Drohnen: Fliegende Roboter als Lebensretter	Fernsehbeitrag	Ralph Wege	<a href="#">BR</a> <sup>⌵</sup> , 17:30 28.05.2014

## 11.8 Das AQopterI8 Team



Abbildung 11.33: AQopterI8 Team (v.l.n.r.: Erik Dilger, Nils Gageik, Qasim Ali, Sascha Dechend, Paul Benz, Tim Horst, Robin Leblebici, Lekha Sumathy)

## 11.9 Quellen- und Literaturverzeichnis

- [1] BR, „Förderpreis für "Rettungshelfer mit Propellern",“ 28 04 2014. [Online]. Available: <http://www.br.de/nachrichten/unterfranken/quadrokopter-120.html>. [Zugriff am 30 10 2014].
- [2] H. W. Laumanns, Drohnen, Stuttgart: Motorbuch Verlag, 2012.
- [3] MikroKopter, 16 09 2014. [Online]. Available: <http://www.mikrokopter.de>. [Zugriff am 16 09 2014].
- [4] R. Büchi, Faszination Quadrokopter, vth, 2010.
- [5] Bibliographisches Institut GmbH, Duden, 2015.
- [6] K. K. F. S. S. W. W. N. D. Nonami, Autonomous Flying Robots, Springer, 2010.
- [7] Microdrones, 17 09 2014. [Online]. Available: <http://www.microdrones.com>. [Zugriff am 17 09 2014].
- [8] J. Engel, „Accurate Figure Flying with a Quadrocopter Using Onboard Visual and Inertial Sensing,“ in *International Conference on Intelligent Robot Systems*, 2012.
- [9] D. Mellinger, „Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors,“ in *Int. Symposium on Experimental Robotics*, 2010.

- [10] S. Montenegro, *Avioniks für Flugzeuge*, 2012.
- [11] Wikipedia, „Die freie Enzyklopädie,“ 03 09 2014. [Online]. Available: [de.wikipedia.org/](http://de.wikipedia.org/). [Zugriff am 03 09 2014].
- [12] I. R. N. Roland Siegwart, *Introduction to Autonomous Mobile Robots*, Cambridge: A Bradford book, 2004.
- [13] Beo Lingus, „Beo Lingus,“ 2014. [Online]. Available: <http://dict.tu-chemnitz.de>. [Zugriff am 03 06 2014].
- [14] Sharp, „Datenblatt GP2Y0A710K0F,“ 2006.
- [15] Sharp, „Datenblatt GP2Y0A02YK“.
- [16] Tritech, „Datenblatt Micron Sonar Ultra Compact Imaging Sonar,“ 05 01 2015. [Online]. Available: <http://www.tritech.co.uk/media/products/small-rov-mechanical-sector-scanning-sonar-tritech-micron.pdf>. [Zugriff am 05 01 2015].
- [17] Sparkfun, 2014. [Online]. Available: <https://www.sparkfun.com>. [Zugriff am 03 06 2014].
- [18] „Pololu,“ 2014. [Online]. Available: <http://www.pololu.com>. [Zugriff am 06 03 2014].
- [19] „InvenSense,“ 2014. [Online]. Available: <http://invensense.com/>. [Zugriff am 03 06 2014].
- [20] InvenSense Gyro, „Datenblatt ITG-3200,“ 2010.
- [21] Analog Devices, „Datenblatt ADXL345,“ 2009.
- [22] InvenSense Acc, „Datenblatt IMU-3000,“ 2010.
- [23] STMicroelectronics, „Datenblatt L3G4200D,“ 2010.
- [24] STMicroelectronics, „Datenblatt LSM303DLM,“ 2009.
- [25] STMicroelectronics, „Datenblatt L3GD20,“ 2013.
- [26] STMicroelectronics, „Datenblatt LSM303DLHC,“ 2011.
- [27] InvenSense, „MPU-6000/MPU-6050 EV Board User Guide,“ 2011.
- [28] S. Bonnet, „Calibration methods for inertial and magnetic sensors,“ *Elsevier*, pp. 302-311, 2009.
- [29] D. Roetenberg, „Compensation of Magnetic Disturbances Improves Inertial and Magnetic Sensing of Human Body Segment Orientation,“ *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, pp. 395-405, September 2005.



