



Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften  
Department für Informatik

# **Lokalisations- und Regelungsverfahren für einen 4-Rotor-Helikopter.**

Dissertation zur Erlangung des Grades eines  
Doktors der Ingenieurwissenschaften

von

**Dipl.-Inform. Tim Puls**

Gutachter:

**Prof. Dr. Andreas Hein**

**Prof. Dr. Peter Vörsmann**

Tag der Disputation: 26. Juni 2011



## **Abstract**

Zur Zeit gewinnen so genannten Quadrokopter immer mehr an Bedeutung. Sie werden für eine Vielzahl von unterschiedlichen Anwendungsfällen wie z.B. Luftbildarchäologie eingesetzt. In dieser Arbeit werden verschiedene Regelungsverfahren vorgestellt, die notwendig sind, um Rettungskräfte wie Polizei und Feuerwehr bei ihren Einsätzen zu unterstützen. Dazu gehören ein Positions- und Wegpunktregler, ein Trajektorienregler, mit dem man auch Objekte verfolgen kann, und ein Landeargorithmus. Als zweites werden Methoden untersucht, mit denen man die Position und die Höhe eines Quadroopters mit Zentimeter-Genauigkeit lokal bestimmen kann. Untersucht werden sowohl optische als auch Baken basierte Lokalisationssysteme, wobei der Schwerpunkt auf einem Baken basierten System liegt. Optische Systeme haben im Vergleich zu Baken basierten den Nachteil, nicht immer unter allen Sichtbedingungen zu funktionieren, und die Kosten in Bezug auf Gewicht und Rechenaufwand sind relativ hoch. Zwecks genauerer Analysen werden beide Verfahren untersucht und verglichen. Um die Funktionalität der Messmethode in Verbindung mit den Regelalgorithmen nachzuweisen, wird eine autonome Landung eines Quadroopters auf einem fahrenden Fahrzeug durchgeführt.



## Danksagung

Ich möchte meinen Dank an Prof. Dr.-Ing. Andreas Hein für die sehr gute und intensive Betreuung während meiner Doktorarbeit und an Prof. Dr. Peter Vörsmann für die Begutachtung dieser Arbeit aussprechen.

Die vorgestellte Arbeit wurde in der Abteilung für kooperierende und mobile Systeme (CMS) am OFFIS Institut für Informatik unter der Leitung von Matthias Brucke in Kooperation mit der Firma Rheinmetall Defence Electronics GmbH (RDE) durchgeführt. Teile dieser Arbeit wurden durch das go!CART-Forschungskonsortium (go! Competitive Aerial Robot Technologies) gefördert, welches durch Europäische Fördergelder und das Land Bremen finanziert wurde.

Mein besonderer Dank gilt Dr. Reimund Kücke und Matthias Vyshnevskyy von der Firma RDE. Ohne die intensive Zusammenarbeit in den letzten fünf Jahren wäre ich mit meinen Ergebnissen nicht so weit gekommen. Zudem möchte ich meinen Kolleginnen und Kollegen vom OFFIS für die hervorragende Teamarbeit danken. Ich weiß die freundliche, positive und produktive Arbeitsatmosphäre sehr zu schätzen. Vor allem möchte ich mich bei Prof. Dr. Markus Kemper für seine tolle Unterstützung und Vorarbeiten, bei Hannes Winkelmann und Sönke Eilers für die Entwicklung einer ausgezeichneten Software-Infrastruktur und bei Daniel Jelkmann und Sören Schreiner für die Unterstützung bei der Fertigstellung einzelner Hardware-Komponenten und für die vielen Reparaturen bedanken. Des Weiteren bedanke ich mich bei Holger Buss und Ingo Busker von der Firma HiSystems GmbH für die Unterstützung und Bereitstellung einiger Hardwarekomponenten.

Ich möchte zudem meinen Eltern Hans-Adolf und Ingrid Puls für ihre konstante Unterstützung während meines ganzen Lebens danken. Ohne euch wäre ich nicht da, wo ich heute bin. Abschließend will ich mich auch noch bei Nadine und Marianne Kühn bedanken, die sich die Zeit genommen haben, meine Arbeit Korrektur zu lesen.

*Tim Puls*



# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
1.1 Anwendungsfall . . . . .	3
1.2 Problemstellung . . . . .	3
1.3 Gliederung . . . . .	5
<b>2 Stand der Technik</b>	<b>7</b>
2.1 Unbemannte Flugsysteme . . . . .	8
2.2 Quadrokopter . . . . .	9
2.3 Funktionsweise eines Quadroopters . . . . .	10
2.4 Modellierung und Lageregelung eines Quadroopters . . . . .	13
2.4.1 PID-Regler . . . . .	13
2.4.2 LQ-Regler . . . . .	14
2.4.3 Feedback Linearization . . . . .	14
2.4.4 Nicht linerare Regler . . . . .	15
2.4.5 Backstepping Verfahren . . . . .	15
2.4.6 Visuell gestützte Lageregelung . . . . .	16
2.4.7 Fazit . . . . .	16
2.5 Lokalisation . . . . .	17
2.5.1 Globale Navigationssatellitensysteme . . . . .	17
2.5.2 Optische Sensoren . . . . .	17
2.5.3 Tabellarische Übersicht über Lokalisationsverfahren . . . . .	20
2.5.4 Fazit . . . . .	22

2.6	Automatischer Flug . . . . .	22
2.6.1	Positionshaltung . . . . .	22
2.6.2	Trajektorienregelung . . . . .	23
2.6.3	Fazit . . . . .	25
2.7	Übergeordnete Aufgaben . . . . .	26
2.7.1	Kollisionsvermeidung . . . . .	26
2.7.2	Pfadplanung . . . . .	27
2.7.3	Landeverfahren . . . . .	28
2.7.4	Fazit . . . . .	30
<b>3</b>	<b>Eigener Ansatz</b>	<b>33</b>
<b>4</b>	<b>Flugsysteme</b>	<b>35</b>
4.1	Flugsysteme . . . . .	35
4.1.1	Guard M . . . . .	35
4.1.2	Guard S . . . . .	36
4.2	Hardwarekomponenten . . . . .	36
4.2.1	Avionik . . . . .	37
4.2.2	Motorsteller . . . . .	39
4.2.3	Bodenkontrollstation . . . . .	39
4.3	Sensorfusion . . . . .	39
4.3.1	Höhenberechnung . . . . .	39
4.3.2	Positionsberechnung . . . . .	40
<b>5</b>	<b>Verfahren zur Lokalisation im Nahbereich</b>	<b>43</b>
5.1	Optische Messverfahren . . . . .	43
5.2	Lokalisation über Bakensystem . . . . .	45
5.2.1	Messgenauigkeiten . . . . .	46
5.2.2	Positionsberechnung . . . . .	49
5.2.3	Filterung . . . . .	52
5.2.4	Diskussion . . . . .	52

<b>6 Regelalgorithmen</b>	<b>53</b>
6.1 Lageregelung . . . . .	54
6.2 Berechnung von „Bearing“ und „Distance“ . . . . .	55
6.2.1 Kurze Distanzen . . . . .	56
6.2.2 Lange Distanzen . . . . .	57
6.2.3 Unterschiede . . . . .	57
6.3 2D-Wegpunktregler . . . . .	58
6.3.1 Höhenregelung . . . . .	58
6.3.2 Positionsregelung . . . . .	63
6.3.3 Diskussion . . . . .	71
6.4 Trajektorienregler . . . . .	72
6.4.1 Positionsregelung . . . . .	72
6.4.2 Geschwindigkeits- und Beschleunigungsregelung . . . . .	73
6.4.3 Stabilitätsanalyse . . . . .	74
6.4.3.1 Methode nach Lyapunov . . . . .	74
6.4.3.2 Lyapunov-Analyse . . . . .	76
6.4.4 Trajektorienberechnung . . . . .	79
6.4.5 Diskussion . . . . .	80
6.5 Reglerentwurf . . . . .	81
6.6 Objekt (Fahrzeug) verfolgen . . . . .	82
6.7 Automatischer Start . . . . .	83
6.8 Automatische Landung . . . . .	84
6.9 Landung auf fahrendem Fahrzeug . . . . .	86
<b>7 Simulation</b>	<b>89</b>
7.1 Lageregelung inklusive I-Anteil . . . . .	89
7.2 2D-Wegpunktregler . . . . .	90
7.2.1 Höhenregelung . . . . .	90
7.2.2 Positionsaltung . . . . .	90

7.2.3	Streckenflug . . . . .	92
7.2.4	Sollstandlinienregelung . . . . .	93
7.2.5	Simultane Höhen- und Positionsänderung . . . . .	94
7.3	Trajektorienregler . . . . .	96
7.3.1	Rampenfunktionen . . . . .	96
7.3.2	Geozentrischer Kreis . . . . .	98
7.3.3	Höhenregelung . . . . .	101
7.4	Objekt (Fahrzeug) verfolgen . . . . .	102
7.5	Automatischer Start . . . . .	102
7.6	Automatische Landung . . . . .	103
7.7	Landung auf fahrendem Fahrzeug . . . . .	104
<b>8</b>	<b>Experiment</b>	<b>107</b>
8.1	2D-Wegpunktregler . . . . .	107
8.1.1	Höhenregelung . . . . .	107
8.1.2	Streckenflug . . . . .	108
8.1.3	Simultane Höhen- und Positionsänderung . . . . .	109
8.2	Trajektorienregler . . . . .	110
8.2.1	Lineare 3D-Trajektorie . . . . .	111
8.2.2	Geozentrischer Kreis . . . . .	112
8.3	Objekt (Fahrzeug) verfolgen . . . . .	113
8.4	Automatische Landung . . . . .	114
8.5	Landung auf fahrendem Fahrzeug . . . . .	118
<b>9</b>	<b>Zusammenfassung</b>	<b>121</b>
<b>Literaturverzeichnis</b>		<b>125</b>
<b>Abbildungsverzeichnis</b>		<b>141</b>
<b>Tabellenverzeichnis</b>		<b>145</b>

Variablenverzeichnis

147

Abkürzungsverzeichnis

149



# Kapitel 1

## Einleitung

Die Entwicklung von autonomen unbemannten Flugsystemen (UAS, Unmanned Aerial Systems)<sup>1</sup> wurde von Beginn an hauptsächlich vom Militär vorangetrieben. Ziel war es, Gebiete zu überwachen oder aufzuklären, ohne einen Menschen dabei zu gefährden. Laut [Sul07a] und [Sul07b] wurden im Jahre 2010 alleine in Europa rund 300 neue militärische UAS angeschafft, mit steigender Tendenz in den nächsten Jahren. Seit ein paar Jahren entwickelt sich neben dem militärischen auch einen immer größer werdenden zivilen und kommerziellen Markt für UASs. Dabei wird erwartet, dass der zivile Markt den militärischen langfristig übertreffen wird. Im zivilen Bereich werden UAS als Service- und Überwachungssysteme eingesetzt. Die Vielzahl ziviler Anwendungen lässt sich in folgende Kategorien einordnen (vgl. auch [MC99]):

- **Observation und Monitoring:**

Vor allem bei hoheitlichen Anwendungen<sup>2</sup> z.B. durch Polizei, Feuerwehr, Grenzschutz oder Küstenwache ist die Fähigkeit zur Observation und des Monitoring essentiell. Es geht darum, sich vor Ort im Einsatzgebiet schnell einen Überblick über die aktuelle Lage zu verschaffen. Dabei können neben Video- und Fotoaufnahmen auch andere sensorische Daten wie Temperaturen, Luftdrücke, Gaskonzentrationen usw. ermittelt und dem Einsatzteam zur Verfügung gestellt werden.

Neben diesen hoheitlichen Szenarien werden diese Fähigkeiten auch für andere Anwendungen genutzt. Dazu gehören Fotografieren von archäologischen Ausgrabungsstätten, Überwachung von Firmengeländen, Luftaufnahmen zur Erstellung topologischer Karten und viele mehr.

---

<sup>1</sup>veraltet Unmanned Aerial Vehicle (UAV). Heutzutage hat sich der Begriff UAS durchgesetzt, da zu einem unbemannten Flugsystem nicht nur das Luftfahrzeug (UAV) gehört, sondern auch eine Bodenkontrollstation, ein Bediener, eine Funkstrecke und andere Teilsysteme.

<sup>2</sup>laut [Sul07b, S.8] der größte Markt bei zivilen UASs: ca. 30-40 Einheiten pro Jahr ab 2015

- **Inspektion und Wartung:**

Überall da, wo Inspektionen oder Wartungen durchgeführt werden müssen, bei denen der Zugang für den Menschen zu schwierig, kostenaufwändig oder gefährlich ist, wird darüber nachgedacht, UASs einzusetzen. Dazu gehört beispielsweise das Inspizieren von Hochspannungs-, Öl- oder Gasleitungen<sup>3</sup>, von vor der Küste gelegenen Windparks oder auch von durch Hagel beschädigte Agrarfeldern.

- **Kommunikation und Broadcasting:**

Durch UASs können Kommunikationshubs (proxys) schnell und kostengünstig ersetzt oder neu errichtet werden. Für diese Aufgabe werden meistens so genannte HALE (High Altitude Long Endurance) oder MALE (Medium Altitude Long Endurance) Systeme verwendet, die sich durch lange Standzeiten in der Luft auszeichnen.

Im allgemeinen lässt sich zusammenfassen, dass zivile UASs in so genannten „DDD“-Anwendungen (dull, dirty, dangerous) eingesetzt werden. Damit sind zum einen sich immer wiederholende Anwendungen gemeint, die für den Menschen zu „eintönig“ (dull) sind. Die Gefahr bei solchen Aufgaben besteht darin, dass der Mensch diese Aufgaben nach einiger Zeit nicht mehr mit der gleichen Sorgfalt ausführt wie am Anfang. Mit „schmutzig“ (dirty) und „gefährlich“ (dangerous) sind Anwendungen gemeint, die in Umgebungen stattfinden, die für den Menschen z.B. schwer zugänglich oder gesundheitsgefährdend sind.

Zur Zeit gibt es noch einige Hindernisse, die die Verbreitung von UAS im zivilen Markt behindern. Eines der größten Probleme sind die fehlenden gesetzlichen Grundlagen und Regularien für den Betrieb solcher Systeme. Dieses Problem wird in dieser Arbeit nicht adressiert, hingegen das der einfachen Bedienbarkeit. Damit UASs eine große und weite Akzeptanz finden, müssen sie robust und zuverlässig zu bedienen sein. Die Systeme müssen „intelligenter“/„autonomer“ werden, damit sie die an sie gestellten Aufgaben soweit wie möglich eigenständig bewältigen können, ohne dass ein versierter, aufwändig geschulter Bediener nötig ist. „Intelligent“/„Autonom“ heißt in diesem Zusammenhang nicht, dass die Systeme selber entscheiden, welche Missionen, sondern wie sie die im Vorfeld definierten Aufgaben automatisch, korrekt und robust ausführen. In dieser Arbeit werden deshalb Lösungsstrategien vorgestellt, durch die der Grad der Autonomie so erhöht wird, dass komplett Missionen vom Start bis zur Landung eigenständig ausgeführt werden können.

---

<sup>3</sup>ca. 10 Systeme pro Jahr ab 2014 [Sul07b, S.8]

## 1.1 Anwendungsfall

Für viele der oben genannten Missionen bzw. Anwendungsfälle für den zivilen Markt sind so genannte „Vertical TakeOff and Landing“ Systeme (VTOL) am besten geeignet. Das sind Flugsysteme, die die Fähigkeit haben, senkrecht zu starten, über vordefinierte Positionen zu schweben, langsam bestimmte Bereiche abzufliegen und wieder senkrecht zu landen. Diese Eigenschaften sind z.B. für Polizei und Feuerwehr von Bedeutung. Zum einen wird keine aufwendige Infrastruktur zum Starten und Landen benötigt (Landebahn, Katapult usw.) und zum anderen kann man mit solchen Systemen auch in niedriger Höhe in urbanen Gebieten fliegen.

Der Anwendungsfall, der die Grundlage dieser Arbeit darstellt, lässt sich wie folgt beschreiben:

*Sowohl Polizei als auch Feuerwehr sollen mit einem Mini-UAS ausgestattet werden, welches in der Lage ist, vor Ort im Einsatzgebiet bei Bedarf automatisch von einem Fahrzeug aus (z.B. aus einer Box) zu starten, „schnell“ ins Einsatzgebiet zu fliegen, vor Ort vordefinierte Missionen zu erfüllen und am Ende wieder automatisch auf dem Fahrzeug (in der Box) zu landen. Um die Mobilität und Flexibilität der Polizei oder Feuerwehr nicht einzuschränken, sollte das Starten und Landen auch während der Fahrt möglich sein. Zu den vordefinierten Missionen gehören unter anderem das Abfliegen von geozentrischen Kreisen, um z.B. 360°-Aufnahmen von einem Szenario zu bekommen, oder das Verfolgen von Objekten. Eine wichtige Anforderung dabei ist, dass das System zu jeder Tageszeit, also auch nachts, einsetzbar ist und das es unterschiedliche Nutzlasten tragen kann.*

Als VTOL-Konfiguration wurde ein so genannter 4-Rotor-Helikopter oder Quadrokopter gewählt, der zur Kategorie „Drehflügler“ gehört. Ein Quadrokopter zeichnet sich durch einen einfachen robusten mechanischen Aufbau, hohe Agilität und die Möglichkeit aus, dass er ohne größere Probleme miniaturisiert werden kann [Kem07]. Diesen Vorteilen stehen aber auch einige Herausforderungen gegenüber.

## 1.2 Problemstellung

Um den skizzierten Anwendungsfall ausführen zu können, muss der verwendete Quadrokopter Regelungsstrategien besitzen, die es ihm erlauben, an einer festen Position zu schweben und zwischen Wegpunkten in unterschiedlichen Art und Weisen zu manövrieren. Hinzu kommt das Folgen von Objekten, das Abfliegen vordefinierter Trajektorien und das autonome Starten und Landen. Hierzu muss der Quadrokopter seine eigene Position

so genau bestimmen und auch einhalten können, dass das automatische Landen auf einem Fahrzeug oder in einer Box möglich ist. Im folgenden werden zu den notwendigen Fähigkeiten die zu lösenden technischen Herausforderungen aufgelistet.

- **Lagestabilisierung:**

Eine der Hauptherausforderungen ist es, Quadrokopter aufgrund ihrer immanent instabilen Eigenschaft in der Luft zu halten [BMS04]. Im Gegensatz zu eigenstabilen Systemen wie Starrflüglern oder klassischen Helikoptern, muss die Lage eines Quadroopters deshalb permanent geregelt werden.

- **Positionshaltung:**

Darauf aufbauend ist es eine weitere Herausforderung, solche Systeme konstant auf einer Position zu halten [KMF03]. Diese Aufgabe muss sowohl bei konstanten Winden als auch bei temporär auftretenden Windböen funktionieren. Des Weiteren muss der Quadroopter in der Lage sein, unterschiedliche Arten von Lasten wie Sensoren zu tragen, die auch unterschiedlich schwer sein können.

- **Wegpunktnavigation:**

Eine weitere Anforderung ist die Fähigkeit, vordefinierte Wegpunkte anzufliegen. Dabei ist besonders in urbanen Gebieten darauf zu achten, dass das Fluggerät nicht von seinem Kurs abkommt. Des Weiteren muss zugesichert werden können, dass das UAS nicht unter eine vorgegebene Höhe absinkt. Dies kann der Fall sein, sobald sich die Lagewinkel des UAS schlagartig stark ändern. Dies hat zur Folge, dass der maximal zur Verfügung stehende Schub der Motoren nicht mehr ausreichend ist, um die Höhe des UASs konstant zu halten. Zudem können während des Streckenflugs Gegenwinde auftreten, die den angestellten Quadroopter nach unten drücken. Trotz dieser Gefahr ist es aber dennoch erwünscht, auf Grund der begrenzten Flugzeit von Drehflüglern ohne Geschwindigkeitsbegrenzungen zu fliegen. Dies bedeutet, dass neben der Positionsregelung auch eine adäquate Höhenregelung vorhanden sein muss, die eng mit der Positionsregelung zusammenarbeitet.