- [30] H. R. Everett, *Sensors for mobile Robots*, Natick: A K Peters Ltd, 1995.
- [31] Avago Technologies, „ADNS-3080 High-Performance Optical Mouse Sensor,“ 2012.
- [32] PMD Technologies GmbH, „A performance review of 3D TOF vision systems in comparison to stereo vision systems,“ Siegen, 2012.
- [33] J. Roßmann, *Mensch-Maschine-Interaktion und Robotik 1 (Grundlagen)*, Aachen: Institut für Mensch-Maschine-Interaktion, RWTH Aachen, 2008.
- [34] J. B. Kuipers, *Quaternions and Rotation Sequences*, Princeton: Princeton University Press, 1999.
- [35] A. M. Sabatini, „Quaternion-Based Extended Kalman Filter for Determining Orientation by Inertial and Magnetic Sensing,“ in *TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 2006.
- [36] A. Tobergte, „Robust Multi Sensor Pose Estimation for Medical Applications,“ in *International Conference on Intelligent Robots and Systems*, St. Louis, USA, 2009.
- [37] N. Parnian, *Integration of Local Positioning System & Strapdown Inertial Navigation System for Hand-Held Tool Tracking*, Waterloo, Ontario, Canada: University of Waterloo, 2008.
- [38] B. Sadeghi, „Second-order EKF and Unscented Kalman Filter Fusion for Tracking Maneuvering Targets,“ *IEEE*, 2007.
- [39] L. Klingebeitl, *Entwicklung eines modularen und skalierbaren Sensorsystems zur Erfassung von Position und Orientierung bewegter Objekte*, Bonn: ISSN-0172-8741, 2006.
- [40] D. Roetenberg, *Inertial and Magnetic Sensing of Human Motion*, Twente: University of Twente, 2006.
- [41] J. Kotlarski, „Improving the performance of an optical tracking system using data fusion with an inertial measurement unit,“ in *11th International Conference on Mechatronics Technology (ICMT)*, Ulsan, Korea, 2007.
- [42] N. Gageik, *Entwicklung und Evaluierung eines Systems zur redundanten Positionserfassung eines neurochirurgischen robotischen Sicherheitstrepanationssystems*, Aachen: Rheinisch-Westfälische Technische Hochschule Aachen, 2010.
- [43] R. Mahony, „Nonlinear Complementary Filters on the Special Orthogonal Group,“ *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, pp. 1203-1218, Juni 2008.
- [44] J. Lunze, *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*, Springer, 2014.
- [45] M. Horn und N. Dourdoumas, *Regelungstechnik*, Pearson Studium, 2006.

- [46] M. Reuter und S. Zacher, *Regelungstechnik für Ingenieure*, Wiesbaden: Vieweg + Teubner, 2008.
- [47] R. W. Beard, *Quadrotor Dynamics and Control*, Brigham Young University, 2008.
- [48] S. Bouabdallah, *Design and control of quadrotors with application to autonomous flying*, Lausanne, 2007.
- [49] C. Chauffaut, „Towards Gun- and Aircraft-launched MAVs: Embedded Flight Control System,“ *Universite de Technologie de Compiegne*, Compiegne, Frankreich, 2011.
- [50] I. S. a. P. Corke, „System Identification, Estimation and Control for a Cost Effective Open-Source Quadcopter,“ in *IEEE International Conference on Robotics and Automation*, 2011.
- [51] K. S. Fu, *Robotics: Control, Sensing, Vision and Intelligence*, McGraw-Hil Inc., 1987.
- [52] B. Herisse, „Hovering flight and vertical landing control of a VTOL Unmanned Aerial Vehicle using Optical Flow,“ in *International Conference on Intelligent Robots and Systems*, Nice, France, 2008.
- [53] G. M. Hoffmann, „Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment,“ *American Institute of Aeronautics and Astronautics*, 2007.
- [54] T. Luukkonen, „Modelling and control of quadcopter,“ Espoo, 2011.
- [55] R. Mahony, „A coupled estimation and control analysis for attitude stabilisation of mini aerial vehicles,“ Dept. Eng., ANU, ACT., Australia, 2006.
- [56] P. S. Maybeck, *Stochastic models, estimation, and control*, New York, San Francisco, London: Academic Press, 1979.
- [57] S. Salazar-Cruz, „Embedded control system for a four-rotor UAV,“ *International Journal of Adaptive Control and Signal Processing*, pp. 189-204, 31 01 2007.
- [58] J. Barron, „Performance of Optical Flow Techniques,“ *International Journal of Computer Vision*, pp. 43-77, 1994.
- [59] S. Beauchemin, „The Computation of Optical Flow,“ London, Ontario, Canada, 1995.
- [60] B. Jähne, *Digitale Bildverarbeitung*, Springer, 2005.
- [61] B. D. Lucas und T. Kanade, „An Iterative Image Registration Technique with an Application to Stereo Vision,“ in *Proc 7th Intl Joint Conf on Artificial Intelligence (IJCAIA)*, Vancouver, British Columbia, 1981.
- [62] M. V. Srinivasan, „An image-interpolation technique for the computation of optic flow and egomotion,“ *Biological Cybernetics*, pp. 401-415, 03 09 1993.

- [63] A. Averbuch, „A unified approach to FFT based image registration,“ Tel Aviv University, 2002.
- [64] B. S. Reddy, „An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration,“ in *IEEE Transactions on Image Processing*, 1996.
- [65] R. Siegwart, *Introduction to Autonomous Mobile Robots*, Cambridge: The MIT Press, 2004.
- [66] S. BOUABDALLAH, „Toward Obstacle Avoidance on Quadrotors,“ in *Proceedings of the XII International Symposium on Dynamic Problems of Mechanics*, Ilhabela, Brazil, 2007.
- [67] J. F. Roberts, *Quadrotor Using Minimal Sensing For Autonomous Indoor Flight*, 2007.
- [68] C. Raviteja, „Development of an autonomous aerial vehicle using Laser range,“ 2010.
- [69] Walkera, 2014. [Online]. Available: <http://www.walkera.com>. [Zugriff am 25 07 2014].
- [70] J.-Y. Bouguet, 09 09 2014. [Online]. Available: <http://www.vision.caltech.edu/bouguet/>. [Zugriff am 09 09 2014].
- [71] G. Bradski, *Learning OpenCV: Computer Vision with the OpenCV Library*, Sebastopol, CA, USA: O'Reilly Media, 2008.
- [72] L. De-Maetzu und S. Mattocchia, „Linear stereo matching,“ in *International Conference on Computer Vision (ICCV)*, Barcelona, 2011.
- [73] H. Hirschmüller, „Accurate and efficient stereo processing by semi-global matching and mutual information,“ in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [74] H. Hirschmüller, „Stereo Vision in Structured Environments by Consistent Semi-Global Matching,“ in *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [75] H. Hirschmüller, „Stereo Processing by Semiglobal Matching and Mutual Information,“ in *Transactions on Pattern Analysis and Machine Intelligence*, 2008.
- [76] MPr, PMD Technologies, „4 phase algorithm using raw data for PMD[vision] CamCube/CamBoard,“ Siegen, 2012.
- [77] S. Grönzka, *Mapping, State Estimation, and Navigation for Quadrotors and Human-Worn Sensor Systems*, Albert-Ludwigs-Universität Freiburg im Breisgau: Dissertation, 2011.
- [78] D. B. a. G. D. Hager, „V-GPS(SLAM): Vision-Based Inertial System for Mobile Robots,“ in *IEEE: International Conference on Robotics & Automation*, New Orleans, 2004.
- [79] S. Weiss, „Monocular-SLAM-Based Navigation for Autonomous Micro Helicopter in GPS-Denied Environments,“ *Journal of Field Robotics*, 2011.