- **Objektverfolgung / Komplexere Trajektorien abfliegen:**

Sowohl bei der Objektverfolgung als auch beim Abfliegen von komplexen Trajektorien müssen ähnliche Fragestellungen bearbeitet werden. Zu jedem Zeitpunkt muss das UAS den Zustand des zu folgenden Objektes oder den Zustand der Trajektorie annehmen. Neben der Position gehören dazu auch die Geschwindigkeit und die Beschleunigung.

- **Automatischer Start / Automatische Landung:**

Vor allem bei der automatischen Landung an einer vordefinierten Position müssen

Verfahren zur Verfügung stehen, mit Hilfe derer die Position des UASs mit hoher Genauigkeit von wenigen Zentimeter bestimmt werden kann. Aktuelle UASs besitzen zwar GPS-Sensoren (Global Positioning System) für die Navigation, diese sind mit einer Genauigkeit von bis zu  $10m$  für das betrachtete Einsatzszenario aber nicht genau genug [ubl09]. Da das System unabhängig von der Tageszeit betrieben werden soll, muss das Lokalisationsverfahren unabhängig von Lichtverhältnissen funktionieren. Das automatische Landen auf einem Fahrzeug stellt weitere Anforderungen. Neben einem Landeargorithmus muss sich das UAS während des Landevorgangs an das Verhalten des Fahrzeugs anpassen können. Es müssen dazu ähnlich Fragestellungen wie bei der Objektverfolgung bearbeitet werden.

Ein Aspekt, der in dieser Arbeit nicht adressiert wird, ist das Thema Kollisionsvermeidung (auch als „Sense and Avoid“ bezeichnet). In Abschnitt 2.7.1 werden diesbezüglich Arbeiten vorgestellt, die in Zukunft zur Erhöhung der Sicherheit und Autonomie herangezogen werden können. Zur Zeit sind solche Systeme aber nicht robust und zuverlässig genug, so dass Kollisionsvermeidung zur Zeit dadurch gewährleistet wird, die abzufliegenden Trajektorien unter Berücksichtigung bekannter Hindernisse zu planen. Dieses Vorgehen wird auch bei dem Anwendungsfall angenommen.

## 1.3 Gliederung

Die Arbeit gliedert sich in 9 Kapitel. Kapitel 2 gibt einen Überblick über den Stand von Forschung und Technik bezüglich UASs und im speziellen über das in 1.2 skizzierte Einsatzszenario. Es werden verschiedene Verfahren und Algorithmen vorgestellt und auf ihre Anwendbarkeit in Bezug auf den Anwendungsfall überprüft. Im Anschluss wird in Kapitel 3 der eigene Lösungsansatz erläutert. Die für die Arbeit verwendeten Testsysteme bzw. Flugsysteme werden in Kapitel 4 vorgestellt. Zusätzlich werden hier wichtige technische Lösungen (z.B Filterverfahren) erläutert, die für die Funktion des Flugsystems wichtig sind. In Kapitel 5 werden das entwickelte Lokalisationsverfahren zur zentimetergenauen Messung der eigenen Position vorgestellt und mit einem optischen Lokalisationsverfahren verglichen. Die Kapitel 6, 7 und 8 beschäftigen sich mit den entwickelten Regelalgorithmen. Jeder einzelne Algorithmus wird erläutert, mittels Simulationen getestet und experimentell validiert. Am Ende der Arbeit folgt in Kapitel 9 eine Zusammenfassung und ein Ausblick über weitere Arbeiten.



# Kapitel 2

## Stand der Technik

Nach dem Luftverkehrsgesetz (LuftVG §1 Absatz (2) Stand 05.08.2010) werden Luftfahrzeuge aufgeteilt in „1. Flugzeuge[,] 2. Drehflügler[,] 3. Luftschiffe[,] 4. Segelflugzeuge[,] 5. Motorsegler[,] 6. Frei- und Fesselballone[,] 7. Drachen[,] 8. Rettungsfallschirme[,] 9. Flugmodelle[,] 10. Luftsportgeräte [und] 11. sonstige [...] Geräte“, die für den Luftraum bestimmt sind. Dabei wird nicht unterschieden, ob es sich um bemannte oder unbemannte Luftfahrzeuge handelt. In der Literatur gibt es für die Vielzahl unterschiedlicher Arten von Luftfahrzeugen viele unterschiedliche Arten der Klassifizierung. So lassen sich Luftfahrzeuge z.B. nach Flugzeit, Gewicht, Start- bzw. Landeverfahren oder nach der Art der Auftriebsgenerierung einteilen. In Abb. 2.1 ist eine Einteilung unbemannter Luftfahrzeuge nach maximalem Abfluggewicht (MTOW, Maximum Take Off Weight), Einsatzhöhe (Altitude) und Einsatzszenario dargestellt [Rhe10].

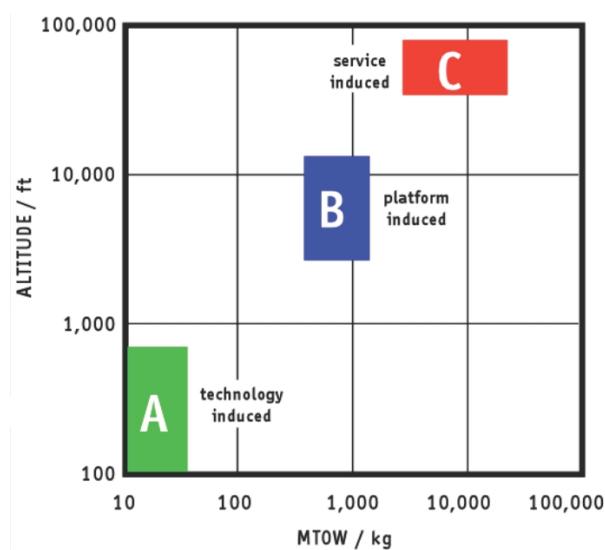


Abb. 2.1: Klassifizierung der Flugsysteme nach Flughöhe (Altitude) und maximalen Abfluggewicht (MTOW, Maximum Take Off Weight) [Rhe10]

Zur Klasse C gehören dabei z.B. geostationäre Luftfahrzeuge mit hohen Abflugmassen, die in großen Höhen ( $> 40,000\text{ft} \approx 10,000\text{m}$ ) operieren. Man spricht in diesem Bereich von den „service induced“ Luftfahrzeugen, die z.B. verwendet werden, um Kommunikationsinfrastrukturen aufzubauen. In der Klasse B werden so genannten „platform induced“ Luftfahrzeuge zusammengefasst. Damit sind sowohl schon existierende militärische UAS gemeint, die für zivile Anwendungen adaptiert, als auch zivile Luftfahrzeuge, die für den autonomen Betrieb nachgerüstet werden. Die Luftfahrzeuge, die für den genannten Anwendungsfällen relevant sind, gehören zur Klasse A mit einem Gewicht  $< 50\text{kg}$ . Unter dieser Klasse werden die technologisch induzierten Luftfahrzeuge zusammengefasst. Auf Grund des geringen Gewichtes handelt es sich hierbei ausschließlich um UAS bzw. Mini-UAS. Im folgenden werden unterschiedliche UAS aus dieser Klasse vorgestellt. Zur weiteren Aufteilung wird eine Klassifizierung der Flugsysteme nach Art der Auftriebsgenerierung vorgenommen.

## 2.1 Unbemannte Flugsysteme

Bei den unbemannten Flugsystemen sind Luftschiffe (Aerostaten), Starrflügler, Schlagflügler und Drehflügler Gegenstand der aktuellen Forschung, wobei Luftschiffe und Schlagflügler zur Zeit eher rein wissenschaftlichen Nutzen haben. Starrflügler und Drehflügler dagegen gibt es in vielen Ausprägungen und werden auch schon für kommerzielle zivile Anwendungsfelder verwendet. [SK08]

- **Luftschiffe / Aerostaten:**

Luftschiffe werden in der Forschung aus zweierlei Gründen verwendet. Zum einen werden wie bei Starrflüglern auch für Luftschiffe verschiedene Regelungsalgorithmen entwickelt, damit sie autonom fliegen können (z.B. [RPHB07], [HO99]). Zum anderen werden Luftschiffe dazu verwendet, um mit ihnen Navigations- oder auch Kollisionsvermeidungsstrategien zu untersuchen (z.B. [iBPV05], [MK06]). Luftschiffe haben den Vorteil, dass sie relativ träge sind, keinen dynamischen Auftrieb erzeugen müssen und unempfindlich gegenüber Kollisionen mit der Umwelt sind. Aus diesen Gründen sind sie daher als Testplattform für solche Aufgaben prädestiniert.

- **Starrflügler:**

Starrflügler sind in der Wissenschaft die zur Zeit am besten untersuchten Flugsysteme. In Arbeiten wie [Bro01] und [RK07] werden Regelungs- und Navigationsprinzipien von klassischen bemannten Starrflächenfliegern gut beschrieben, einschließlich Fragestellungen bezüglich der Wegpunktnavigation und der Sollstandlinienregelung.

Diese Verfahren können auch bei unbemannten Mini-UAS angewendet werden. Dazu gibt es viele Arbeiten, wie z.B. [VSB<sup>+</sup>03], [VSB<sup>+</sup>04], [Kor05], [Kor02], [LZF09], [Mor97], [WBV06] oder [Hol04], die sich mit unterschiedlichsten Regelungsstrategien und Lösungsansätzen bezüglich des unbemannten Fliegens beschäftigen. Auch Fragestellungen bezüglich automatischen Landens ([Sch08]) oder Kollisionsvermeidung ([MMLF10], [LZF09]) werden adressiert. Diese Art von Systemen sind aber nicht für den geschilderten Anwendungsfall geeignet. Des Weiteren können diese Lösungsansätze auf Grund der unterschiedlichen Konfiguration und, da es sich bei einem Quadrokopter um ein unteraktuiertes System handelt, nicht direkt für Quadrokopter angepasst werden [Kem07].

- **Schlagflügler:**

Hauptaugenmerk liegt hier auf der Entwicklung von biologisch inspirierten, insektenähnlichen Flugsystemen. In Arbeiten wie [DSS01] oder auch [HLK09] werden Methoden und Regelungsalgorithmen vorgestellt, wie man zum einen das Flügelschlagen eines Insektes nachbilden kann oder zum anderen, wie man solche Systeme regelt. Diese Systeme werden aber auf Grund ihrer geringen Tragkraft und Flugzeit (< 10min [HLK09]) bisher nicht kommerziell eingesetzt.

- **Drehflügler:**

Ähnlich wie bei den Starrflüglern gibt es auch bei den klassischen Helikoptern eine Vielzahl von Arbeiten, die sich mit den unterschiedlichsten Aspekten wie Regelung (z.B. [APG10]) oder Navigation (z.B. [CGPV08]) beschäftigen. Neben den klassischen Helikoptern gibt es weitere Drehflügler-Systeme, wie so genannte Heckstarter („Tail-Sitter“) [WGLL07] oder auch Koaxial-Helikopter [PSTS97], [HSDM09]. Aber auch Fragestellungen wie Schwarmflug [GL10] oder auch alternative Befehlseingaben [HTB<sup>+</sup>10] werden anhand von Drehflüglern untersucht.

Für kommerzielle Zwecke werden verschiedene Arten von Drehflüglern eingesetzt. Dazu gehören z.B. Systeme wie der Fancopter (EMT), der Camcopter (Schiebel), der Hummingbird (Boeing), der RQ-16A T-Hawk (Honeywell) oder der Cypher (Sikorsky [MC99]).

## 2.2 Quadrokopter

Einen Spezialfall der Drehflügler bilden die so genannten Quadrokopter. Die Entwicklung von Quadrokoptern ist nicht neu. Den ersten Prototyp gab es schon am Anfang des 20.Jahrhunderts mit dem „Gyroplane No. 1“ [Pro], der von Louis Bréguet auf der Vorarbeit von Charles Richet 1907 entwickelt wurde. Aber erst seit der Verfügbarkeit von

kleinen kostengünstigen MEMS-basierten Sensoren und der steigenden Energiedichte von Akkumulatoren in den letzten zehn Jahren rückte dieses Flugprinzip wieder in den Fokus diverser Firmen (wie z.B. AirRobot, Microdrones, Mikrokopter, Ascending Technologies) und Forschungsgruppen (wie z.B. MIT, TU München, Universität Karlsruhe uvm.). Erste Arbeiten bezüglich der Konstruktion und theoretischer Grundlagen von Quadrooptern wurden in [PMHR02] oder [HRW<sup>+</sup>04] beschrieben. Ausgehend von Quadrooptern haben sich weitere Arten von solchen Drehflüglern entwickelt, wie Drehflügler mit sechs, acht oder mehr Propeller. In den Abbildungen 2.2 sind zwei unterschiedliche Konfigurationen zu sehen.



Abb. 2.2: Quadroopterkonfigurationen, links: die MD-200 von Microdrones [Mic10] und rechts: der Folcon 8 von Ascending Technologies [Asc10]

Für die Untersuchungen in dieser Arbeit macht es keinen Unterschied, ob es sich um einen Drehflügler mit vier, sechs, acht oder mehr Propeller handelt. Die Ergebnisse können für sämtliche dieser Konfigurationen verwendet werden.

## 2.3 Funktionsweise eines Quadroopters

Ein Quadroopter besteht aus vier Propeller, die symmetrisch in einem Quadrat angeordnet sind und deren Schubvektoren alle in die gleiche Richtung zeigen.

In Abb. 2.3 ist das mathematische Modell eines Quadroopters zu sehen mit dem inertialen Koordinatensystem im geometrischen Mittelpunkt des System. Die Orientierung des Systems kann durch eine Rotation  $R$  vom körperfesten  $f \rightarrow$  ins geodätische Koordinaten- system  $g$  beschrieben werden [Kem07].

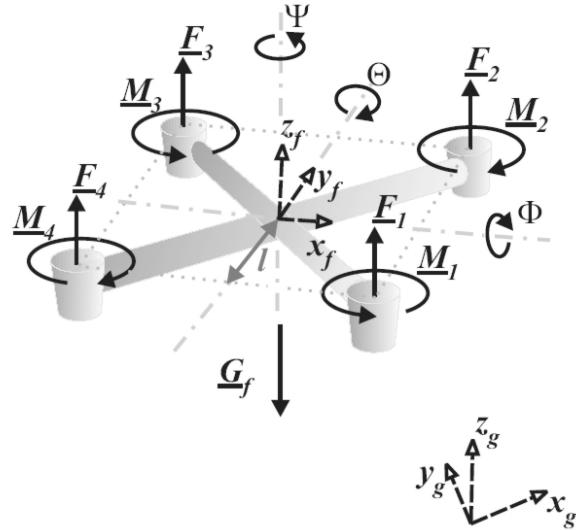


Abb. 2.3: Modell: System mit Kräften, Momenten und Koordinatensystemen [Kem07]

Mit Hilfe dieses Modells wurden die drei Rotations- und die drei Translationsdifferentialgleichungen des Systems aufgestellt. Die folgenden Gleichungen bilden die Grundlage für die Entwicklung der einzelnen Regelalgorithmen und der durchgeführten Simulationen.

$$I_{xx} \cdot \ddot{\phi} = l \cdot (u_2 - c_{D4} \cdot \dot{\phi}^2) - (I_{xx} - I_{yy})\dot{\theta}\dot{\psi} + \dot{\theta} \cdot I_z^{rot} \cdot \Omega_z \quad (2.1)$$

$$I_{yy} \cdot \ddot{\theta} = l \cdot (u_3 - c_{D5} \cdot \dot{\theta}^2) - (I_{yy} - I_{zz})\dot{\psi}\dot{\phi} + \dot{\phi} \cdot I_z^{rot} \cdot \Omega_z \quad (2.2)$$

$$I_{zz} \cdot \ddot{\psi} = u_4 - c_{D5} \cdot \dot{\psi}^2 - (I_{yy} - I_{xx})\dot{\phi}\dot{\theta} \quad (2.3)$$

#### Rotations-Differentialgleichungen

$$m \cdot \ddot{x}_g = u_1 \cdot (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - c_{D1} \cdot \dot{x}_g^2 - (w_y \dot{z}_g - w_z \dot{y}_g) \quad (2.4)$$

$$m \cdot \ddot{y}_g = u_1 \cdot (\cos \phi \sin \theta \cos \psi - \sin \phi \sin \psi) - c_{D2} \cdot \dot{y}_g^2 - (w_z \dot{x}_g - w_x \dot{z}_g) \quad (2.5)$$

$$m \cdot \ddot{z}_g = u_1 \cdot (\cos \phi \cos \theta) - mg - c_{D3} \cdot \dot{z}_g^2 - (w_x \dot{y}_g - w_y \dot{x}_g) \quad (2.6)$$

#### Translations-Differentialgleichungen

wobei  $I$  der Trägheitstensor,  $l$  der Abstand zwischen Rotor und Mittelpunkt,  $m$  das Gewicht des Systems,  $u_i$  die Eingaben und  $c_{Di}$  die Proportionalitätsfaktoren der Widerstandskräfte sind. Für die Translation ist  $c_{Di} = c_w \frac{\rho}{2} A_i$  (in  $kg/m$ ), wobei  $\rho$  die Dichte der Luft (in  $kg/m^3$ ) und  $A_i$  die Querschnittsfläche des UAS (in  $m^2$ ) ist. Der Widerstandsbeiwert  $c_w$  ist dimensionslos. Bei der Rotation hat  $c_{Di}$  die Einheit  $kg \cdot m$ .

In der Abb. 2.3 ist zu erkennen, dass die Drehrichtung zweier gegenüberliegenden Propeller (Motor 2 und 4 im Uhrzeigersinn) jeweils gleich ist, die der anderen beiden entgegengesetzt (Motor 1 und 3 entgegen dem Uhrzeigersinn). Durch diese Tatsache wird das Gesamtdrehmoment der vier Propeller neutralisiert, und ein unkontrolliertes Drehen um die  $\psi$ -Achse vermieden. Selbst wenn sich der Quadrokopter neigt, gleicht sich das Gesamtdrehmoment der Motoren durch diese Konfiguration aus.

Da solche Systeme inhärent instabil sind, müssen sie kontinuierlich geregelt werden. Dieses geschieht alleine über Drehzahländerungen der einzelnen Propeller. Durch die Konfiguration der Propeller können nur vier der sechs vorhandenen Freiheitsgrade direkt beeinflusst werden ( $\phi$ ,  $\theta$ ,  $\psi$  und  $z$ ). Deshalb handelt es sich bei einem Quadrokopter um ein unteraktuiertes System. Das hat zur Folge, dass die beiden Freiheitsgrade in  $x$ - und  $y$ -Richtung nur über eine vorherige Modifikation der  $\theta_d$ - bzw.  $\phi_d$ -Winkel verändert werden können. In Abb. 2.4 ist das Funktionsprinzip eines Quadrokopters anhand der schematische Darstellung einer Reglerstruktur dargestellt.

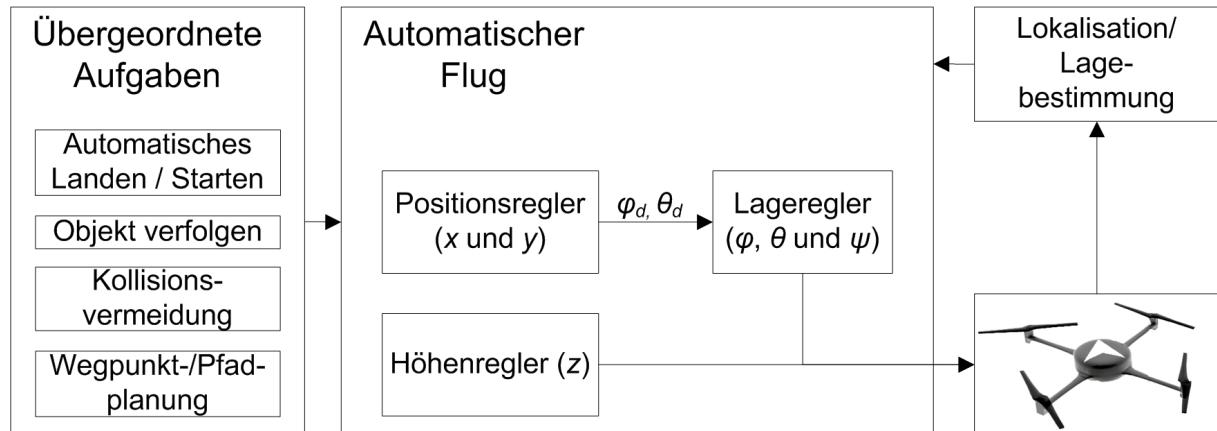


Abb. 2.4: Reglerstruktur eines Quadrokopters: Lageregler (siehe Abschnitt 2.4), Lokalisierung (2.5), Positions-/Höhenregler (2.6.1 und 2.6.2), Kollisionsvermeidung (2.7.1), Pfadplanung (2.7.2), Automatisches Landen (2.7.3)

Ein Quadrokopter ist mit diversen Sensoren ausgestattet, um die aktuelle Pose (Position und Lage) des UASs zu messen und zu berechnen. Dazu gehören drei Drehratensensoren für die Winkel  $\phi$ ,  $\theta$ ,  $\psi$ , drei Beschleunigungssensoren für  $x_f$ ,  $y_f$ ,  $z_f$  und drei Magnetfeldsensoren. Durch diese Sensoren können sowohl die aktuelle Lage als auch die Position/Höhe des UASs zu jedem Zeitpunkt berechnet werden. Zur Berechnung der Lage und Position/Höhe wurden die Arbeiten [Dor] und [SPKR06] als Grundlage herangezogen. Zur Berechnung der Position bzw. Höhe müssen die Sensorwerte der körperfesten Beschleunigungssensoren in das geodätische Koordinatensystem überführt und dann zweimal integriert werden. Auf Grund der inhärenten Ungenauigkeit und der vorhandenen Drift der

inertialen Sensoren ist es notwendig, zusätzliche komplementäre Sensorik zu verwenden, um die Berechnung der Höhe und der Position zu stützen (siehe Abschnitt 4.3).

Zur Berechnung der Lage wird auf die Standardverfahren aus [Dor] und [SPKR06] zurückgegriffen und ist deshalb nicht Inhalt dieser Arbeit.

## 2.4 Modellierung und Lageregelung eines Quadrokopfers

Es gibt viele Veröffentlichungen bezüglich der Modellierung und Lageregelung von Quadrooptern. In Bezug auf die Modellierung sind unter anderem die Arbeiten [BMS04], [Kem07], [McK04], [HRW<sup>+</sup>04], [PMHR02], [PMC], [DMB06], [YLW05] zu nennen. In den meisten Arbeiten wird das dynamische Verhalten des Quadroopters mit Hilfe von Differentialgleichungen beschrieben. Die Grundstruktur sieht bei allen relativ ähnlich aus (vgl. auch Abschnitt 2.3). Sie unterscheiden sich aber in ihrer Komplexität. So wird z.B. in manchen Arbeiten die Korioliskraft berücksichtigt [Kem07], in anderen die Verschiebung des Schwerpunktes [Kem07], [McK04] des Systems. Auch Fragestellungen bzgl. der Blattelementetheorien [Kem07], [PMC], [McK04] oder des Ansprechverhaltens von Motoren [PMHR02] werden berücksichtigt.

Aber die Themen Modellierung und Regelung sind nicht voneinander zu trennen, da im Regelfall der Regler anhand eines entwickelten Modells hergeleitet wird. Zu nennen ist da vor allem die Arbeit [Bou07], in der verschiedene Modellierungsansätze (z.B. Euler-Lagrange Formalismen oder Newton-Euler Formalismen) und Regelungsansätze (PID-Regler, LQ-Regler, Backstepping-Regler, nicht lineare Regler u.a.) miteinander verglichen werden.

Die verschiedenen Regelungsansätze sollen im folgenden vorgestellt werden.

### 2.4.1 PID-Regler

In [BNS04], [AOT05], [GSGA09] und [HHWT09] werden unter anderem PID-Regler verwendet, um die Lage des Quadroopters zu regeln. PID-Regler gehören zu den analogen bzw. stetigen Reglern, bei denen die Stellgröße innerhalb eines Stellbereiches jeden Wert annehmen kann. So wird z.B. in [AOT05] der Einsatz eines PD-Reglers sowohl für die Lage als auch für die Position beschrieben. Es konnte nachgewiesen werden, dass die Lage und die Position mit einem PD-Regler „gut“ geregelt werden können. In [HHWT09], [GSGA09] und [BNS04] wurde der Einsatz eines PID-Reglers zur Regelung der Lage und der Höhe untersucht. In allen Arbeiten konnte sowohl die Lage als auch die Höhe erfolgreich geregelt werden.

Ein PID-Regler kann sowohl durch Hardware mittels Operationsverstärker als auch durch Software mittels mathematischer Formeln diskret realisiert werden. In Gl. (2.7) ist der digitale Regelalgorithmus eines PID-Reglers dargestellt.

$$u(kT) = k_R \cdot e(kT) + \frac{T}{T_1} \cdot \sum_{i=1}^k e(iT) + \frac{TD}{T} \cdot [e(kT) - e([k-1]T)] \quad (2.7)$$

wobei  $u(kT)$  der Reglerausgang und  $e(kT)$  die Regeldifferenz zum Zeitpunkt  $kT$  ist. Dabei ist  $T$  der zeitliche Abstand zwischen zwei Berechnungen und  $k$  der aktuelle Berechnungsschritt. Durch die drei Freiheitsgrade  $k_R$  (P-Anteil),  $T_1$  (I-Anteil) und  $TD$  (D-Anteil) kann das Reglerverhalten eingestellt werden [Fac97, Seite 570f].

### 2.4.2 LQ-Regler

In [BNS04] wird für einen Quadrokopter neben einem PID-Regler auch ein LQ-Regler beschrieben, und die beiden unterschiedlichen Regelungsansätze werden diskutiert. Die Ergebnisse waren folgende:

- **PID:** Die klassische Regelung mit PD-Reglern funktioniert bei kleinen Störungen gut. Unter Hinzunahme eines integralen Anteils konnten auch permanente Regelfehler infolge von äußeren Störeinflüssen oder Asymmetrien des Systems ausgeglichen werden.
- **LQ:** Unter der Annahme, dass es sich um ein exakt symmetrisches System handelt und keine äußeren Störungen auftreten, funktioniert diese Regelung gut. Zwei unterschiedliche Methoden wurden untersucht, die Pearson- und die Sage-Eisenberg-Methode. Simulationen haben ergeben, dass die Sage-Eisenberg-Methode besser funktioniert. Trotzdem waren beide Ergebnisse im Vergleich zum PID-Regler langsamer und weniger dynamisch.

### 2.4.3 Feedback Linearization

Die „Feedback Linearization“ oder auch „Exakte Linearisierung“, wie sie z.B. in [MBM01], [MBD06], [MB04], [AOT05] oder [AS06] beschrieben wird, beruht dabei auf differentialgeometrischen Methoden. Sie ermöglicht es, für eine beachtliche Menge von nichtlinearen Systemen eine geeignete Zustandstransformation (Diffeomorphismus) und Zustandsrückführung anzugeben, um ein lineares System in neuen Koordinaten zu erhalten. Die Linearisierung kann dabei den gesamten Zustandsraum umfassen. Die Exakte Linearisierung hat

aber bedeutende Nachteile. Nichtlinearitäten werden aus dem System entfernt, auch wenn sie einen stabilisierenden oder destabilisierenden Einfluss auf das System haben. Das heißt, dass das linearisierte System nicht mehr korrekt das eigentliche System widerspiegelt.

Das in [AOM02] verwendete „Feedback Linearization“-Verfahren geht wie folgt vor. Zuerst werden die einzelnen Bewegungen des Drehflüglers aufgeteilt in Schweben, Starten, Landen, Links-/Rechts-Fliegen, Nicken, Rollen, Gieren und so weiter. Ein High-Level-Controller muss die einzelnen Aufgaben so kombinieren, so dass die gewünschte Zielposition erreicht wird. Für die einzelnen Bewegungen wurde festgestellt, dass die exakte Eingangs-Ausgangs-Linearisierung durchgeführt werden kann.

#### 2.4.4 Nicht linerare Regler

Bei der Entwicklung von nicht linearen Reglern wird im Allgemeinen versucht, mit Hilfe einer Lyapunov-Funktion  $V$ , die das Energieniveau des Systems beschreibt, die globale asymptotische oder die Lyapunov Stabilität [Kem07] des entwickelten Reglers nachzuweisen (siehe auch Abschnitt 6.4.3.1). Solche Reglerentwürfe werden z.B. in [Kem07], [BMS05], [BMS04], [CDL04], [GMLD08] und [Voo09a] durchgeführt. Die Arbeiten unterscheiden sich dabei in der Komplexität der verwendeten Differentialgleichungen, durch die das dynamische Verhalten des Systems beschrieben wird und der Wahl einer geeigneten Lyapunov-Funktion  $V$ . Gemeinsam haben sie, dass die Stabilität der entwickelten Regler nachgewiesen werden kann.

#### 2.4.5 Backstepping Verfahren

Das rekursive „Backstepping“-Verfahren (vgl. [Kno00]) gehört zu den nicht linearen Reglern. Es legt in jedem Entwurfsabschnitt Eigenschaften des Systems fest. Es werden jeweils Regelgesetze für ein Teilsystem erschaffen. Wird das Regelziel erreicht, wird das Teilsystem bei jedem Schritt vergrößert, bis sich ein Regelgesetz für das Gesamtsystem ergibt. „Backstepping“ zählt zu den Lyapunov-basierten Entwurfsverfahren. [Vie]. Dieses Verfahren wird z.B. in den Arbeiten von [BS05], [BBT07], [AOT05] oder [AGMD07] untersucht.

In [AOM02] wird ein Vergleich zwischen „Feedback Linearization“ und „Backstepping“ durchgeführt. Das vorgestellte „Backstepping“-Verfahren wurde in dieser Arbeit auf zwei Bewegungen angewandt. Dieses Verfahren ist sinnvoll, wenn die Änderung eines Systemzustandes eine Änderung eines anderen Systemzustandes voraussetzt. Im Fall des Quadrokopters geht der Positionsänderung ( $x_g$ -,  $y_g$ -Position) immer eine Änderung der Lagewinkel<sup>1</sup>

---

<sup>1</sup>In [AOM02] sind die Winkelbezeichnungen  $\phi$  und  $\theta$  vertauscht

$\phi$  und  $\theta$  voraus. Es wurde angenommen, dass sich der Gierwinkel nur marginal ändert, damit bestimmte Terme der Bewegungsdifferentialgleichungen entfallen können und somit der Regelungsalgorithmus einfacher wird. Das Ergebnis war, dass beim „Backstepping“-Verfahren die gewünschten Sollwerte schneller eingenommen werden konnten als bei der „Feedback Linearization“.

#### 2.4.6 Visuell gestützte Lageregelung

[SWT04] beschreibt das Design eines autonomen Quadroopters, das auf dem Einsatz von Kameras basiert. Die Positions- und Lagebestimmung wird durch eine MEMS-IMU durchgeführt und mit Kameradaten kombiniert. Die „Onboard“-Kamera nimmt dazu ständig Bilder vom Boden, auf dem verschieden kodierte Markierungen angebracht sind, auf und sendet diese mit den IMU-Daten zusammen zu einer Bodenstation. Dort werden die Bilder verarbeitet und anschließend aus den Bild- und IMU-Daten die Lage und Position bestimmt. Um die beiden Sensordaten miteinander zu kombinieren, kommt ein Kalman-Filter zum Einsatz. Nachdem die Bodenstation den Regelalgorithmus abgearbeitet hat, sendet sie über Funk die neuen Stellwerte zum Quadroopter. Die Marker können 256 verschiedene Informationen enthalten und somit voneinander unterschieden werden. Die Marker dienen zum einen dazu, die Position sowie die Lage des UAs zu bestimmen, und zum anderen, um online eine Karte der Umgebung zu erstellen. Messungen haben ergeben, dass die Standardabweichung der Lage in Bezug auf den Roll- und Nickwinkel ungefähr  $4^{\circ}$ - $4,5^{\circ}$  und des Gierwinkels etwa  $1.25^{\circ}$  beträgt. Die  $x_g$ - beziehungsweise  $y_g$ -Position konnte auf circa  $0.02 - 0.03m$  genau eingehalten werden.

#### 2.4.7 Fazit

In dem Bereich der Modellierung und der Lageregelung gibt es eine Vielzahl von Arbeiten und Ergebnisse. Aber fast alle vorgestellten Lageregler können konstante Lageabweichungen infolge von Asymmetrien oder Störungen nicht ausgleichen, da notwendige integrale Regelanteile fehlen. Selbst bei den vorgestellten PID-Reglern wird entweder der Parameter für den I-Anteil auf 0 gesetzt (vgl. [GSGA09]) oder wird nicht weiter erwähnt. Aus diesem Grund beschäftigt sich diese Arbeit mit diesem Aspekt (siehe Abschnitt: 6.1). Es wird ein Lageregler benötigt, der robust ist, schnell auf Störungen und Sollwinkeländerungen reagiert, konstante Regelabweichungen ausgleicht und global stabil ist (siehe Kapitel 3).

## 2.5 Lokalisation

Eine verlässliche Lokalisation stellt die Grundlage jeglicher darauf aufbauenden Navigationsaufgaben dar. Ziel ist es, driftfreie Positionsdaten mit einer möglichst hohen Aktualisierungsrate und einer hohen Genauigkeit zu bekommen. In der Literatur werden dabei hauptsächlich GPS-basierte (für „Outdoor“-Anwendungen) und optische (meistens für Indoor-Anwendungen) Methoden beschrieben, von denen im folgenden exemplarisch einige vorgestellt werden.

### 2.5.1 Globale Navigationssatellitensysteme

Globale Navigationssatellitensysteme (GNSS, Global Navigation Satellite System) sind Lokalisationssysteme, mit deren Hilfe Positionen auf der Erde mit einer Genauigkeit von ca.  $10m$  (beim GPS) bestimmt werden können. Zu den GNSS gehören Systeme wie das Amerikanische NAVSTAR (GPS), das Europäische Galileo oder auch das Russische GLONASS, wobei zur Zeit fast ausschließlich das GPS eingesetzt wird. In [Wen07] und [ubl07] werden Verfahren vorgestellt, mit denen aus GPS-Signalen präzise geodätische Positionen berechnet werden können. Diese Positionsdaten werden dann verwendet, um die durch Inertialsensoren (vgl. [Lea99]) berechnete Position zu korrigieren. In [Wen07] oder [GW01] werden solche Verfahren, die auf der Verwendung von Kalman-Filtern beruhen, vorgestellt. Ziel dieser Filterungen ist es, die Eigenschaften der hohen Aktualisierungsraten von IMUs und die der Langzeitstabilität und der Genauigkeit von GPS-Sensoren zu vereinen. Bei den vorgestellten Filterverfahren wird ein Aspekt aber nicht betrachtet. Das ist die zeitliche Verzögerung der GPS-Signale, die bei der Positionsrechnung bei sich bewegenden Objekten zu Fehlern führt. Deshalb wird in dieser Arbeit dieser Aspekt betrachtet (siehe Abschnitt 4.3.2).

### 2.5.2 Optische Sensoren

Bei optischen Verfahren gibt es den Nachteil, dass viele Algorithmen zur Zeit so rechenintensiv sind, dass sie nicht ohne weiteres an Bord eines Mini-UASs berechnet werden können. Aus diesem Grund ist es üblich, die Bildverarbeitungssoftware auf einem externen PC laufen zu lassen (z.B. [TVH06], [MCM009], [HRHM08], [BF07], [EVDF10], [CAM<sup>+</sup>09], [RGCF10]). Um diesen Nachteil kompensieren, gibt es zwei verschiedene Ansätze. Zum einen werden zur Zeit immer kleinere und leistungsfähigere Hardwareplattformen entwickelt, wie z.B. [LLF<sup>+</sup>06], [tas10] oder [gum10]. Der zweite Ansatz ist, Algorithmen soweit zu vereinfachen, dass sie durch leistungsschwache  $\mu$ -Prozessoren berechnet werden können (vgl. [WRZ10]). Im folgenden werden einige optische Verfahren vorgestellt.

In [CAM<sup>+</sup>09] wird ein Verfahren mit nur einer Kamera beschrieben. Ziel ist es, die Höhe des UASs aus der Abfolge von Videobildern zu bestimmen. Die Grundidee des Algorithmus ist es, die Zuordnung zwischen den Texturen in einem Bild und einer möglichen Höhe zu erlernen. Der vorgestellte Algorithmus war in der Lage, die Höhe über die Flugzeit mit einer durchschnittlichen Genauigkeit von  $0.35 - 0.50m$  zu berechnen. Der maximale Fehler betrug bis zu  $1.5m$ .

[RGCF10] beschreibt einen Algorithmus, der neben der Höhe auch die Position und die Geschwindigkeit eines UASs mit Hilfe einer nach unten schauenden Kamera bestimmt. Voraussetzung für den Algorithmus ist ein wohldefinierter strukturierter Untergrund (Richtungsfahrbahn), an dem sich das UAS ausrichten kann. Der Bildverarbeitungsalgorithmus und der vorgestellte Regler waren in der Lage, mit einer vorgegebenen Geschwindigkeit entlang der Fahrbahn zu fliegen.

In [AAL<sup>+</sup>09] wird ein Verfahren beschrieben, mit dessen Hilfe die durch eine IMU ermittelten Positionsdaten durch optischen Fluss gestützt werden. Als experimentelle Plattform wurde ein klassischer Helikopter mit einer Nadirkamera als optischen Sensor verwendet. Mit Hilfe des optischen Fluxes konnte die Drift der IMU-Daten weitestgehend kompensiert werden. Nach einem Testflug, währenddessen der Helikopter ein Viereck abfliegen musste, betrug der Positionsfehler am Ende  $1 - 2m$ .

In [TVH06] wird ein Positionsregler implementiert und getestet, der auch auf Bildverarbeitung über eine Nadir-Kamera basiert. Zur Bestimmung der aktuellen Position wurde ein speziell angefertigtes Muster („Moiré target“) verwendet. Der Quadrokopter war in der Lage, die Position mit einer Genauigkeit von einem Meter zu halten.

Ein weiteres kamerabasiertes Navigationsverfahren wird in [KN09] beschrieben. Es wird ein relatives optisches Navigationssystem als alternative Methode zur Bestimmung der Position vorgestellt, welches auf „feature tracking“ und „path integration“ beruht. Als Kamerasystem wird eine einzelne Nadir-Kamera verwendet und die Bildverarbeitung auf einer Bodenstation durchgeführt. Um die Genauigkeit und Robustheit des Verfahrens zu erhöhen, werden die berechneten Daten mit einer IMU und einem Drucksensor gefiltert. Mit Hilfe dieses Verfahrens konnte unter anderem ein kleines Fahrzeug (ca.  $0.5 \times 1m$ )  $20m$  weit verfolgt werden mit einer maximalen Abweichung von  $\pm 1m$ . Des Weiteren konnte ein automatischer Start mit anschließender automatischer Landung durchgeführt werden. Die Genauigkeit der Landung lag bei  $0.25m$ . Bei dem Verfahren hat sich herausgestellt, dass sich der Messfehler in geringen Höhen erhöht.

Ein ähnliches Verfahren wird in [SKPN09] vorgestellt. Auch hier wird ein optisches Navigationsverfahren beschrieben, welches sowohl Farben als auch „features“ erkennen kann. Mit diesem Verfahren war es möglich, für mehr als  $8min$  über einem vorher definierten

Ziel zu schweben. Für die Experimente wurde ein nichtlinearer Regler verwendet, der in [KFL08] beschrieben wird. Die Genauigkeit bei der Positionshaltung lag bei  $1 - 2m$ . Die Geschwindigkeit könnte mit  $\pm 0.5m/s$  gehalten werden. Des Weiteren konnten die optisch ermittelten Positionsdaten mit den GPS-basierten Daten kombiniert werden, um die Robustheit zu erhöhen.

Ein Verfahren, dass auf optischem Fluss basiert, wird in [KFN09] vorgestellt. Das System, dass aus einer nach unten gerichteten Kamera besteht, ist sowohl für den Einsatz im Freien als auch im Inneren von Gebäuden geeignet. Voraussetzung ist eine strukturierte Oberfläche und passende Lichtverhältnisse. Experimente haben gezeigt, dass die Position nach 200s Flugzeit mit einer Genauigkeit von  $\pm 3m$  gehalten werden konnte.