- [80] „ArduCopter,“ 16 09 2014. [Online]. Available: <http://www.arducopter.co.uk/>. [Zugriff am 16 09 2014].
- [81] S. A. Raza, „Intelligent Flight Control of an Autonomous Quadrotor,“ *InTech*, 2010.
- [82] Natural Point, Inc., 16 09 2014. [Online]. Available: <http://www.naturalpoint.com>. [Zugriff am 16 09 2014].
- [83] AggressiveManeuvers, „Youtube,“ 21 05 2010. [Online]. Available: <http://www.youtube.com/watch?v=MVRTALJp8DM>. [Zugriff am 16 09 2014].
- [84] NanoSwarm, „Youtube,“ 31 01 2012. [Online]. Available: <http://www.youtube.com/watch?v=YQIMGV5vtd4>. [Zugriff am 16 09 2014].
- [85] QuadrotorsMusic, „Youtube,“ 28 02 2012. [Online]. Available: [http://www.youtube.com/watch?v=\\_sUeGC-8dyk](http://www.youtube.com/watch?v=_sUeGC-8dyk). [Zugriff am 16 09 2014].
- [86] AstoundingAthletics, „Youtube,“ 11 06 2013. [Online]. Available: <http://www.youtube.com/watch?v=w2itwFJcGfQ>. [Zugriff am 16 09 2014].
- [87] DIY, „DIY Drones,“ 16 09 2014. [Online]. Available: <http://diydrones.com/>. [Zugriff am 16 09 2014].
- [88] 3D Robotics, 16 09 2014. [Online]. Available: <http://3drobotics.com>. [Zugriff am 16 09 2014].
- [89] Aibotix, 19 09 2014. [Online]. Available: <http://www.aibotix.de/>. [Zugriff am 19 09 2014].
- [90] Roboterpartner, 19 09 2014. [Online]. Available: <http://www.roboterpartner.com/roboter-drohnen/hexacopter-drohnen-aibotix/>. [Zugriff am 19 09 2014].
- [91] „News AT,“ 22 05 2012. [Online]. Available: <http://www.news.at/a/microdrones-drohnen-alltag-328580>. [Zugriff am 17 09 2014].
- [92] M. Blösch, „Vision Based MAV Navigation in Unknown and Unstructured Environments,“ in *Robotics and Automation (ICRA)*, Anchorage, AK, 2010.
- [93] S. Grzonka, „A Fully Autonomous Indoor Quadrotor,“ *IEEE Transactions on Robotics*, Vol. 28, Issue: 1, pp. 90-100, 02 02 2012.
- [94] S. Lange, „Autonomous Landing for a Multirotor UAV Using Vision,“ in *International Conference on Simulation, Modeling and Programming for Autonomous Robots*, Venedig, 2008.
- [95] J. Engel, „Camera-based Navigation of a Low-Cost Quadrocopter,“ in *International Conference on Intelligent Robot Systems*, 2012.

- [96] Ascending Technologies, „Asctec,“ 16 09 2014. [Online]. Available: <http://www.asctec.de/>. [Zugriff am 16 09 2014].
- [97] Ascending Technologies, „Research Price List,“ Krailling, Germany, 2013.
- [98] S. Shen, „Autonomous Multi-Floor Indoor Navigation with a Computationally Constrained MAV,“ in *International Conference on Robotics and Automation*, Shanghai, 2011.
- [99] P. Bouffard, „Quadrotor + Kinect,“ 2010. [Online]. Available: <http://hybrid.eecs.berkeley.edu/~bouffard/kinect.html>. [Zugriff am 18 09 2014].
- [100] S. Crawford, „How Microsoft Kinect Works,“ 2010. [Online]. Available: <http://electronics.howstuffworks.com/microsoft-kinect2.htm>. [Zugriff am 18 09 2014].
- [101] H. Martin, „OpenKinect,“ 2010. [Online]. Available: <https://github.com/OpenKinect/libfreenect>. [Zugriff am 18 09 2014].
- [102] Microsoft, „Kinect for Windows,“ 2014. [Online]. Available: <http://www.microsoft.com/en-us/kinectforwindows/>. [Zugriff am 18 09 2014].
- [103] ROS, „Quadrotor Altitude Control and Obstacle Avoidance,“ 09 03 2011. [Online]. Available: <http://wiki.ros.org/openni/Contests/ROS%203D/Quadrotor%20Altitude%20Control%20and%20Obstacle%20Avoidance>. [Zugriff am 18 09 2014].
- [104] M. Becker und S. Bouabdallah, „In-Flight Collision Avoidance Controller Based Only on OS4 Embedded Sensors,“ *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, pp. 294-307, Juli-September 2012.
- [105] igus, 05 01 2015. [Online]. Available: <http://www.igus.de>. [Zugriff am 05 01 2015].
- [106] ddp-bay, „Chemieunfall an der Uni Würzburg,“ *Augsburger Allgemeine*, 2008.
- [107] H. Crolly, „200 Helfer versuchen, den Höhlenforscher zu retten,“ *Die Welt*, <http://www.welt.de/>, 2014.
- [108] Bosch, *Datenblatt BMP085 Digital pressure sensor*, 2009.
- [109] Datenblatt SRF02, „robotikhardware.de,“ [Online].
- [110] Devantech Ltd, „Robot Electronics,“ 21 04 2015. [Online]. Available: <http://www.robot-electronics.co.uk>. [Zugriff am 21 04 2015].
- [111] Logitech, 2015. [Online]. Available: <http://www.logitech.com/>. [Zugriff am 21 04 2015].
- [112] Robo Peak, „RPLIDAR,“ 21 04 2015. [Online]. Available: <http://rplidar.robopeak.com/>. [Zugriff am 21 04 2015].