In [CMGM09] wird ein Framework für optische Navigationsverfahren vorgestellt, bei der die Kamera nicht nach unten gerichtet ist sondern in Flugrichtung. Ziel ist es, die Umgebung des UASs anhand von natürlichen Landmarken zu ermitteln, um sich einen Pfad durch die Umgebung zu suchen und abzufliegen. Sowohl die Bildverarbeitung als auch die Positionsregelung wird auf einem externen Rechner durchgeführt.

Neben Arbeiten, in denen eine Kamera verwendet wird, gibt es auch Arbeiten, die sich auf stereoskopische Kamerasysteme konzentrieren. In [EVDF10] wird z.B. ein hybrides Nadir-Stereokamerasystem vorgestellt, das in der Lage ist, sowohl die Höhe als auch die Lage eines UASs zu ermitteln. Zudem kann mit diesem Verfahren eine Segmentierung des Untergrundes vorgenommen werden. Dadurch können im Bild ebene flache Bereiche von hervorstehenden Objekten unterschieden werden. Im Vergleich zu Lasermessungen betrug der Fehler der Messmethode 2.41%. Zusätzlich wurde in der Arbeit untersucht, inwieweit man den entwickelten Algorithmus auf einen  $\mu$ -Prozessor portieren kann. Die Ergebnisse haben gezeigt, dass die erzielte Aktualisierungsrate von  $5Hz$  zu niedrig ist, um damit navigieren zu können.

In [JTB<sup>+</sup>09] wird ein weiteres stereoskopisches Kamerasystem beschrieben. Es wird ein Verfahren vorgestellt, mit dem ein UAS (hier ein Starrflügler) Geländestrukturen, wie z.B. Schluchten, folgen und dabei Hindernisse erkennen kann. Zu diesem Zweck wird ein spezielles Spiegelsystem eingesetzt, welches die Bilder in ein zylinderförmiges Koordinatensystem projiziert, um die Berechnungen zu vereinfachen und schneller zu machen. Mit Hilfe von Disparitäten ist es nun möglich, sowohl Pfade durch Geländestrukturen zu berechnen als auch Hindernisse zu erkennen.

Ein weiteres optisches Navigationssystem wird in [ABH<sup>+</sup>09] vorgestellt. Neben einem Stereokamerasystem wird hier ein Lasersensor verwendet, um die Bewegungsrichtungen eines UAS zu bestimmen. Während eines ein minütigen Fluges lag die durchschnittliche Positionsabweichung während des Fluges bei  $0.015m$  und die der Geschwindigkeit bei

$0.025m/s$ . Eines der Hauptprobleme hier ist das Gewicht<sup>2</sup> des Systems, welches für Mini-UASs (< 1kg) zu groß ist, um längere Flugzeiten zu erlauben.

### 2.5.3 Tabellarische Übersicht über Lokalisationsverfahren

Die Tabelle 2.1 gibt einen Überblick über verschiedene Lokalisationsverfahren. Es wird eine Einschätzung ihrer Vor- und Nachteile in Bezug auf den Anwendungsfall (siehe 1.1) gegeben.

Tab. 2.1: Übersicht Lokalisationsverfahren

Messprinzip	Eigenschaften / Vor- & Nachteile	Fazit
GPS	<ul style="list-style-type: none"> <li>+ unbegrenzte Reichweite</li> <li>+ geringer Energieverbrauch</li> <li>+ lichtunabhängig</li> <li>- ungenügende Messgenauigkeit: &lt;10m</li> <li>- Störungen durch Mehrwegeausbreitungen und Abschattungen</li> <li>- geringe Updaterate: 5Hz</li> </ul>	<ul style="list-style-type: none"> <li>• für Feinpositionierung zu ungenau</li> <li>• für Grobpositionierung geeignet, z.B. für Wegpunkte abfliegen</li> </ul>
optisch	<ul style="list-style-type: none"> <li>+ hohe Messgenauigkeit: abhängig von Auflösung</li> <li>+ sehr hohe Updateraten möglich</li> <li>- stark lichtabhängig</li> <li>- Reichweite ist abhängig vom Objektiv</li> <li>- Ziel muss immer im Öffnungswinkel der Kamera liegen</li> <li>- es handelt sich oft um relative Lokalisationsverfahren (z.B. beim optischen Fluss)</li> <li>- rechenaufwändige Berechnungen</li> </ul>	<ul style="list-style-type: none"> <li>• für Feinpositionierung im Prinzip geeignet, aber wegen der Lichtabhängigkeit und der Tatsache, dass das Ziel immer im Öffnungswinkel der Kamera sein muss nicht die optimale Lösung!</li> </ul>

---

<sup>2</sup>nicht näher spezifiziert. Der Lasersensor alleine wiegt ca. 160g

Messprinzip	Eigenschaften / Vor- & Nachteile	Fazit
Funklaufzeit	+ ausreichende Reichweite: <100m + geringer Energieverbrauch + lichtunabhängig - ungenügende Messgenauigkeit: <1m - geringe Updaterate: 15Hz	• für Feinpositionierung ungeeignet • wegen großer Reichweite als Übergangsmessprinzip zwischen GPS und USS geeignet
Ultraschall	+ hohe Messgenauigkeit: 0.02-0.03m + geringer Energieverbrauch + lichtunabhängig - geringe Reichweite: <10m - geringe Updaterate: 15Hz - Störungen durch äußere USS-Signale möglich	• für Feinpositionierung gut geeignet • wegen geringer Reichweite als alleiniger Sensor zur Landung ungeeignet
Lasermessung <sup>3</sup>	+ hohe Messgenauigkeit: <0.05m - hoher Energieverbrauch: <12W - geringe Reichweite: <30m - Updaterate von Untergrundfarbe abhängig: <40Hz - Sonnenlicht beeinflusst die Genauigkeit des Sensors - hohe Anschaffungskosten: >2000€	• für Feinpositionierung geeignet • Sensoren sind aber zu teuer, zu schwer und haben einen zu hohen Energieverbrauch

Neben diesen Messverfahren gibt es noch weitere Sensoren, die in dieser Arbeit nicht näher betrachtet werden. Dazu gehören z.B. Radarsensoren, Tachymeter, stationäre Laserscanner oder auch Multifrequenz-GPS-Empfänger: Diese Sensoren sind entweder zu schwer, haben zu große Abmaße oder einen zu hohen Energieverbrauch oder sind einfach zu teuer für den Anwendungsfall.

---

<sup>3</sup>in diesem Fall kleine Laserscanner von [Hok] (UTM-30LX)

## 2.5.4 Fazit

Es gibt eine Vielzahl von optischen Methoden zur Bestimmung der aktuellen Position eines UASs. Unter definierten Bedingungen können mit optischen Methoden Messgenauigkeiten von wenigen Zentimetern erreicht werden. Diese Methoden sind aber alle abhängig von geeigneten Lichtverhältnissen und anderen Einflüssen (siehe Abschnitt 5.1). Des Weiteren sind optische Verfahren fast alle so rechenaufwändig, dass sie auf externen Computern berechnet werden müssen. Das führt dazu, dass auch breitbandige Übertragungsverfahren mit großer Reichweite ( $> 1 - 2\text{ km}$ ) zu Verfügung stehen müssen. Es fehlen deshalb leichte und zentimetergenaue Lokalisationssysteme für nächtliche Einsätze, die für kleine unbemannte Flugsysteme eingesetzt werden können. Deshalb wird in Abschnitt 5.2 ein neues Lokalisationsverfahren vorgestellt, dass unabhängig von Lichtverhältnissen funktioniert, ein geringes Gewicht aufweist und wenig Rechenzeit benötigt (siehe Kapitel 3).

# 2.6 Automatischer Flug

## 2.6.1 Positionshaltung

Neben der Lokalisation ist die Fähigkeit der Positionshaltung (Hovering) die nächste wichtige Grundvoraussetzung, um zwischen Wegpunkten zu navigieren. Eine der ersten Arbeiten bzgl. Positionshaltung mit und ohne Unterstützung von GPS-Signalen wird in [Wen07] vorgestellt.

In [AOT05] wird neben den PD-Lagereglern auch ein PD-Regler für die Positionshaltung vorgestellt. In Simulationen konnte die Position „gut“ geregelt werden und Positionsänderungen ohne Überschwingen durchgeführt werden. Da in diesem Regler kein integraler Anteil vorhanden ist, können aber Störungen nicht ausgeglichen werden. Zu diesem Zweck wird in [HHWT07] ein PID-Regler vorgestellt. Experimentelle Versuche haben ergeben, dass der Regler in der Lage ist, „Indoor“ die Position eines Quadroopters innerhalb von  $0.4\text{ m}$  zu halten. Zur Bestimmung der Position wurde eine Deckenkamera verwendet. Die Autoren hatten aber das Problem, dass auch der PID-Regler nur schlecht in der Lage war, z.B. Störungen in Folge von Wind auszugleichen.

Neben den stetigen PID-Reglern wurden auch andere Strategien untersucht. So werden z.B. in [HMC02] und [HM04] optische Verfahren zur Regelung der Position beschrieben. Der in [HMC02] entwickelte Positionsregler, der auf „Backstepping“-Techniken beruht, konnte erfolgreich in Simulationen getestet werden. Der Regler war in der Lage, einen simulierten Kreis abzufliegen. In [HM04] wird ein rein optisches Verfahren zur Regelung

von unteraktuierten Systemen vorgestellt. Dieses Verfahren ist nicht reduziert auf Quadrokopter, sondern ist in der Lage, in der Simulation jegliches unteraktuierte Flugsystem zu regeln.

Weitere Regelverfahren werden in [WHJT05] beschrieben. Hier werden zum einen zwei unterschiedliche Strategien zur Höhenregelung vorgestellt, die „Integral Sliding Mode“ und die „Reinforcement Learning“-Regelung. Des Weiteren wird ein integraler LQ-Regler beschrieben, um die Position eines Quadroopters zu stabilisieren. Die maximale Abweichung während eines zweiminütigen Fluges lag bei  $3m$ .

Der LQ-Regler wurde unter anderem auch in [CLD05] untersucht. Der klassische LQ-Regler ohne integralen Anteil wurde mit einem nicht linearen Regelalgorithmus verglichen, der auf „nested saturation control“-Strategien basiert und durch die Lagrange-Methode hergeleitet wurde. Der LQ-Regler konnte hier im Gegensatz zu [WHJT05] keine erfolgreichen Ergebnisse liefern. Der nicht lineare Regler konnte in Experimenten aber erfolgreich getestet werden, so dass die Autoren diesen Regler präferieren.

[YLW05] beschäftigt sich mit der Lösung eines „time-optimal control“-Problems (TOMP) zur Regelung der Position. Um das nicht lineare Optimierungsproblem zu lösen, wird eine initiale mögliche Lösung benötigt, die mittels genetischer Algorithmen gesucht wird. In Simulationen konnten kleinste Positionsänderungen (von  $0.5m$ ) so in optimaler Zeit durchgeführt werden.

## 2.6.2 Trajektorienregelung

Die Trajektorienregelung hat die Aufgabe, die durch die Pfadplanung (siehe Abschnitt 2.7.2) erstellten Pfade bzw. Trajektorien abzufliegen. Dazu gibt es verschiedene Methoden. So werden in [MMW<sup>+</sup>07] ein halb automatischer Positionshalten-Algorithmus und eine Wegpunkt Navigation vorgestellt. Dieser Algorithmus betrachtet allerdings keine Störeinflüsse während des Fluges von außen, wie zum Beispiel, dass das UAS von der Trajektorie abkommt oder Höhe verliert.

Ein weiterer Algorithmus zur Regelung von Trajektorien ist die nichtlineare Modellprädiktive Regelung (Model Predictive Control (MPC)) oder auch Receding Horizon Control (RHC) genannt. In [BA05] wird so ein Algorithmus vorgestellt. Die Autoren splitten dazu eine linearen 3D-Trajektorie in ihre  $z$ -,  $x$ - und  $y$ -Anteile auf, die dann hintereinander sequentiell abgeflogen werden. Der „tracking feedback“-Regler basiert auf der „receding horizon“-Punkt-zu-Punkt-Steuerung. Der resultierende Regler beinhaltet die zeitliche Ableitung des gesamten Auftriebs des UASs, wofür die Prinzipien der Flachheit („flatness“) und der „feedback linearization“ verwendet werden. Da diese Methode eine „erhebliche“ Rechenleistung erfordert, gibt es nur Simulationsergebnisse.

Auch in [CMV07] wird eine MPC basierte Trajektorienregelung (Trajectory Tracking (MPCTT)) für kleine UASs vorgestellt. Das MPCTT-System wird in der Arbeit mit einem Geschwindigkeits- und einem Positionsregler basierend auf einem PID-Regler verglichen. Simulationen haben gezeigt, dass das MPCTT-System und der PID-Positionsregler fast identische Ergebnisse bzgl. der Position in  $x$ - bzw.  $y$ - Richtung erzeugen. Der PID-Geschwindigkeitsregler liefert hier schlechtere Ergebnisse. In Bezug auf die Höhenregelung ist das MPCTT-System besser als die beiden PID-Regler. Zudem erzeugt das MPCTT-System geringeren Regelaufwand und Änderungen in den Lagewinkeln, so dass die Autoren diesen Regler bevorzugen. Die Simulationen wurden auf einem AMD Athlon 64 3200-PC (clock of 2,2 GHz, 1 GB Ram) berechnet.

[HWT08] beschreibt ein Verfahren, bei dem eine Trajektorie in eine Sequenz von  $N$  gewünschten Wegpunkten und dazugehörigen Geschwindigkeiten eingeteilt wird. Die zu regelnde Größe bei diesem Regler ist die Geschwindigkeit. Der so genannte „along track“-Geschwindigkeitsfehler wird mit einem PI-Regler geregelt. Der „cross track“-Fehler wird mit PID-Regler ausgeglichen. Die Integrale werden nach jedem Segment zurückgesetzt. Experimente haben gezeigt, dass die durchschnittliche Abweichung bei einer linearen Trajektorie während des Fluges bei  $0.5m$  liegt, bei einer Sollgeschwindigkeit von  $2m/s$  und einem Wind von  $5 - 10mph$ .

Ein weiterer Aspekt dieser Arbeit bezieht sich auf die Generierung von einem Pfad um vorhandene Hindernisse und dem sukzessiven Abfliegen der  $N$  generierten Segmente. Beim Abfliegen der einzelnen Segmente findet am Ende eines Segmentes eine Geschwindigkeitsreduzierung statt, um an den Übergängen die Richtungsänderungen mit geringerem Fehler durchführen zu können. Danach wird die Sollgeschwindigkeit wieder erhöht.

In Bezug auf die Trajektorienplanungen werden zwei unterschiedliche Methoden durchgeführt und simulativ evaluiert. Bei der ersten Methode wird immer der kürzeste Weg um die Hindernisse gewählt, ohne dabei die kinematischen und dynamischen Eigenschaften des Systems zu berücksichtigen. Das bedeutet, dass bei Richtungsänderungen das UAS stark abbremsen muss, damit der Regelfehler nicht zu groß wird. Das in der Simulation vorgestellte Szenario konnte mit dieser Methode in  $13.6s$  durchflogen werden.

Die zweite Methode ist der so genannte „fast marching“-Algorithmus mit Gradientenabstieg zur Berechnung einer Kurve durch die Hindernisse mit Rücksicht auf Geschwindigkeitsänderungen. Die so erzeugte Trajektorie ist zwar länger, aber es muss weniger abgebremst werden, da die dynamischen Eigenschaften des UASs berücksichtigt werden. Da die durchschnittliche Geschwindigkeit dadurch steigt, ist die Gesamtflugzeit von  $13.8s$  fast identisch.

In [CWC07b] und [CWC07a] wird ein quasioptimaler Trajektorien-Planer (basierend auf der differentialen Flachheit (vgl. [BA05]) und direkten Methoden) vorgestellt, mit dem

eine zeitoptimale Trajektorie generiert werden kann. Als Gütfunktion wird auch hier eine Geschwindigkeitskostenfunktion verwendet. Um die Trajektorie abzufliegen, wird sie in kleine Segmente eingeteilt, und es wird jeweils ein Geschwindigkeitsprofil dazu berechnet. Die Trajektorie wird dann mit Hilfe eines linearen LQ-Reglers abgeflogen. In Simulationen konnten vorgegebene Szenarien so erfolgreich abgeflogen werden.

Ähnlich wie in [HWT08] wird auch in [KY10] der komplette Pfad in eine Sequenz von  $N$  aufeinander folgenden Wegpunkten eingeteilt. Jedem Segment wird weiterhin eine Sollgeschwindigkeit zugeordnet. Der Trajektoriengenerator hat nun die Aufgabe, aus den linearen Segmenten abfliegbare Trajektorien zu erzeugen, die aus der 3D-Position  $[x_d(t), y_d(t), z_d(t)]$ , der Ausrichtung  $\psi_d(t)$  und deren zeitlichen Ableitungen  $[\dot{x}_d(t), \dot{y}_d(t), \dot{z}_d(t), \dot{\psi}_d(t)]$  bestehen . Die Trajektorie wird in zwei Schritten berechnet. Zuerst wird aus den maximalen Beschleunigungen eine Geschwindigkeitsfunktion erzeugt, aus der dann durch Integration die dazugehörigen Positionsdaten berechnet werden. Um die Position und die Geschwindigkeit zu regeln, wird ein PD- oder PID-Regler eingesetzt. Experimente haben gezeigt, dass dieser Regler es ermöglicht, z.B. eine vorgegebene Spirale mit einem maximalen Positionsfehler von ca.2m zu fliegen.

### 2.6.3 Fazit

In der Literatur gibt es viele Ansätze in Bezug auf Positions- und Trajektorienregelung. Für den Anwendungsfall müssen aber noch weitergehende Überlegungen gemacht werden. So fokussieren sich viele Arbeiten auf die Regelung der Geschwindigkeit beim Abfliegen von Trajektorien [HWT08], [CWC07b] und nicht primär auf die der Position. Gerade dieses ist aber zum Beispiel beim Landen auf sich bewegende Oberflächen wichtig, wo sich das UAS sowohl an die Geschwindigkeit, als auch an die Position anpassen muss. In [KY10] wird zwar diese Verbindung hergestellt, aber leider andere Aspekte nicht betrachtet. In keiner dieser Arbeiten werden Beschleunigungen im System geregelt, durch die die Güte der Regelung verbessert werden kann. Zum anderen werden Sollgeschwindigkeitsänderungen nur über Rampen oder sprunghaft geändert. Um die Regelgenauigkeit bzgl. der Position während einer Trajektorie zu erhöhen, müssten aber auch andere Übergangsfunktionen in Betracht gezogen werden. Ein weiterer Aspekt ist das schnelle Fliegen von einem Wegpunkt zu einem anderen. Während so eines Fluges braucht die Geschwindigkeit nicht geregelt zu werden, es sollte einfach so schnell geflogen werden, wie es der Antrieb und der Wind zulässt. Aus diesen Gründen werden in dieser Arbeit weitere Untersuchungen bezüglich der Navigation durchgeführt (siehe Kapitel 3).

## 2.7 Übergeordnete Aufgaben

### 2.7.1 Kollisionsvermeidung

In der letzten Vergangenheit untersuchten Forscher Lokalisations- und Kollisionsvermeidungsstrategien bei Drehflüglern vorwiegend in „indoor“-Szenarien. Hauptaugenmerk bei diesen Arbeiten liegt dabei auf den „Simultaneous Localization and Mapping“ (SLAM) Verfahren. Dabei geht es darum, dass sich Systeme in unbekannten Umgebungen autonom zurechtfinden können. Dazu müssen sie in der Lage sein, Karten ihrer Umgebung zu erstellen und sich anhand solcher Karten zu orientieren.

In [EKF04] werden erste Ansätze beschrieben, schon existierende 2D SLAM Verfahren auf 3D-Szenarien anzupassen. Als Ansatz werden Monte-Carlo-Localization-Methoden verwendet, die auf einem Framework von rekursiven Bayesian-Filters basieren, um aus Entfernung- oder optischen Sensoren die aktuelle Position zu schätzen. Anhand so ermittelte Positionen ist es dann möglich, z.B. verhaltensbasierte Navigation durchzuführen. In [MK06] werden solche Fragestellungen untersucht. Mit Hilfe eines Luftschiffes, welcher mit Ultraschallsensoren ausgestattet ist, ist es möglich, sich in Räumen und Gängen zu lokalisieren, mit festen Abständen an Wänden entlang zu fliegen, Durchgänge zu orten und zu entscheiden, in welche Richtung geflogen werden soll.

Eine der ersten Arbeiten, in denen ein Quadrokopter mit SLAM-Verfahren ausgerüstet wurde, wird in [GGB09] beschrieben. Als Sensor wird ein Laser-Entfernungssensor eingesetzt, der zum einen waagerecht zum Boden in einer Ebene die Abstände zu Wänden und Gegenständen misst und zum anderen über einen Spiegel den Abstand zum Boden berechnen kann. Das Navigationssystem beinhaltet mehrere Module für Lokalisation, Höhen- und Lageberechnung und SLAM. So konnte in Experimenten mit diesen Verfahren die Höhe des Quadroopters mit einer Genauigkeit von  $\pm 0.1m$  geregelt werden. Des Weiteren war das System in der Lage, sich nach einem Flug von  $5m$  in einer existierenden Karte zu lokalisieren.

Neben Abstandssensoren werden auch optische Sensoren für SLAM-Methoden eingesetzt. So wird in [Ahr08] ein Verfahren vorgestellt, welches auf den Einsatz einer WiFi-Kamera und Bildverarbeitung mit Hilfe der Computer Vision Library (openCV) [Int07] beruht. Experimente haben zum einen gezeigt, dass mit Hilfe von „Feature Tracking“ die Position eines Quadroopter driftfrei gehalten werden konnte. Dieses ist möglich, so lange die erkannten Merkmale im Bild während des Fluges nicht verloren gehen, also nicht aus dem Bildbereich verschwinden. Der zweite Teil dieser Arbeit beschäftigt sich mit Kollisionserkennung und -vermeidungsstrategien. So konnten bei einem experimentellen Flug

Hindernisse erkannt und über- bzw. umflogen werden. Bei einem ca.  $5m$  Flug, während dessen ein Stuhl überflogen wurde, driftete die Positionsberechnung dabei um einen Meter. In [ZSWS10] wird ein Verfahren vorgestellt, mit Hilfe dessen es möglich ist, entlang eines Korridors zu fliegen. Zu diesem Zweck wurde eine omnidirektionale Fischaugenkamera mit einer  $190^\circ$  Linse verwendet. Mit diesem Verfahren konnten Tiefenkarten mit Hilfe des optischen Fluxes während des Fluges berechnet werden, anhand derer sich der Quadrokopter orientieren konnte. So konnte in Experimenten die Position während eines Fluges mit einer durchschnittlichen Genauigkeit von 7.3% bei einem maximalen Fehler von 27% bestimmt werden.

## 2.7.2 Pfadplanung

Bei der Pfadplanung geht es um Methoden und Algorithmen, mit denen Wege und Pfade durch Gebiete, unter Berücksichtigung von Hindernissen oder anderen Anforderungen, berechnen werden können.

So wird z.B. in [MNC09] eine neuartige Methode vorgestellt, die es ermöglicht, Pfade durch eine Umgebung mit einer variablen Anzahl von Hindernissen zu planen. Die Methode ist eine Variation der „Rapidly-exploring Random Trees“. Der Ansatz in dieser Arbeit beruht auf dem Einsatz von „Pythagorean Hodograph“-Kurven (z.B. Bézier-Kurven), mit denen die einzelnen Knoten des Baums miteinander verbunden werden. Mit Hilfe dieser Methode ist es möglich, auch kinematische und dynamische Effekte des Systems mit zu berücksichtigen.

In [HPR08] wird eine Trajektorienplanung für einen Quadrokopter beschrieben, der in Gebieten, in denen kein GPS-Signal vorhanden ist, eingesetzt werden kann. Es wird ein so genannter „Belief Roadmap“-Algorithmus (BRM) verwendet, um „indoor“-Trajektorien zu planen. Dieser Algorithmus hat den Vorteil, dass er Sensorungenauigkeiten berücksichtigen kann. In Experimenten wurde dieser Algorithmus mit einem traditionellen „Probabilistic Roadmap“-Algorithmus (PRM) verglichen. Es hat sich gezeigt, dass der Fehler bei der Lokalisation mit dem BRM bei  $0.17m$  und die der PRM bei  $3.6m$  lag. Im Experiment wurde ein Laserabstandssensor verwendet, um sich in einer schon existierenden Karte zu lokalisieren.

Ein weiterer Aspekt bei der Pfadplanung liegt in der Berechnung von zeitoptimalen Pfaden durch ein Gebiet. Ziel ist es also, die Flugzeit zu minimieren.

So wird in [BHC08] ein numerisches Verfahren vorgestellt, durch das man zeitoptimale Pfade planen kann. Zur Berechnung der passenden Parameter wird ein nicht lineares Optimierungsverfahren eingesetzt. Um die Stützpunkte des Pfades miteinander zu verbinden, werden „b-spline“ Funktionen verwendet. Simulationen zeigen gute Ergebnisse.

In [HG10] wird ein Verfahren für Starrflügler vorgestellt, welches es ermöglicht, optimale noch fliegbare Pfade durch einen dreidimensionalen Raum zu finden. Minimale Kurvenradien des Flugsystems werden nicht unterschritten. Dabei können sowohl der Start als auch Endzustand des Systems in Bezug auf Position, Ausrichtung und Geschwindigkeit eingestellt werden. Neben einer numerischen Methode wird auch ein Verfahren untersucht, das auf geometrische Verfahren basiert. Das geometrische Verfahren hat sich dabei in Bezug auf Realzeitanwendungen als besser herausgestellt.

### 2.7.3 Landeverfahren

Bezüglich der Landeverfahren gibt es in der Literatur einige Arbeiten, die sich zum Teil mit dem automatische Landevorgang selber, als auch in Ansätzen mit einer Landung auf beweglichen Oberflächen beschäftigen.

Wie in Abschnitt 2.5 beschrieben, gibt es in der Literatur viele Ansätze, bei denen Kameras als Sensor zur Lokalisation eingesetzt werden. Diese Art der Lokalisation ist daher auch Grundlage vieler Landeverfahren (z.B. [FMS<sup>+</sup>08]). Dabei werden zwei unterschiedliche Methoden unterschieden (siehe Abschnitt 5.1). Bei der einen Methode befinden sich Kameras am Boden und erfassen das UAS in der Luft. Bei der anderen befinden sich Kameras an Bord des UASs und erkennen so genannte Landmarken, um sich an diesen auszurichten. In Abb. 2.5 sind exemplarisch zwei Landmarken dargestellt (vgl. auch 5.1). Häufig werden asymmetrische Landmarken verwendet, um zusätzlich zu der Position auch die Orientierung des UASs bestimmen zu können.

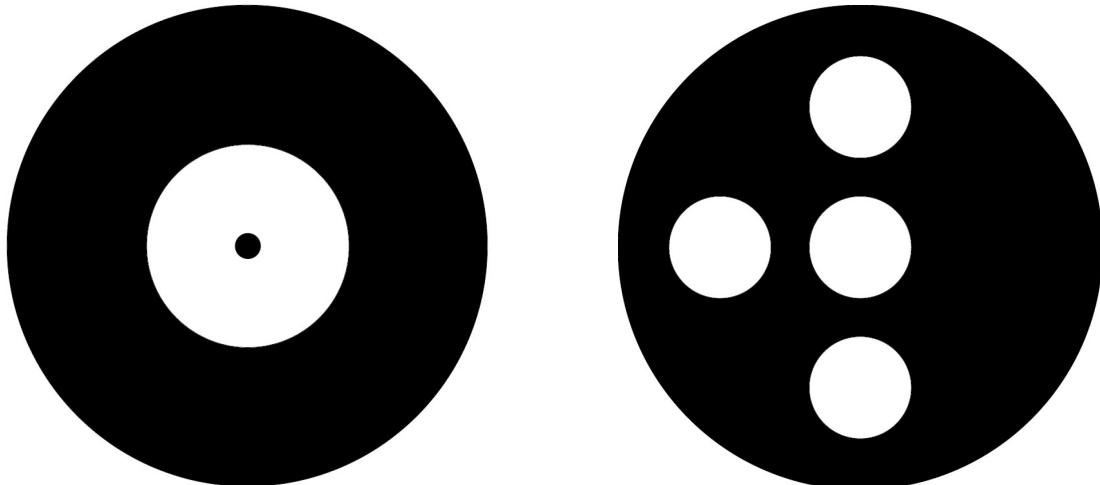


Abb. 2.5: Landmarken: symmetrische Landmarke (links) - asymmetrische Landmarke (rechts)

In [LSP08] wird beispielsweise solch eine Landmarke beschrieben und erläutert, wie der dazugehörige Trackingalgorithmus funktioniert. Es konnte gezeigt werden, dass mit Hilfe so eines Verfahrens die Position und Höhe des UASs mit einer Genauigkeit von 0.04 bzw. 0.02m bestimmt werden kann.

In anderen Arbeiten, wie z.B. in [SFS02] beschrieben, wurden solche Verfahren mit Hilfe eines klassischen Helikopters schon experimentell durchgeführt. Als Landmarke wurde hier das „H“ eines Hubschrauberlandeplatzes verwendet. Des Weiteren wird in dieser Arbeit ein Landeargorithmus vorgestellt, mit dessen Hilfe der Helikopter automatisch auf dieser Landefläche landen kann. Dabei wird der Helikopter über GPS zur Landeposition geführt. Kommt das „H“ ins Blickfeld der Kamera, übernimmt die Bildverarbeitung die Lokalisation und führt den Helikopter zum Landepunkt. Die durchschnittliche Genauigkeit nach mehreren Landungen lag bei ca. 0.4m. Das „H“ bietet zusätzlich die Möglichkeit, die Ausrichtung des Helikopters zu erkennen<sup>4</sup>. Die Genauigkeit der Ausrichtung nach dem Landen lag dabei bei 7°. Das Verfahren hat allerdings den Nachteil, dass es vorkommen kann, dass die GPS-Position nicht genau genug ist, damit das „H“ trotz „Suchmodus“ in Sichtweite der Kameras kommt. Des Weiteren ist das verwendete Prozessorbord PC-104 relativ schwer und groß, so dass es für kleinere UASs nur schlecht adaptierbar ist.

In [MCM009] wird dagegen ein Verfahren vorgestellt, bei dem drei am Boden befindliche Kameras einen klassischen Helikopter am Himmel tracken, um ihn so punktgenau zum Landepunkt zu manövrieren. Zu diesem Zweck wurden die Kameras so in einem Halbkreis am Boden angeordnet, dass sich deren Sichtbereiche im Bereich oberhalb des Landepunktes treffen. Am Helikopter wurden vier farbige Kugeln angebracht, die von den Kameras erkannt werden sollten. Mit dieser Methode war es möglich, die Position, die Höhe, die relative Ausrichtung und die Lage des Helikopters zu bestimmen. Die Genauigkeit bei der Positionsmessung lag bei 0.05 – 0.1m und eine automatische Landung konnte gezeigt werden. Aber auch hier ist das Problem des zuverlässigen Übergangs von GPS-Navigation hin zur optischen Navigation nicht gelöst.

Erste Arbeiten zum Thema Landen auf nicht geraden Oberflächen werden zum Beispiel in [YGP10] und [BF07] vorgestellt. In [YGP10] geht es hauptsächlich um die Entwicklung einer geeigneten Prozedur zur Bestimmung der aktuellen Deckbewegung eines Schiffes. Es geht darum, den Trend der Bewegung des Decks zu ermitteln. In [BF07] wird dagegen beschrieben, wie man mit einem Helikopter dynamisch auf geneigten Flächen landen kann, wenn die Lage der Fläche bekannt ist. Experimentelle Ergebnisse haben gezeigt, dass das Verfahren es ermöglicht, auf Landeflächen mit einem Verkipungswinkel von bis zu 60° zu landen. Für die Experimente war es notwendig, die aktuelle Position des Helikopters mit

---

<sup>4</sup>180° Verdrehungen werden nicht erkannt.

wenigen Zentimetern genau zu bestimmen. Zu diesem Zweck wurde ein bodenbasiertes Kamera-Positionierungsverfahren von der Firma VICON verwendet.

In den Arbeiten [Voo09b] und [VN09] wird ein Verfahren zum Landen auf einer sich bewegenden Plattform beschrieben. In der Simulation wurde gezeigt, dass der entwickelte Regelalgorithmus, der aus einem linearen Höhenregler und einem nichtlinearen 2D-Tracking-Algorithmus mit „feedback linearization“ besteht, funktioniert. Das Landeverfahren selber wird in zwei Phasen eingeteilt, einer Justierungsphase („align“) und einer Landephase („land“). In der Arbeit werden keine experimentellen Ergebnisse vorgestellt. Auch wurde außer Acht gelassen, wie die Position des UASs während des Landevorgangs ermittelt werden soll.

Diese Fragestellung wurde unter anderem in den Arbeiten von [HRHM08] und [HHMR10] erörtert. Es wird ein Höhen- und Positionsregler vorgestellt, der auf dem optischen Fluss basiert. Mit diesem Algorithmus konnte eine Landung auf einer sich dynamisch bewegenden Oberfläche durchgeführt werden. Der Regler ist in der Lage, sowohl horizontale als auch vertikale Bewegungen der Oberfläche auszugleichen. Der optische Fluss hat aber den Nachteil, dass er keine absolute Lokalisation bieten kann und auf passende Lichtverhältnisse angewiesen ist. Dieses Verfahren bietet aber die Möglichkeit, das in dieser Arbeit beschriebene Landeverfahren (siehe Abschnitt 6.9) zu ergänzen.

Ein weiteres rein optisches Verfahren zur Landung auf einer sich bewegenden Plattform wird in [WRZ10] und [WMZ10b] vorgestellt. Dieses Verfahren basiert auf dem Trackingalgorithmus, welches in der Spielekonsole „Wii“ eingebaut ist. Im Gegensatz zum optischen Fluss, ist es mit Hilfe dieses Verfahrens möglich, die genaue relative Position des UASs in Bezug auf die Landefläche zu bestimmen. Zu diesem Zweck wurde eine Wiimote-Kamera unterhalb des UASs angebracht und die dazugehörigen LED-Dioden in einem Dreieck auf der Landefläche platziert. Solange die LED-Dioden im Sichtfeld der Kamera sind und der Abstand zwischen Kamera und Dioden nicht zu groß ist, funktioniert dieses Verfahren. Dieser Trackingalgorithmus wurde auch verwendet, um Personen zu verfolgen [WMZ10a]. Zu diesem Zweck wurden die LED-Dioden in eine Mütze integriert und Personen auf den Kopf gesetzt. Das UAS war nun in der Lage, in eine Höhe von  $0.4m$  den Personen zu folgen. Die durchschnittliche Positionsabweichung bei einem 5-minütigen Flug betrug „indoor“  $0.03m$ .

## 2.7.4 Fazit

Trotz der schon vorhandenen Arbeiten ist die Aufgabe das vollautomatische Landens noch nicht endgültig gelöst. In den Arbeiten werden hauptsächlich Teilespekte wie optische Lokalisationsverfahren beschrieben, die man zum punktgenauen Landen benötigt. Der

Übergang von GPS basierter Navigation zur hochgenauen Nahfeldlokalisierung wird aber nicht behandelt. Da die vorgestellten Lokalisationsverfahren alle eine begrenzte Reichweite haben, stellt gerade der robuste und zuverlässige Übergang eine große Herausforderung dar. In Bezug auf das Landen auf sich bewegenden Oberflächen gibt es bis jetzt nur optische Methoden, die für den Einsatz im Dunkeln nicht geeignet sind. Vor allem das Zusammenspiel zwischen Fahrzeugannäherung, -verfolgung und Landevorgang wird in den Arbeiten nicht behandelt. Diese genannten Herausforderungen und Aspekte werden in dieser Arbeit näher betrachtet (siehe Kapitel 3).



# Kapitel 3

## Eigener Ansatz

Im Rahmen dieser Arbeit werden folgende Themen detailliert betrachtet und stellen somit die eigenen Ansätze und Arbeiten des Autors dar.

1. Entwicklung von Hardwarekomponenten für Mini-UAS, die speziell für den erwähnten Anwendungsfall aus Abschnitt 1.1 geeignet sind. Dazu gehören eine Avionik und Stellglieder für Motoren, die auf den Zielsystemen 4.1.1 oder 4.1.2 installiert werden, und eine Referenz-Bodenstation (siehe 4.2). Die Avionik und Motorsteller bieten die Möglichkeit, die zur Stabilität des Flugsystems notwendigen hohen Regelfrequenzen trotz aufwändiger Berechnungen umzusetzen. Des Weiteren wird eine IMU-GPS-Filterung mit Berücksichtigung der zeitlichen Verzögerung von GPS-Signalen (siehe 4.3.2) beschrieben. Diese Filterung ist notwendig, um die absolute Positionsgenauigkeit während des Fluges zu erhöhen. Zur Lokalisation wird ein Messsystem für den Nah- und Mittelbereich basierend auf einem „Wireless Sensor Network“ (WSN) (siehe 5.2) entwickelt. Dieses System bietet die Möglichkeit tageszeitunabhängig robust die Position des UAS zentimetergenau zu bestimmen.
2. Basierend auf der optimierten Hardwareplattform wurden neue Reglerstrategien entwickelt. Dazu gehört zum einen die Weiterentwicklung einer schon vorhandenen global stabilen Lageregelung mit einem zusätzlichen integralen Anteil zum Ausgleichen von Asymmetrien und Störeinflüssen. (siehe 6.1 und 7.1). Des Weiteren wird ein neuer 2D-Wegpunktregler beschrieben, der es ermöglicht, zwischen Wegpunkten ohne Geschwindigkeitsbegrenzung entlang einer Sollstandlinie zu fliegen, um so „schnell“ das Einsatzgebiet zu erreichen. Der Regler stellt sicher, dass das UAS nicht unter eine gewisse Sollhöhe absinkt (siehe 6.3, 7.2 und 8.1). Als letztes wird ein neuer 3D-Trajektorienregler vorgestellt, der in der Lage ist, Objekte zu verfolgen und 3D-Trajektorien wie beispielsweise die geozentrischen Kreise aus dem Anwendungsfall

abzufliegen. Hauptaugenmerk liegt dabei auf der Positionsgenauigkeit während des Fluges, um auch die Landung auf einem beweglichen Fahrzeug zu gewährleisten (siehe 6.4-6.6, 7.3-7.4 und 8.2-8.3).

3. Entwicklung eines vierstufigen Landealgorithmus basierend auf dem WSN, der es ermöglicht, automatisch aus jeder Position zuverlässig in einem vordefinierten Bereich (z.B. in einer Box) zu landen. Dieser Algorithmus schließt die Lücke zwischen der groben GPS- und der zentimetergenauen Nahfeldlokalisierung. Zusammen mit den vorgestellten Regelungsstrategien sind Landungen auf beweglichen Landeplattformen möglich (siehe 6.8, 7.6 und 8.4).

# Kapitel 4

## Flugsysteme

In diesem Kapitel werden die in der Arbeit verwendeten Flugsysteme und ihre Teilsysteme beschrieben. Zum Beginn wird ein Überblick über die beiden Flugsysteme *Guard M* (Medium) und *Guard S* (Small) gegeben. Es werden spezielle Teilsysteme wie Avionik, Motorsteller und Bodenstation vorgestellt, die für den Betrieb essenziell sind. Zum Schluss wird ein Überblick über wichtige Aspekte bezüglich Sensorsignalerfassung und -aufbereitung gegeben.

### 4.1 Flugsysteme

#### 4.1.1 Guard M

Die Abb. 4.1 zeigt ein Foto von dem Guard M.