- [113] pmdtechnologies gmbH, „pmd CamBoard nano,“ 2014. [Online]. Available: [http://www.pmdtec.com/products\\_services/reference\\_design.php](http://www.pmdtec.com/products_services/reference_design.php). [Zugriff am 21 04 2015].
- [114] Spektrum, 2014. [Online]. Available: <http://www.spektrumrc.com/>. [Zugriff am 21 04 2015].
- [115] Hackers, „Hackers Motor GmbH,“ 2013. [Online]. Available: <http://www.hacker-motor.com/>. [Zugriff am 21 4 2015].
- [116] Hobbywing, „Hobbywing Technology CO., LTD.,“ 2010. [Online]. Available: <http://www.hobbywing.com>. [Zugriff am 21 04 2015].
- [117] Traco Power, „Datenblatt DC/DC Converter,“ 2014.
- [118] G. Welch und G. Bishop, „An Introduction to the Kalman Filter,“ 2006.
- [119] M. S. Grewal, Kalman Filtering, Hoboken, New Jersey: John Wiley & Sons Inc., 2008.
- [120] P. Zarchan, Fundamenals of Kalman Filtering: A Pracitcal Approach, American Institute of Aeronautics and Astronautics, 2005.
- [121] S. Haykin, Kalman Filtering and Neural Networks, McMaster University, Hamilton, Ontario, Canada: John Wiley & Sons, Inc., 2001.
- [122] B. Jähne, Practical Handbook on Image Processing for Scientific Applications, Crc Press, 1997.
- [123] pmdtechnologies gmbH, „<http://www.pmdtec.com>,“ 2014. [Online]. Available: <http://www.pmdtec.com/>. [Zugriff am 08 06 2015].
- [124] T. I. Fossen, Handbook of Marine Craft Hydrodynamics and Motion Control, John Wiley & Sons, 2011.
- [125] Oculus VR LLC, „Oculus,“ 2015. [Online]. Available: <https://www.oculus.com/en-us/>. [Zugriff am 13 07 2015].
- [126] World Viz, 2015. [Online]. Available: <http://www.worldviz.com/>. [Zugriff am 15 08 2015].
- [127] S. v. Borstel, „Firmen klagen über zu viele Fachidioten,“ *Die Welt*, pp. <http://www.welt.de/wirtschaft/article12260743/Firmen-klagen-ueber-zu-viele-Fachidioten.html>, 2011.
- [128] ROBEX, „Robotische Exploration unter Extrembedingungen,“ 30 10 2014. [Online]. Available: <http://www.robex-allianz.de/>. [Zugriff am 30 10 2014].
- [129] D. S. S. W. R. S. Simon Zingg, „MAV Navigation through Indoor Corridors Using Optical Flow,“ in *International Conference om Robotics and Automation*, Anchorage, 2010.