Abb. 4.1: Guard M

Der Rahmen des Systems besteht aus einer Gemisch aus Glasfasern, Kohlefasern und Kevlar. Bei dem Design des Gehäuses lag das Hauptaugenmerk auf niedrigem Gewicht, Festigkeit und Schutz der Umgebung vor den schnell drehenden scharfen Rotorblättern.

Die Abmaße des Systems sind  $1.1m \times 1.1m \times 0.2m$ . Auf Grund des Abfluggewichts von ungefähr 3kg ist das System mit 4 Motor-Rotor-Kombinationen ausgestattet, die jeweils 1.7kg Schub erzeugen. Das bedeutet, dass das System in der Lage ist, ein zusätzliches Gewicht von 1kg zu tragen, und dennoch genügend Schub vorhanden ist, seine Agilität beizubehalten. Eine Motor-Rotor-Kombination besteht aus einem bürstenlosen Motor<sup>1</sup> (Direktantrieb), einem passenden Propeller und einem Motorsteller. Als Avionik wird die in Abschnitt 4.2.1 vorgestellte Avionik verwendet. Die Flugzeit des Systems beträgt ca.15min.

#### 4.1.2 Guard S

Die Abb. 4.2 zeigt ein Foto von dem Guard S.



Abb. 4.2: Guard S

Der Guard S basiert auf dem System von [Mik10]. Übernommen wurden der Rahmen, die Motoren und die Motorsteller. Aber wie der Guard M wurde auch dieses System mit der im Abschnitt 4.2.1 beschriebenen Avionik bestückt. Die Abmaße des Systems sind ca  $0.4m \times 0.4m \times 0.15m$ . Das System wiegt ungefähr 900g und kann ein zusätzliches Gewicht von ca. 500g tragen. Die Flugzeit des Systems beträgt ca.20min.

## 4.2 Hardwarekomponenten

Die Hardware, die für den autonomen Betrieb eines Quadroopters nötig ist, besteht aus einer Avionik (siehe 4.2.1), vier Motorstellern (siehe 4.2.2) und einer Bodenkontrollstation (siehe 4.2.3).

---

<sup>1</sup>Orbit 20-18 von der Firma Plettenberg

#### 4.2.1 Avionik

Hauptbestandteil eines UASs ist die so genannte Avionik (oder auch Flugführungsrechner). Die Avionik hat die Aufgabe, alle Sensorsignale einzulesen und aufzubereiten, über einen Flugregler neue Stellwerte zu berechnen und diese dann an die Motorsteller zu übertragen. In Abb. 4.3 ist die entwickelte Avionik zu sehen.

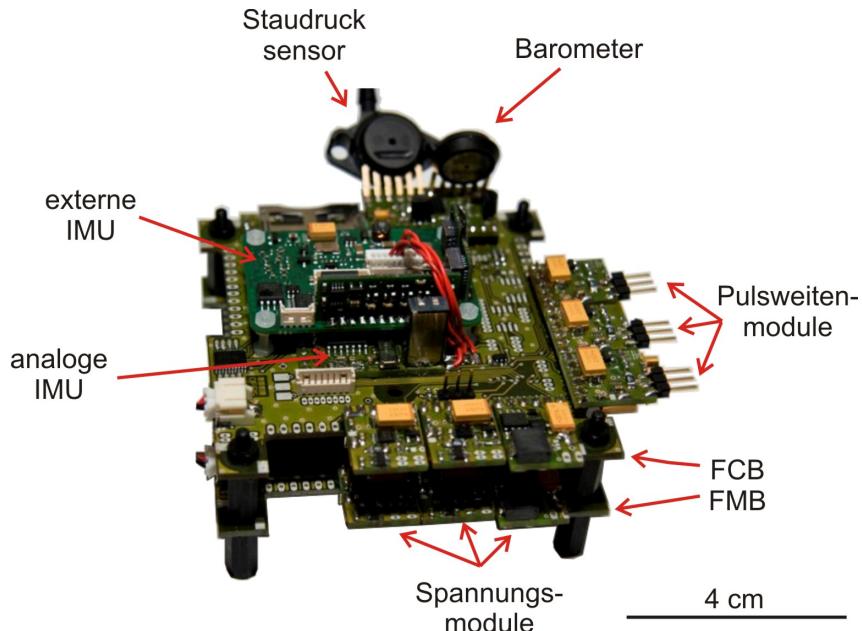


Abb. 4.3: Foto der Avionik mit Modulen: Der GPS-Sensor und das Funkmodul befinden sich auf dem FMB und sind im Bild nicht zu sehen.

Die Avionik besteht aus zwei übereinander liegenden Platinen. Die obere Platine ist für die Sensoraufbereitung und Flugregelung zuständig und wird als „Flight Control Board“ (FCB) bezeichnet. Die untere Platine hat die Funktion, Navigationsaufgaben wie Wegpunktberechnung und Kommunikation zu bearbeiten und wird als „Flight Management Board“ (FMB) bezeichnet. Als Prozessor wird ein TriCore® TC1796 von der Firma Infineon verwendet. Dabei handelt es sich um einen 32-Bit-Prozessor, der mit 150MHz betrieben wird. Dieser Prozessor wurde gewählt, um trotz ressourcenaufwändiger Berechnungen<sup>2</sup> möglichst hohe Regelfrequenzen (bis 1kHz) zu erreichen, um Regelstörungen „schnell“ ausgleichen zu können [HP90].

In Abb. 4.4 ist eine schematische Darstellung der entwickelten Avionik mit den Aktualisierungsraten der einzelnen Komponenten dargestellt.

<sup>2</sup>z.B. GPS-Koordinaten sollten im double-Format (64bit) gerechnet werden, um die Genauigkeit zu erhöhen.

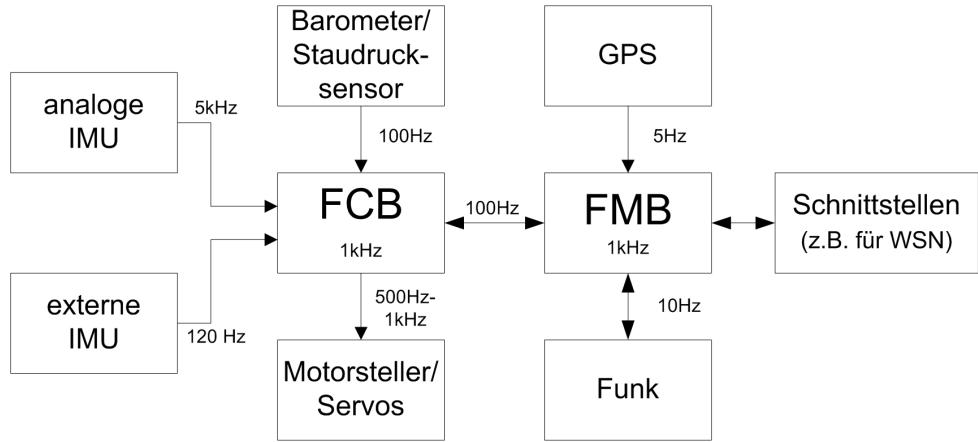


Abb. 4.4: Avionik

Das Design der Avionik zeichnet sich durch seine modulare Bauweise aus, die es ermöglicht, sie nicht nur für Quadrokopter, sondern für viele Arten von autonomen Systemen, wie z.B. Starrflügler, Fahrzeuge usw. zu nutzen. Die Avionik kann mit folgenden Modulen ausgestattet werden:

- **analoge inertiale Messeinheit (IMU)** bestehend aus 3 Drehraten- und 3 Beschleunigungssensoren zur Bestimmung der aktuellen Lage (ohne Ausrichtung gen Norden) mit einer Aktualisierungsrate von 5kHz.
- **extern inertiale Messeinheit (IMU)** zur Kalibrierung der analogen IMU inklusive Kompass.
- **Barometer und Staudrucksensor** zur Ermittlung der Flughöhe und Fluggeschwindigkeit bei Starrflüglern.
- **Global Positioning System (GPS)** zur Ermittlung der aktuellen Position im Freien.
- **Pulsweitenmodule (PWM)** über die bis zu 11 Servosteller für z.B. Landeklappen oder Ruder angeschlossen werden können.
- **Funkmodul** zur Kommunikation mit der Bodenkontrollstation (siehe 4.2.3)

Zusätzlich besitzt die Avionik diverse Schnittstellen zum Anschließen weiterer externer Sensoren und Aktuatoren.

### **4.2.2 Motorsteller**

Neben der Avionik ist der Motorsteller das wichtigste Subsystem eines Quadroopters. Er muss in der Lage sein, die hohen Regelfrequenzen der Avionik zu unterstützen. Zu diesem Zweck wurde ein Motorsteller entwickelt, der neue Stellwerte mit einer Frequenz von 1kHz empfangen kann. Die Leistungsdaten des entwickelten Motorstellers für bürstenlose Direktantriebe sind folgende:

- Leistungsaufnahme: bis 250W
- maximaler Strom: ca. 20A Dauer, 30A Spitze
- maximale Spannung: 13V
- maximale Umdrehung: bis 8000 u/min
- Ansteuerungsfrequenz: bis 1kHz

### **4.2.3 Bodenkontrollstation**

Die Bodenkontrollstation (BKS) hat die Aufgabe, die Kommunikation mit den einzelnen UASs aufrecht zu erhalten. Sie besteht, ähnlich wie die Avionik, aus einer mit einem TriCore® TC1796 bestückten Platine und wird als „Ground Control Board“ (GCB) bezeichnet. Zusätzlich besitzt das GCB ein Barometer, um Korrekturdaten an die einzelnen UAS zu senden (siehe 4.3.1) und einen GPS-Sensor zur Ermittlung der eigenen Position.

## **4.3 Sensorfusion**

Zur Stützung der durch die IMU berechneten Positionsdaten müssen zusätzliche Sensoren verwendet werden. Als komplementäre Sensorik wird für die Höhe ein Barometer und für die Position ein GPS-Sensor verwendet. Die Arbeiten von [DZ] und [Lea99] dienten als Grundlage für die Entwicklung der im folgenden beschriebenen Filter zur Stützung und Berechnung der Höhe und der Position.

### **4.3.1 Höhenberechnung**

Zur Stützung der Höhenberechnung wird ein Barometer verwendet. Der verwendete Sensor<sup>3</sup> hat eine Messgenauigkeit von ca. 0.2m. Die Höhenmessungen mit Barometern unterliegen aber witterungsbedingten Schwankungen. Aus diesem Grund befindet sich in der

---

<sup>3</sup>MPX4115

Bodenkontrollstation das gleiche Barometer. Die aktuelle Höhe berechnet sich nun aus der Differenz der beiden Barometerwerte, so dass witterungsbedingte Schwankungen kompensiert werden können. Dieser Lösungsansatz hat einen weiteren Vorteil. Im Anwendungsfall (siehe 1.1) wurde beschrieben, dass das UAS auch auf einem fahrenden Fahrzeug landen soll. Durch den Einsatz des Referenzbarometers passt sich das UAS so automatisch den Höhenänderungen des Fahrzeugs an. Zur Berechnung der aktuellen Höhe wird ein Kalman-Filter verwendet. Die Gl. (4.1) zeigt die Funktionsweise des Filters.

$$z_{t+1} = z_t + K_z \cdot (z_t^{Baro} - z_t) \quad (4.1)$$

mit  $z_t^{Baro} = z_t^{UAS} - z_t^{BKS}$  und  $K_z$  ist der Kalman-Faktor,  $0 \leq K_z \leq 1$ .

### 4.3.2 Positions berechnung

Zur Stützung der Position wird ein GPS-Sensor verwendet (vgl. auch [Wen07], [GW01]). Bei den GPS-Daten muss beachtet werden, dass es eine zeitliche Verzögerung zwischen dem aktuellen Zustand  $S_t$  des UASs und den durch den GPS-Sensor gemessenen Zustand  $S_{t-t_p}$  gibt. In Abb. 4.5 ist exemplarisch der zeitliche Versatz zwischen IMU berechneter Position  $p_t$  und GPS<sup>4</sup> ermittelter Position  $p_t^{GPS}$  dargestellt.

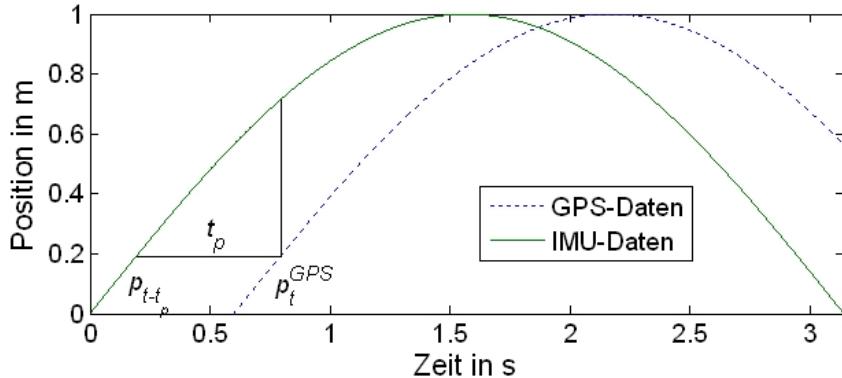


Abb. 4.5: Signalverzögerung bei GPS-Signalen bei linearer Bewegung

Es ist zu erkennen, dass das GPS-Signal eine konstante zeitliche Verzögerung von ungefähr 600ms besitzt. Diese Zeit entspricht der auf der Avionik gemessenen Verzögerung. Sie  $t_p$  setzt sich zusammen aus Signallaufzeiten  $t_g$ , Berechnungszeiten im GPS-Sensor  $t_c$  und Übertragungszeiten an Schnittstellen  $t_s$ . Dieser Versatz tritt sowohl bei der Position  $p_t^{GPS}$  als auch bei den Geschwindigkeitsdaten  $v_t^{GPS}$  auf. Würde dieser Versatz nicht berücksich-

<sup>4</sup>GPS-Sensor von der Firma  $\mu$ -blox

tigt werden, würde z.B. bei einer Geschwindigkeit des UASs von  $10m/s$  ein Positionsfehler von ca.  $5m$  auftreten.

Es gibt zwei unterschiedliche Methoden, um diesen Fehler zu vermeiden. Zum einen kann der zeitliche Versatz bei der Modellierung des Gesamtsystems und der Auslegung der Regler mit berücksichtigt werden. Dieses Verfahren wird z.B. in [Kor05] vorgestellt.

Die zweite Möglichkeit besteht darin, den zeitlichen Versatz bei der Fusionierung mit den aktuellen IMU-Daten herauszurechnen. Dieses wird in der Art realisiert, dass die GPS-Position  $p_t^{GPS}$  und Geschwindigkeit  $v_t^{GPS}$  mit der von der IMU berechneten Position  $p_{t-t_p}$  und Geschwindigkeit  $v_{t-t_p}$  aus der Vergangenheit gefiltert werden. Diese Methode hat den Vorteil, dass keine Anpassungen an den Modellen und Reglerstrukturen durchgeführt werden müssen.

Als Filter wird ein modifizierter Kalman-Filter verwendet (siehe Gl. (4.2), Filter für den 1D-Fall).

$$p_{t+1} = p_t + K \cdot (p_t^{gps} - p_{t-t_p}) \quad (4.2)$$

wobei  $K$  der Kalman-Faktor ist,  $0 \leq K \leq 1$ .  $v_{t+1}$  wird analog berechnet.

In Abb. 4.6 ist das Ergebnis der Filterung zu sehen.

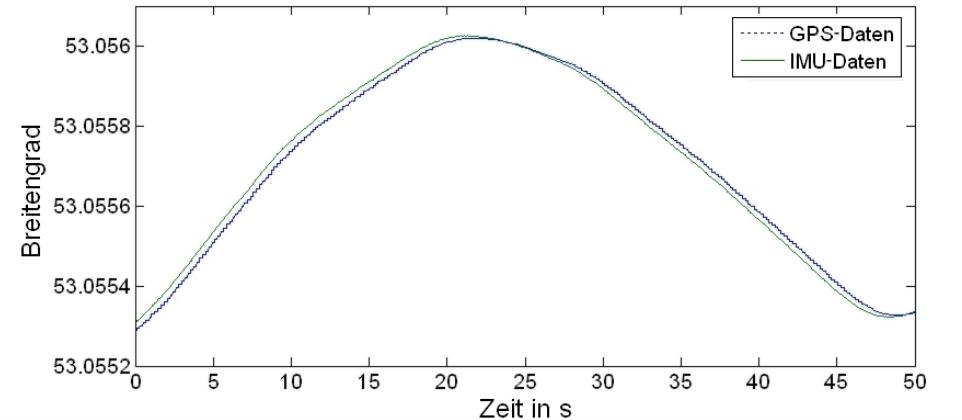


Abb. 4.6: Laufzeitverzögerung: Signal nach Filterung

Man kann erkennen, dass das Signal driftfrei (mit Ausnahme der Drift vom GPS-Sensor) ist und der zeitliche Versatz kompensiert wird, also das Signal trotz der Filterung aktuell bleibt. Abb. 4.7 zeigt einen detaillierteren Ausschnitt des Signalverlaufs, in dem die zeitliche Verzögerung der 600ms des GPS-Signals zu sehen ist.

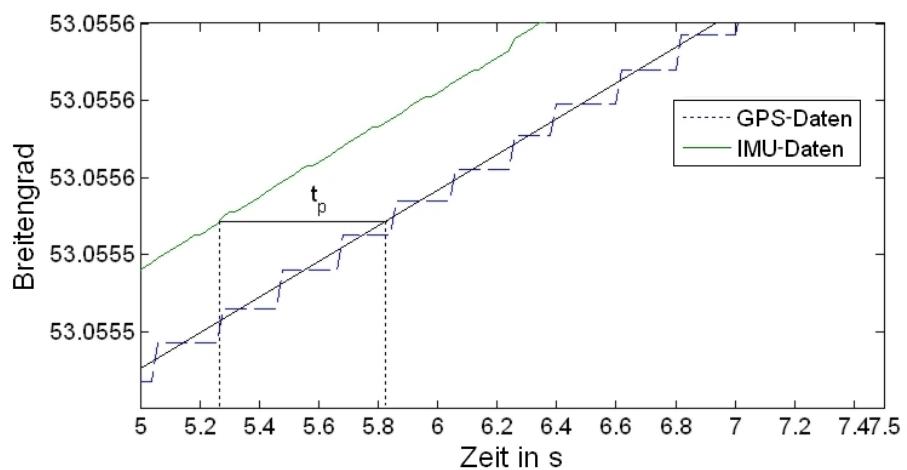


Abb. 4.7: Laufzeitverzögerung: Signal nach Filterung (Ausschnitt)

# Kapitel 5

## Verfahren zur Lokalisation im Nahbereich

In diesem Kapitel werden Verfahren zur Lokalisation im Nahbereich mit einer Messgenauigkeit von wenigen Zentimetern vorgestellt. Solch ein Verfahren ist die Grundlage für das punktgenaue automatische Landen.

### 5.1 Optische Messverfahren

Da in der Forschung die optischen Verfahren weit verbreitet sind, wurden zuerst zwei optische Verfahren zur Lokalisation untersucht. Beim ersten Verfahren wurde eine nach unten gerichtete so genannte Nadir-Kamera verwendet, um eine sich am Boden befindliche Landmarke zu tracken (vgl. auch [LSP08], [WRZ10], [SFS02], [WMZ10b]). Beim zweiten Verfahren wurde eine Zenit-Kamera am Boden platziert, um das UAS von unten aus zu tracken (vgl. auch [MCMO09]).

Beide Verfahren sind in der Lage, unter definierten Bedingungen die Position des UASs relativ zur Landeposition zuverlässig zu bestimmen. Während es bei dem Zenit-Kamera-System zusätzlich auch möglich ist, die aktuelle Geschwindigkeit des UASs zu ermitteln, kann man beim Nadir-Kamera-System durch den Einsatz asymmetrischer Landmarken auch die Ausrichtung des UASs bestimmen .

In Abb. 5.1 sind beide Verfahren dargestellt.



Abb. 5.1: Kamerasysteme: Kamera (Nadir) trackt Zielmarke am Boden (links) - Kamera (Zenit) trackt UAS am Himmel (rechts)

Beide Systeme haben aber drei nennenswerte Nachteile, weswegen sie für die automatische Landung nicht als optimale Lösung angesehen werden.