- [130] F. Kendoul, „Optic flow-based vision system for an autonomous 3D localization and control of small aerial vehicles,“ *Robotics and Autonomous Systems*, pp. 591-602, 20 02 2009.
- [131] A. Rav-Acha, „Lucas-Kanade without iterative Warping,“ 2006.
- [132] P. Corke, „An Inertial and Visual Sensing System for a Small Autonomous Helicopter,“ *Journal of Robotic Systems* 21(2), pp. 43-51, 04 11 2004.
- [133] W. Ding, „Adding Optical Flow into the GPS/INS Integration for UAV navigation,“ in *International Global Navigation Satellite Systems Society*, 2009.
- [134] J. Wang, „INTEGRATION OF GPS/INS/VISION SENSORS TO NAVIGATE UNMANNED AERIAL VEHICLES,“ *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2008.
- [135] M. Ahmadi, „Orientation Estimation by Error-State Extended Kalman Filter in Quaternion Vector Space,“ in *SICE Annual Conference*, Kagawa University, Japan, 2007.
- [136] J. E. Bortz, „A New Mathematical Formulation for Strapdown Inertial Navigation,“ *IEE: Transactions on Aerospace and Electronic Systems*, January 1971.
- [137] C. Carraro, „Square Root Kalman Algorithms in Econometrics,“ *Computer Science in Economics and Management* 1, pp. 41-51, 1988.
- [138] J. C. K. Chou, „Quaternion Kinematic and Dynamic Differential Equations,“ *IEEE Transactions on Robotics and Automation*, 1992.
- [139] H. G. Deppner, „Drehratenmessgeber“. *STN Atlas Elektronik*.
- [140] W. E. Hamilton, *Elements of Quaternions*, 1866.
- [141] C. Hide, „Adaptive Kalman Filtering for Low-cost INS/GPS,“ *THE JOURNAL OF NAVIGATION*, pp. 143-152, 2003.
- [142] G. Krishnan, „Micromachined High-Resolution Accelerometers,“ *Journal of the Indian Institute of Science*, pp. 333-361, Jul-Sep 2007.
- [143] H. J. Luinge, *Inertial Sensing of Human Movement*, Enschede, Niederlande: Twente University Press, 2002.
- [144] H. J. Luinge, „Measuring orientation of human body segments using miniature gyscopes and accelerometers,“ *Medical & Biological Engineering & Computing*, pp. 273-282, 2005.
- [145] H. Luo, „A 1 mG LATERAL CMOS-MEMS ACCELEROMETER,“ *IEEE*, pp. 502-507, 2000.
- [146] MEMSense, „MEMSense,“ 2010. [Online]. Available: [www.memsense.com](http://www.memsense.com).

- [147] C. Konvalin, „Compensating for Tilt, Hard Iron and Soft Iron Effects,“ MEMSense, 2008.
- [148] R. v. d. Merwe, Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models, 2004.
- [149] R. Nikoukhah, „Kalman filtering and Riccati equations for descriptor systems,“ 1990.
- [150] D. Roetenberg, „Ambulatory Position and Orientation Tracking Fusing Magnetic and Inertial Sensing,“ *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, pp. 883-890, May 2007.
- [151] S. I. Roumeliotis, „Circumventing Dynamic Modeling: Evaluation of the Error-State Kalman Filter applied to Mobile Robot Localization,“ *International Conference on Robotics & Automation*, May 1999.
- [152] S. Särkkä, „Notes on Quaternions,“ 2007.
- [153] M. St-Pierre, „Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system,“ in *Intelligent Vehicles Symposium*, Parma, Italy, 2004.
- [154] D. B. Sweetser, *Doing Physics with Quaternions*, 2005.
- [155] Y. Tao, „Integration of Vision and Inertial Sensors for 3D Arm Motion Tracking in Home-based Rehabilitation,“ *The International Journal of Robotics Research*, Juni 2007.
- [156] G. B. Greg Welch, „An Introduction to the Kalman Filter,“ 2001.
- [157] J. Wendel, *Integrierte Navigationssysteme*, München: Oldenbourg Verlag München Wien, 2007.
- [158] D. E. Williams, „Complementary Filter to Estimate Inertial Velocity,“ *Journal of Dynamic Systems, Measurement, and Control (ASME)*, 2009.
- [159] O. J. Woodman, „An introduction to inertial navigation,“ University of Cambridge, Cambridge, United Kingdom, 2007.
- [160] M. H. Sadraey, *Aircraft Design: A Systems Engineering Approach*, Wiley, 2012.
- [161] M. Kayton, *Avionics Navigation Systems*, 1997.
- [162] P. Angelov, *Sense and Avoid in UAS Research and Applications*, Wiley, 2012.
- [163] R. Lozano, *Unmanned Aerial Vehicles*, Wiley, 2010.
- [164] N. Michael, „The GRASP Multiple Micro UAV Testbed,“ *IEEE Robotics and Automation Magazine*, 2010.