1. **Lichtverhältnisse:** Optische Systeme haben den Nachteil, dass sie auf adäquate Lichtverhältnisse angewiesen sind oder man z.B. bei Dunkelheit für zusätzliche Lichtquellen sorgen muss. Des Weiteren haben vor allem beim Zenit-Kamera-System wechselnde Lichtverhältnisse in Folge von Sonneneinstrahlung oder Schatten durch Wolkenbildung das System so sehr gestört, dass zeitweilig das UAS nicht von der Umgebung unterschieden werden konnte.
2. **Begrenzter Öffnungswinkel der Kamera:** Ein zweiter großer Nachteil ist der begrenzte Öffnungswinkel der Kamera. Sobald das UAS aus irgendeinem Grund aus dem Sichtbereich der Kamera gerät, bleibt keine andere Möglichkeit, sich auf andere Sensoren wie z.B. GPS-Sensoren zu verlassen. Lösungsansätze wie bewegliche Kameras (vgl.[WMZ10b]) sind grundsätzlich möglich, aber immer mit zusätzlichem Kalibrierungs- und Rechenaufwand verbunden.
3. **Entfernungsvariant:** Ein dritter Nachteil dieser beiden Methoden ist die Entfernungshängigkeit. Befanden sich das UAS oder die Landmarke in der Nähe des Fokus der Kamera, funktionierten beide Verfahren zuverlässig. War der Fokus aber nicht korrekt eingestellt oder das UAS bzw. die Landmarke zu weit weg, konnte keine Lokalisation mehr durchgeführt werden.

Die Punkte 2 und 3 können jeweils durch Objektive entweder in die eine oder in die andere Richtung verbessert werden. Durch einen weiteren Öffnungswinkel wird die Reichweite negativ beeinflusst und umgekehrt.

**Fazit:** Auf Grund der genannten Nachteile wird für die Lokalisation im Nahbereich kein optisches Verfahren eingesetzt. Um die in dem Anwendungsfall (siehe 1.1) geforderten Anforderungen zu erfüllen, wird eine Kombination von Funklaufzeitmessung und Ultraschallmessung verwendet, die im folgenden Abschnitt 5.2 beschrieben wird.

## 5.2 Lokalisation über Bakensystem

Das verwendete Lokalisationssystem (siehe Abb. 5.2) besteht zum einen aus vier einzelnen Funkknoten, die in einem Viereck in einer Ebene am Boden angeordnet sind und zum anderen aus einem Funkknoten, welcher sich unter dem Flugsystem befindet. Vier Funkknoten, anstelle der notwendigen drei, erhöhen die Messgenauigkeit (siehe Abschnitt 5.2.2) und Ausfallsicherheit der Lokalisation.

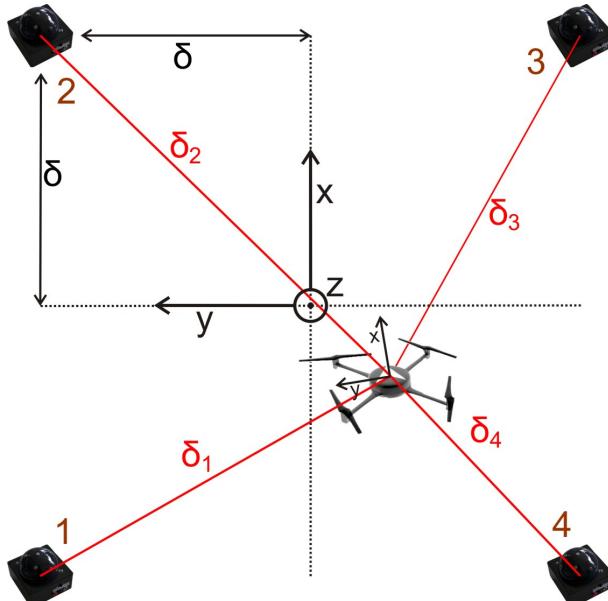


Abb. 5.2: Schema: Bakensystem mit vier passiven Sensorknoten am Boden und einem aktiven unter dem UAS.

Jeder Sensorknoten am Boden ist mit Ultraschallempfängern und einem Funkmodul [Nan] ausgestattet, welches Abstände über die Signallaufzeiten messen kann. An jedem Knoten sind insgesamt 6 Ultraschallempfänger montiert (siehe Abb. 5.3), um einen größeren Empfangsradius (ca. 160° anstatt von 70° Öffnungswinkel bei einem Sensor.) abzudecken. Der Sensorknoten unter dem Flugsystem ist mit einer Ausnahme identisch. Anstatt der Ultraschallempfänger befinden sich am Flugsystem mehrere Ultraschallsender. Diese sind ebenfalls so angeordnet, einen möglichst großen Bereich abzudecken (siehe Abb. 5.3).

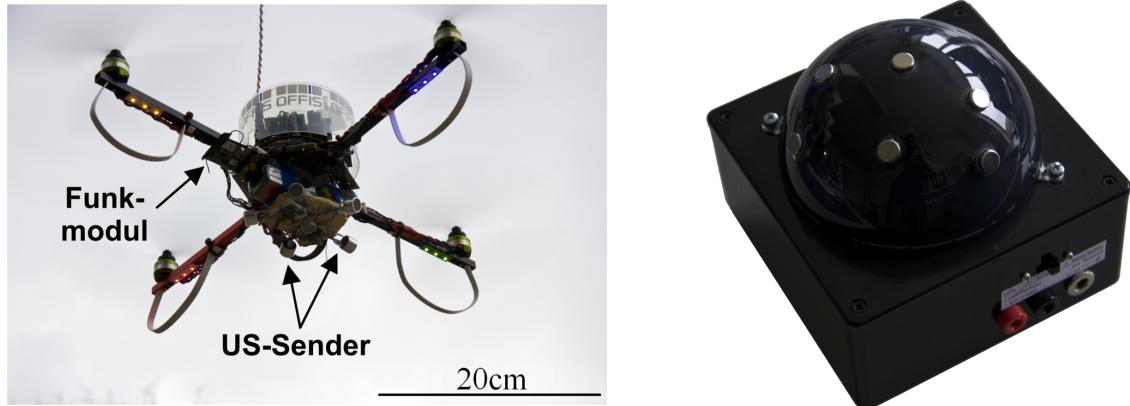


Abb. 5.3: Guard S mit Lokalisationssystem (links) - Funkknoten mit Ultraschallempfängern (rechts)

Die Ultraschallempfänger sollten nicht am Fluggerät angebracht werden. Aufgrund störender Einflüsse wie Vibratoren am Flugsystem oder Schall, der durch die Propeller erzeugt wird, würden sonst die Messdaten zu stark verrauscht und gestört werden. In Experimenten ist die Ultraschallmessung teilweise komplett ausgefallen. Um keine aufwändigen Kalibrierungen bzw. mechanische Anpassungen (z.B. Dämpfungsglieder) am Flugsystem durchführen zu müssen, sollten sich die Ultraschallempfänger am Boden und nur die Sender unter dem Flugsystem befinden.

### 5.2.1 Messgenauigkeiten

Wie im vorigen Abschnitt beschrieben, besteht das Lokalisationsssystem aus einem ultraschall- und aus einem funklaufzeitbasierten Messverfahren. In diesem Kapitel werden beide Messverfahren miteinander verglichen und ihre Messgenauigkeiten ermittelt.

Abb. 5.4 zeigt an einem Beispiel den Messaufbau, mit dem die beiden Messverfahren vermessen wurden. Der Abstand der Knoten zueinander (Kantenlänge des Quadrats) beträgt 5m ( $\delta = 2.5m$ ). Der Abstand darf nicht kleiner sein, als die Messgenauigkeit der Funklaufzeitmessung (siehe unten) und nicht größer als die Reichweite der Ultraschallsensoren ( $\delta < 10m$ ). Während der Messungen wurde der Quadrokopter auf der Spitze eines 0.63m langen Stativs montiert und an vorher präzise ausgemessenen Positionen platziert (hier genau in der Mitte). Während der Messungen wurden die Motoren des Quadrokopters eingeschaltet, um ein möglichst realistisches Umfeld zu generieren. Dann wurde über einen Zeitraum von ca. 100s die Messdaten beider Verfahren aufgezeichnet. Dabei liefert das beschriebene Messsystem neue Daten mit einer Frequenz von 14Hz.



Abb. 5.4: Messaufbau: Quadrokopter im Zentrum des Vierecks

In der Abb. 5.5 sind die Messdaten aller vier Sensorknoten (Knoten 1 - 4) zu erkennen. Bei dieser Messung befand sich der Quadrokopter genau in der Mitte des Vierecks (siehe Abb. 5.4).

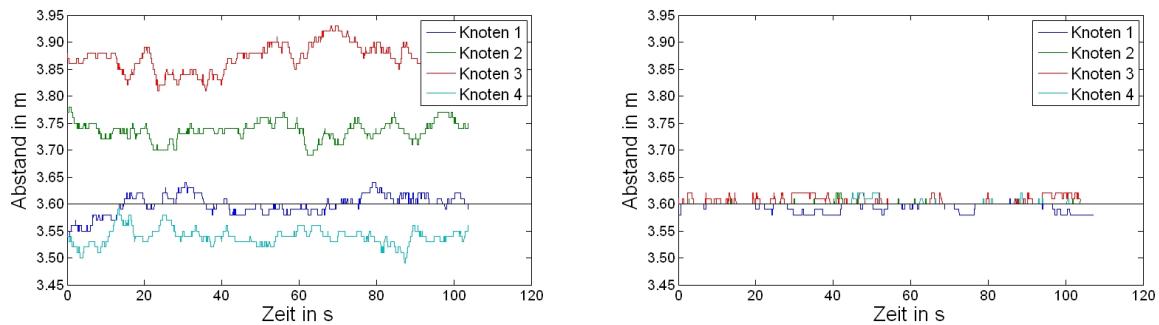


Abb. 5.5: Messdaten der vier Sensorknoten: Funklaufzeit (links) - Ultraschall (rechts)

Es ist zu erkennen, dass das Rauschen bei der Funklaufzeitmessung höher ist (ca.  $0.15m$ ) als bei der Ultraschallmessung (ca.  $0.03m$ ).

In Abb. 5.7 sind die Messdaten eines Sensorknotens (hier Knoten 1) zum Quadrokopter an verschiedenen Positionen zu sehen. Dabei wurde der Quadrokopter einmal zwischen den Sensorknoten 2 und 3 (Rückposition: Abb. 5.6 links) und einmal direkt an der Eckposition von Sensorknoten 3 (Eckposition: Abb. 5.6 rechts) platziert.

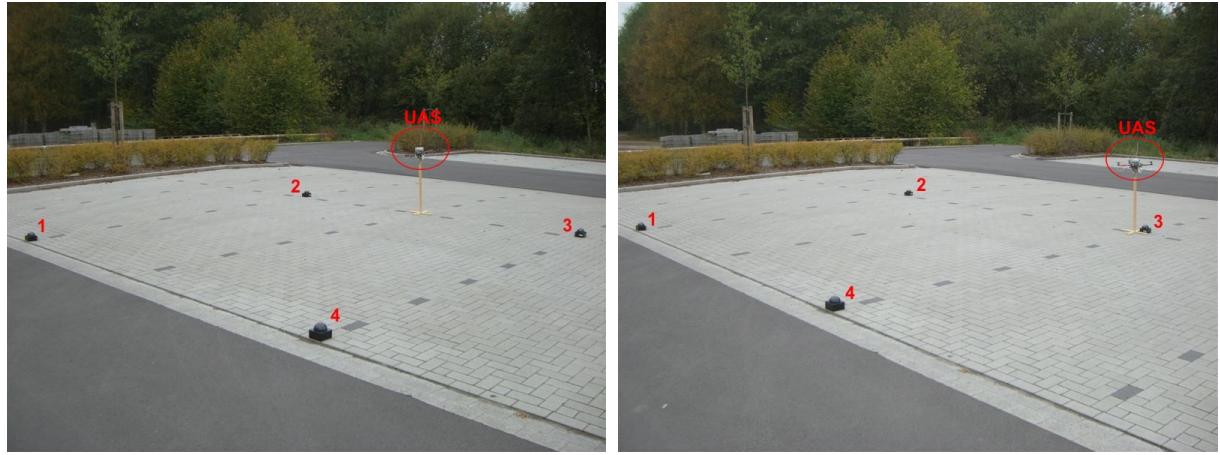


Abb. 5.6: Messaufbau: Quadrokopter an Rückposition (links) - Quadrokopter an Eckposition (rechts)

In der Abb. 5.7 ist der Verlauf der Messdaten von Knoten 1 zu den beiden unterschiedlichen Positionen zu erkennen.

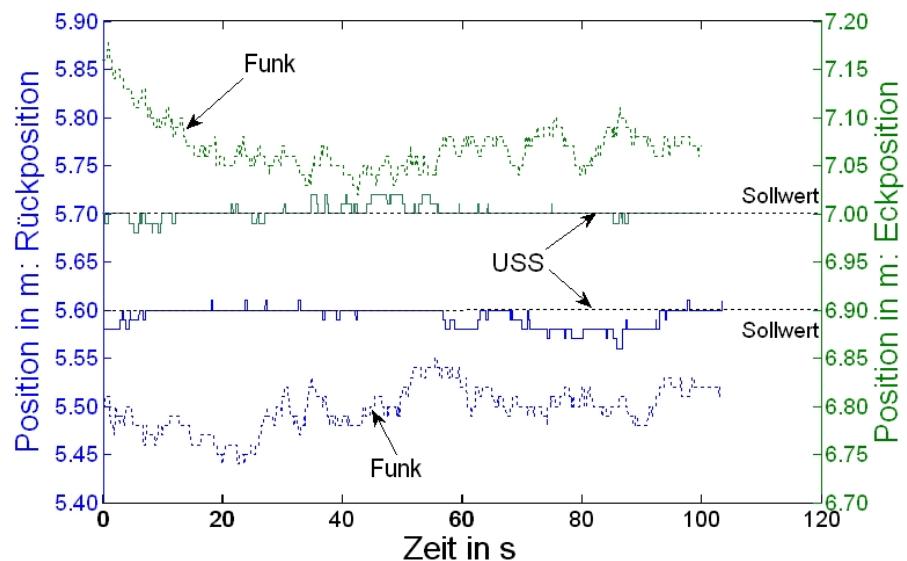


Abb. 5.7: Abstandsdaten vom Knoten 1: UAS an Rückposition und an Eckposition

Der Messfehler bei der Funklaufzeitmessung ist nicht konstant, sondern verändert sich an unterschiedlichen Positionen (Rückposition:  $-0.15m$ , Eckposition:  $+0.2m$ ), während er bei der Ultraschallmessung mit ca.  $0.03m$  konstant ist. Verschiedene Messungen haben gezeigt, dass sogar sich in der Nähe befindliche Gegenstände die Messgenauigkeit der Funklaufzeitmessung beeinflussen können. In einigen Fällen war der Messfehler bis zu  $1m$  groß. Es wird deutlich, dass die Genauigkeit der Funklaufzeitmessung (ca.  $1m$ ) zwar besser ist als die der GPS-Daten (ca.  $10m$ ), sie aber nicht ausreichend ist, um präzise auf

einem Punkt zu landen. Aus diesem Grund werden, wie in Abschnitt 6.8 erläutert, bei dem Landeverfahren diese drei Messsysteme miteinander kombiniert.

### 5.2.2 Positions berechnung

Sei  $N$  die Anzahl der verwendeten Sensorknoten,  $P_i = (x_i, y_i, z_i)$  die Position der Knoten in Bezug auf das Zentrum<sup>1</sup> und  $\delta_i$  der gemessene Abstand zum UAS, dann kann die aktuelle Position des UASs durch folgendes Gleichungssystem beschrieben werden.

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = \delta_i^2, 1 \leq i \leq N, N \geq 3 \quad (5.1)$$

Auf Grund der Tatsache, dass in den meisten Fällen die Landefläche planar und horizontal ausgerichtet ist, wurde eine flache Anordnung der Sensorknoten gewählt. Das heißt, dass sich alle Knoten in einer Ebene befinden. Für  $z_i = 0$  und  $N = 3$  ergibt sich dann z.B. für  $x$  folgende allgemeine Lösung<sup>2</sup>:

$$\begin{aligned} x = & -\frac{1}{2} \frac{1}{-x_2y_1 + x_3y_1 - x_3y_2 - y_3x_1 + y_3x_2 + y_2x_1} \\ & (x_1^2y_3 - x_1^2y_2 - y_1x_3^2 + y_1\delta_3^2 - x_2^2y_3 + y_1^2y_3 + y_1x_2^2 + y_2x_3^2 - y_1\delta_2^2 + \\ & \delta_2^2y_3 - y_2^2y_3 + y_1y_2^2 - y_1^2y_2 - y_1y_3^2 + y_2y_3^2 - \delta_1^2y_3 - y_2\delta_3^2 + \delta_1^2y_2) \end{aligned} \quad (5.2)$$

Des Weiteren wird angenommen, dass die vier verwendeten Sensorknoten in einem Quadrat angeordnet sind. Diese Grundannahmen entsprechen dem vorgestellten Messsystem (vgl. Abb. 5.2) und führen für  $N = 4$  und  $z_i = 0$  zu folgendem vereinfachten Gleichungssystem:

$$(x + \delta)^2 + (y - \delta)^2 + z^2 = \delta_1^2 \quad (5.3)$$

$$\wedge (x - \delta)^2 + (y - \delta)^2 + z^2 = \delta_2^2 \quad (5.4)$$

$$\wedge (x - \delta)^2 + (y + \delta)^2 + z^2 = \delta_3^2 \quad (5.5)$$

$$\wedge (x + \delta)^2 + (y + \delta)^2 + z^2 = \delta_4^2 \quad (5.6)$$

wobei  $\delta$  der Abstand zwischen dem Mittelpunkt des Vierecks und der Knoten in x- bzw. y-Richtung ist.

---

<sup>1</sup>Koordinatensystem aus Abb. 5.2

<sup>2</sup>Die Lösung für  $y$  ist ähnlich

Um das überbestimmte Gleichungssystem (bestehend aus 5.3 - 5.6) zu lösen, kann z.B. der klassische Weg der Singulärwertzerlegung [KL80] verwendet werden. Aber um die Berechnungszeit zu verkürzen und aus der Tatsache, dass wenn ein Knoten ausfällt, die anderen drei Knoten die Position alleine berechnen müssen, wurde der folgende Lösungsweg verwendet.

Das Gleichungssystem wird dazu in vier Gleichungssysteme mit 3 Gleichungen aufgeteilt und jedes separat gelöst. Die einzelnen Lösungen werden dann über das arithmetische Mittel miteinander kombiniert. Das Ergebnis des Gleichungssystems bestehend aus den Gleichungen 5.3 - 5.5 ist:

$$x = -\frac{1}{4} \frac{\delta_2^2 - \delta_1^2}{\delta} \quad (5.7)$$

$$y = -\frac{1}{4} \frac{\delta_3^2 - \delta_2^2}{\delta} \quad (5.8)$$

$$z = \sqrt{\delta_1^2 - (x + \delta)^2 - (y - \delta)^2} \quad (5.9)$$

Die Lösungen der anderen drei Gleichungssysteme werden analog berechnet.

In den Abbildungen 5.8, 5.9 und 5.10 sind die Resultate der Lokalisation mit Hilfe des Bakensystems zu sehen. Dabei sind sowohl die Ergebnisse der Funklaufzeitmessung als auch die der Ultraschallmessung dargestellt.

In Abb. 5.8 wurde bis zur Sekunde 50 die Position über die Funklaufzeit und danach mit Hilfe der Ultraschallsensoren berechnet. Während dieser Messung befand sich das UAS im Zentrum. Es ist offensichtlich, dass bei der Ultraschallmessung sowohl das Messrauschen geringer als auch die Lokalisation genauer ist.

Abb. 5.9 zeigt ein ähnliches Ergebnis. Bei dieser Messung stand das UAS in der Eckposition. Bis Sekunde 40 wurde die Position mit Hilfe der Funklaufzeit berechnet. Auch hier ist zu erkennen, dass die Positions berechnung über die Ultraschallsensoren weniger verrauscht und genauer ist.