- [165] J. Sturm, „Visual Navigation for Flying Robots,“ TUM, München, 2012.
- [166] Bronstein, Taschenbuch der Mathematik, Frankfurt am Main: Wissenschaftlicher Verlag Harri Deutsch, 2005.
- [167] L. Wendt, Taschenbuch der Regelungstechnik, Frankfurt am Main: Wissenschaftlicher Verlag Harri Deutsch, 2012.
- [168] D. Kortenkamp, Artificial Intelligence and Mobile Robots, Menlo Park, CA: American Association of Artificial Intelligence, 1998.
- [169] R. Brand, J. Pleban, S. Schön, R. Creutzburg und A. Fischer, „Concept for practical exercises for studying autonomous flying robots in a university environment, Part I,“ in *IS&T/SPIE Electronic Imaging*, San Francisco, 2014.
- [170] D. Kingston und R. Beard, „Autonomous Vehicle Technologies for small fixed wing UAVs,“ 2003.
- [171] P. Ulbrich, „I4Copter: An Adaptable and Modular Quadrotor Platform,“ in *Proceedings of the 26th ACM Symposium on Applied Computing (SAC 2011)*, 2011.
- [172] S. Krause, „Multi-purpose environment awareness approach for single line laser scanner in a small rotorcraft UA,“ *Journal of Intelligent and Robotic Systems*, 2011.
- [173] M. Euston, „A complementary filter for attitude estimation of a fixed-wing UAV,“ in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Nice , 2008.
- [174] T. Krajník, „Simple yet stable bearing-only navigation,“ *Journal of Field Robotics*, pp. 511-533, 27 07 2010.
- [175] G. Baldwin und R. Mahony, „Complementary filter design on the Special Euclidean group  $SE(3)$ “.
- [176] S. Baker und I. Matthews, „Lucas-Kanade 20 Years On: A Unifying Framework,“ *International Journal of Computer Vision*, pp. 221-255, 07 02 2003.
- [177] G. Hoffmann, „Application of Quaternions,“ 2005.
- [178] P. Martin und E. Salaün, „The True Role of Accelerometer Feedback in Quadrotor Control,“ in *International Conference on Robotics and Automation*, Anchorage, Alaska, USA, 2010.
- [179] W. Premerlani und P. Bizard, „Direction Cosine Matrix IMU: Theory,“ 2009.
- [180] J. Escareno, „Nonholomic-like Corridor Navigation of a Quad-rotor MAV using Optical Flow,“ *Universite de Technologie de Compiegne*, Compiegne, 2011.

[181] Trittech, 05 01 2015. [Online]. Available: <http://www.tritech.co.uk>. [Zugriff am 05 01 2015].

## 12 Danksagung

An dieser Stelle möchte ich zunächst allen danken, die bis hierhin durchgehalten haben. Zudem gilt mein Dank allen, die mich bei dieser Arbeit mit Rat und Tat unterstützten.

An erster Stelle gilt mein ausdrücklicher Dank meinem Doktorvater Prof. Montenegro, ohne den diese Arbeit wohl kaum Realität geworden wäre. Zu jeder Zeit war seine Unterstützung und herausragende Arbeitsweise spürbar, die ein ganz besonderes, positives Arbeitsklima schaffte. Er hatte immer einen guten Rat und ein paar motivierende Worte für mich.

Ganz herzlich möchte ich mich bei Prof. Fischer für die hilfreichen Ratschläge und die Unterstützung bedanken. Dank eines sehr freundlichen Umgangs war die Zusammenarbeit überaus angenehm.

Bei Prof. Hennecke möchte ich mich für die Hilfe beim Erstellen des Fragebogens zur Evaluierung der Lehre bedanken.

Dem Universitätsbund möchte ich vielfach Danken für das Vertrauen und die finanzielle Unterstützung durch den Förderpreis und andere Förderungen, die meiner wissenschaftlichen Arbeit zu Gute kamen. Ohne diese finanziellen Mittel wäre die Arbeit in dieser Form nicht möglich gewesen.

Mein ganz herzlicher Dank gilt meiner Familie sowie allen Freunden und Arbeitskollegen, die mich über viele Jahre unterstützt haben. Namentlich möchte ich mich bedanken bei Diana, Angelika, Klaus-Peter, Tobias, Manuel, Daniel, Barbara, Atheel, Erik und Anna.

Nicht zuletzt gilt mein Dank allen Studenten, mit denen ich gemeinsam unseren AQopter18 entwickeln durfte. Mein ganz besonderer Dank gilt Sascha Dechend, Thilo Müller, Michael Strohmeier, Christian Reul, Norbert Schmitt, Paul Benz, Eric Reinthal und Tim Horst für deren ausgezeichnete Mithilfe und begeisterte und begeisternde Unterstützung des Projekts AQopter18.