Die Genauigkeit der Positions berechnung mit Hilfe der Ultraschallsensoren in x- und y-Richtung liegt bei ca. 0.05m und ist unabhängig von der aktuellen Position des UASs, wobei die Genauigkeit der Funklaufzeitmessung von 0.2 – 0.3m (in diesem Experiment) bis hin zu 1m variieren kann.

Abb. 5.10 zeigt die Ergebnisse der korrespondierenden Höhenberechnung.

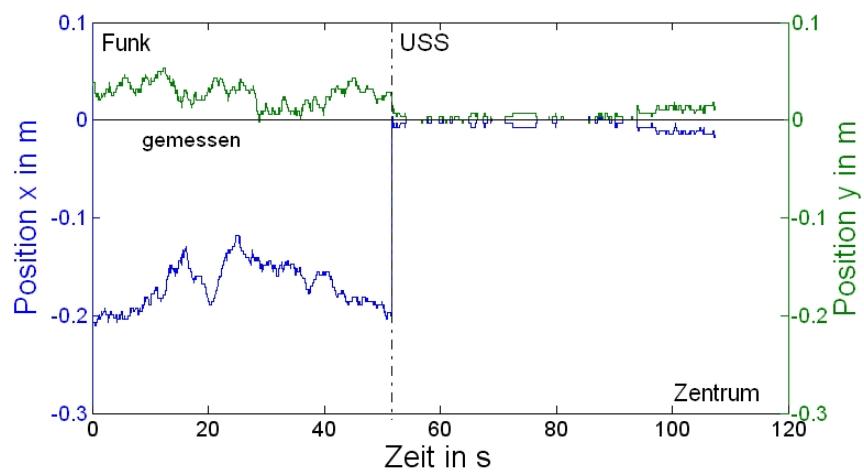


Abb. 5.8: Lokalisation mit Funk und Ultraschall: UAS im Zentrum

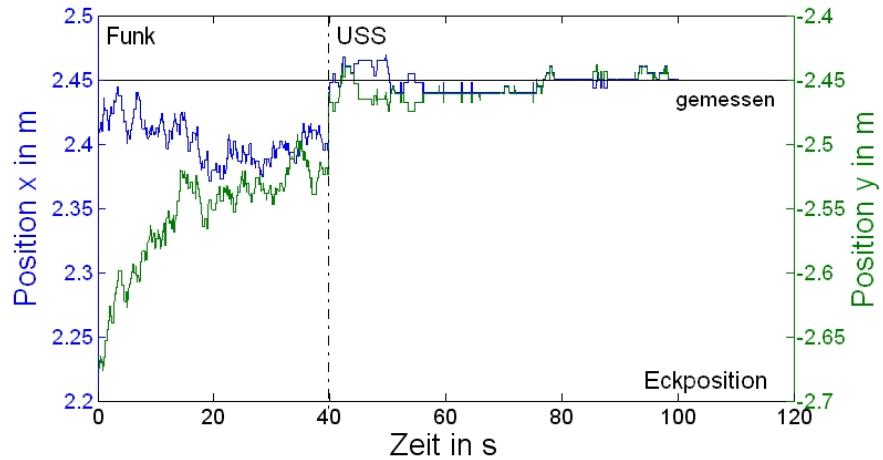


Abb. 5.9: Lokalisation mit Funk und Ultraschall: UAS in Eckposition

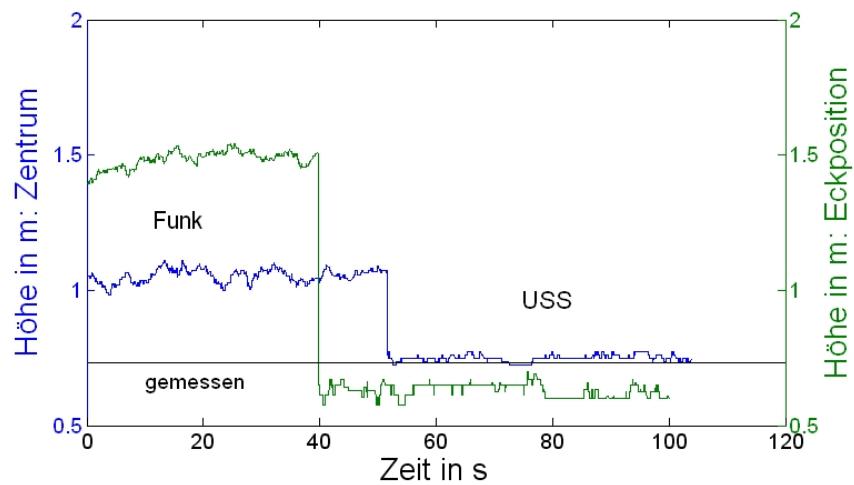


Abb. 5.10: Höhenbestimmung mit Funk und Ultraschall

Bei der Höhenberechnung ist zu erkennen, dass auch hier die Genauigkeit der Ultraschallmessung genauer ist als die der Funklaufzeitmessung, sie aber nicht an die Genauigkeit der x- und y-Messung herankommt. Das liegt vor allem daran, dass zur Berechnung der Höhe die Ergebnisse der x- und y-Position verwendet werden (siehe Gl. (5.9)) und sich dadurch die Abweichungen akkumulieren.

Die Genauigkeit der Höhenberechnung mit Hilfe der Ultraschallsensoren liegt bei ca.  $0.1 - 0.2m$  und ist damit etwas genauer als die der Barometermessung mit ca.  $0.2m$  (siehe Abschnitt 4.3.1). Allerdings kann die Genauigkeit bei der Funklaufzeitmessung fast bis zu  $1m$  schwanken, so dass für die Experimente immer das Barometer für die Höhenberechnung verwendet wurde.

### 5.2.3 Filterung

Wie bei den GPS-Signalen (vgl. Kapitel 4.3.2) ist auch bei der Berechnung der Position über die Sensorknoten eine zeitliche Verzögerung von ungefähr  $600ms$  vorhanden. Diese Verzögerung  $t_p$  setzt sich aus den Signallaufzeiten der Funksignale  $t_r$  und des Schalls  $t_u$  sowie aus den Berechnungszeiten  $t_c$  zusammen:  $t_p = t_u + t_r + t_c$ . Analog zum GPS-Filter wird auch bei der Berechnung der Position durch die Sensorknoten folgender Kalman-Filter verwendet:

$$p_{t+1} = p_t + K \cdot (p_t^u - p_{t-t_p}) \quad (5.10)$$

wobei  $K$  der Kalman-Faktor ist, mit  $0 \leq K \leq 1$ .

### 5.2.4 Diskussion

Neben den genannten positiven Eigenschaften dieses Messsystems gibt es aber auch einen Nachteil. Mit diesem System ist es nicht möglich, die Ausrichtung des UASs (Gierwinkel) bezüglich der Landeplattform zu ermitteln. Diese Tatsache kann vor allem bei Landungen auf metallischen Flächen ein Problem darstellen, da durch magnetische Einflüsse eine Missweisung des verwendeten Kompass entstehen und durch dieses System nicht korrigiert werden kann. Solche Missweisungen können so stark sein (bis zu  $180^\circ$ ), dass ein kontrolliertes Fliegen nicht mehr möglich ist. Um dieses zu verhindern, könnte man zum einen komplementäre Sensoriken einsetzen. Eine weitere Möglichkeit ist es, sich für die Zeit des Landevorgangs nicht auf den Kompass, sondern nur auf die internen Gyroskope zu verlassen.

# Kapitel 6

## Regelalgorithmen

In diesem Kapitel werden verschiedene Regelalgorithmen vorgestellt. Eine Übersicht über die Reglerstruktur mit den im folgenden beschriebenen Reglern ist in Abb. 6.1 dargestellt.

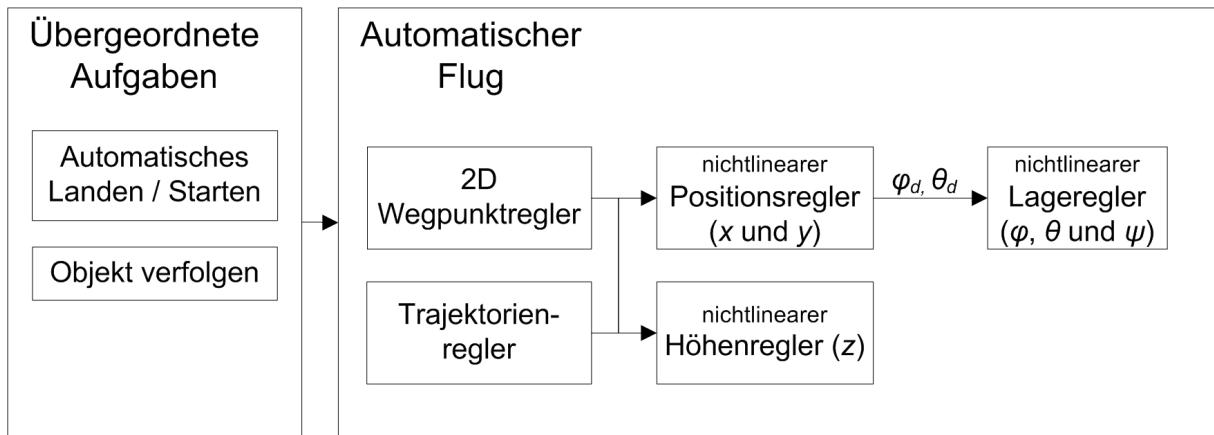


Abb. 6.1: Übersicht über die entwickelten Regler: Lageregler (siehe Abschnitt 6.1), 2D-Wegpunktregler (6.3), Trajektorienregler (6.4), Objekt verfolgen (6.6), Automatischer Start (6.7), Automatische Landung (6.8)

Als erstes wird ein nichtlinearer Lageregler vorgestellt, durch den auch permanente Regelabweichungen ausgeglichen werden können. Der Fokus liegt aber auf einem 2D-Wegpunktregler, einem Trajektorienregler und einem Regelalgorithmus zum automatischen Landen. Sowohl der 2D-Wegpunktregler als auch der Trajektorienregler können eingesetzt werden, um zwischen Wegpunkten zu fliegen. Sie haben aber unterschiedliche Eigenschaften und Fähigkeiten, so dass abhängig von den aktuellen Anforderungen entweder der eine oder der andere eingesetzt wird. Der 2D-Wegpunktregler ist dafür entwickelt worden, möglichst schnell zwischen Wegpunkten zu fliegen. Er bietet die Möglichkeit, den Quadrokopter ohne Geschwindigkeitsbegrenzung zu fliegen und dabei sowohl die gewünschte Höhe zu halten als auch auf der Sollstandlinie zu bleiben. Unter der Sollstandlinie versteht man dabei

die kürzeste Verbindung zwischen zwei Wegpunkten. Der Trajektorienregler wird dagegen dann eingesetzt, wenn es darum geht, sich zu jedem Zeitpunkt mit größtmöglicher Genauigkeit auf einer vorgegebenen Trajektorie zu befinden. Trajektorien können dabei Kreise, Splines oder andere 3D-Figuren im Raum sein. Der Trajektorienregler ermöglicht es des Weiteren, Fahrzeuge zu folgen. Als letztes wird ein Start- und ein Landeargorithmus vorgestellt. Beide Algorithmen ermöglichen es, sowohl aus einer Box aus zu starten als auch zu landen. Als letztes wird ein Verfahren zum Landen auf sich bewegenden Oberflächen vorgestellt.

## 6.1 Lageregelung

Wie in Abschnitt 2.3 beschrieben ist eine funktionierende Lageregelung Grundlage aller darauf aufbauender Positionsregler. Der verwendete Lageregler wurde in Grundzügen von [Kem07] übernommen. Die Theorie des Schwerpunktes wurde aus der Regelung entfernt und durch einen Integralregler ersetzt, da die Anforderung bestand, auch Lasten zu transportieren und absetzen zu können. Diese Tatsache verändert jedes Mal den Schwerpunkt des Systems, so dass der Regler angepasst werden müsste. Des Weiteren ist es nicht trivial, den Schwerpunkt des Systems zu bestimmen, der zusätzlich durch fertigungsbedingte Toleranzen variieren kann.

Der verwendete Regler kann in vier Phasen eingeteilt werden. Am Beispiel des Rollwinkels  $\phi$  wird er im folgenden erläutert. Der Regler für den Nickwinkel  $\theta$  ist analog und der Regler für den Gierwinkel  $\psi$  unterscheidet sich nur durch die Abwesenheit des I-Anteils.

- **Phase 1: Berechnung der Regeldifferenz**

Zu Beginn wird die Regeldifferenz  $e_\phi$  berechnet durch

$$e_\phi = \phi_d - \phi_c, \quad (6.1)$$

wobei  $\phi_d$  der Soll- und  $\phi_c$  der Istwinkel ist.  $-\pi < e_\phi < \pi$

- **Phase 2: P-Regler**

In dieser Phase wird der Stellwert  $y_\phi$  zum ersten Mal berechnet durch

$$y_\phi^P = k_p^\phi \cdot e_\phi \quad (6.2)$$

wobei  $k_p^\phi$  der Regelparameter ist.

- **Phase 3: Dämpfungen**

Zur Stabilisierung der Regelung wird eine Dämpfung eingebaut, die der Drehgeschwindigkeit  $\dot{\phi}_c$  entgegenwirkt.

$$y_\phi^{PD} = y_\phi^P - k_d^\phi \cdot \dot{\phi}_c \quad (6.3)$$

wobei  $k_d^\phi$  der Regelparameter ist.

- **Phase 4: I-Anteil**

Der I-Anteil hat die Aufgabe, permanente Störungen auszugleichen. Zum einen soll dieses schnell passieren, zum anderen darf das System nicht instabil werden. Aus diesem Grund wird ein modifizierter I-Regler verwendet. Die Änderungen bestehen darin, dass der Parameter  $k_i^\phi$  nicht konstant ist, sondern von der Drehrichtung von  $\phi$  abhängt.

$$k_i^\phi = \begin{cases} k_{ih}^\phi, e_\phi \cdot \dot{\phi}_c < 0 \\ k_{il}^\phi, \text{sonst} \end{cases} \quad (6.4)$$

mit  $k_{ih}^\phi > k_{il}^\phi$

Der Stellwert  $y_\phi$  wird dann wie folgt berechnet.

$$y_\phi = y_\phi^{PD} + \int k_i^\phi \cdot e_\phi dt \quad (6.5)$$

Danach wird überprüft, ob der I-Anteil in einem bestimmten Intervall bleibt.  $-I_{MAX} < \int k_i^\phi \cdot e_\phi dt < I_{MAX}$ .

Der Stellwert  $y_\phi$  wird entsprechend dem Modell (vgl Abb. 2.3) auf die Motorsteller gegeben.  $|\underline{F}_1|$  und  $|\underline{F}_4|$  werden um  $\frac{y_\phi}{2}$  reduziert,  $|\underline{F}_2|$  und  $|\underline{F}_3|$  um  $\frac{y_\phi}{2}$  erhöht.

## 6.2 Berechnung von „Bearing“ und „Distance“

Für die Navigation zwischen GPS-Wegpunkten  $WP$ , bestehend aus einem Längengrad  $WP^{lon}$  und einem Breitengrad  $WP^{lat}$ , ist es notwendig, sowohl die Entfernung  $d$  zum Ziel als auch die Peilung  $\omega$  auf das Ziel zu berechnen. Für diese Berechnung gibt es mehrere Ansätze (z.B. Vincenty Formel), die zum Teil in [HWWL03] beschrieben werden. Sie unterscheiden sich in ihrer Komplexität und in dem Modell der Erde (Kugel, Ellipse usw.).

In dieser Arbeit werden zwei Varianten verwendet. Die erste Methode kann verwendet werden, wenn der Abstand zwischen den beiden Wegpunkten  $WP_1$  und  $WP_2$  gering ist, also die Erde als eine flache Ebene angenommen werden kann. Bei dieser Methode wird nur weniger Rechenzeit benötigt, und sie ist deshalb ideal z.B. für die 3D-Trajektorienregelung (siehe 6.4), bei der die Abstände zwischen den Wegpunkten maximal einige wenige Meter beträgt. Die zweite Methode dagegen wird verwendet, wenn die Abstände zwischen den Wegpunkten „sehr“ groß sind und somit die Erdkrümmung mit berücksichtigt werden muss. Um die Komplexität trotzdem gering zu halten, wird die Erde als Kugel angenähert. Diese Berechnung wird z.B. bei dem 2D-Wegpunktregler (siehe 6.3) verwendet, da hier immer die Entfernung zum Zielpunkt berechnet wird. Im folgenden werden beide Verfahren kurz beschrieben.

### 6.2.1 Kurze Distanzen

Bei dieser Methode wird die Erde als eine Ebene angenommen. Zuerst werden die Abstände in  $m$  zwischen den Breitengraden  $d^{lat}$  und den Längengraden  $d^{lon}$  berechnet.

$$d^{lat} = (WP_1^{lat} - WP_2^{lat}) \cdot \frac{\pi \cdot r_{\text{äq}}}{180} \quad (6.6)$$

$$d^{lon} = (WP_2^{lon} - WP_1^{lon}) \cdot \frac{\pi \cdot \cos(WP_1^{lat}) \cdot r_{\text{äq}}}{180} \quad (6.7)$$

mit  $r_{\text{äq}} = 6378137.0m$ . Aus diesen beiden Abständen werden dann die Gesamtdistanz  $d$  und der Winkel  $\omega$  berechnet durch

$$\begin{aligned} d &= \sqrt{(d^{lat})^2 + (d^{lon})^2} \\ \omega &= \text{atan2}(-d^{lon}, -d^{lat}) \end{aligned}$$

wobei  $\text{atan2}$  eine Funktion ist, die den Arkustangens so berechnet, dass der Winkel sich im korrekten Quadranten befindet.

$$\text{atan2}(y, x) := \begin{cases} \arctan \frac{y}{x}, & x > 0 \\ \arctan \frac{y}{x} + \pi, & x < 0, y \geq 0 \\ \arctan \frac{y}{x} - \pi, & x < 0, y < 0 \\ +\pi/2, & x = 0, y > 0 \\ -\pi/2, & x = 0, y < 0 \\ 0, & x = 0, y = 0 \end{cases} \quad (6.8)$$

### 6.2.2 Lange Distanzen

Bei dieser Methode wird die Erde als Kugel angenähert. Die Distanz  $d$  ist dann die Länge der Orthodrome (ugs. auch Luftlinie genannt) zwischen den beiden Wegpunkten auf der Kugel und  $\omega$  der Kurswinkel (auch Azimut) (Winkel zwischen Nordrichtung und Zielrichtung).

Der Kurs berechnet sich wie folgt:

$$\omega = -\text{atan}2(y, x) \quad (6.9)$$

mit

$$x = \cos(WP_1^{lat}) \cdot \sin(WP_2^{lat}) - \sin(WP_1^{lat}) \cdot \cos(WP_2^{lat}) \cdot \cos(WP_1^{lon} - WP_2^{lon}) \quad (6.10)$$

$$y = -\sin(WP_1^{lon} - WP_2^{lon}) \cdot \cos(WP_2^{lat}) \quad (6.11)$$

Die Entfernung durch:

$$d = \sqrt{d_x^2 + d_y^2 + d_z^2} \quad (6.12)$$

mit

$$d_x = (\cos(WP_1^{lat}) \cdot \cos(WP_1^{lon}) - \cos(WP_2^{lat}) \cdot \cos(WP_2^{lon})) \cdot r_{pol} \quad (6.13)$$

$$d_y = (\cos(WP_1^{lat}) \cdot \sin(WP_1^{lon}) - \cos(WP_2^{lat}) \cdot \sin(WP_2^{lon})) \cdot r_{pol} \quad (6.14)$$

$$d_z = (\sin(WP_1^{lat}) - \sin(WP_2^{lat})) \cdot r_{pol} \quad (6.15)$$

mit  $r_{pol} = 6356795.0$

### 6.2.3 Unterschiede

Mit der Methode für die langen Distanzen ist das Ergebnis bezüglich der Entfernung exakter, vor allem in der Nähe der Pole. Ein Nachteil liegt aber in der Komplexität der Berechnung. Auf der Avionik (siehe Kapitel 4.2.1) dauert die Methode für die längeren Distanzen ca.  $1ms$ , während die kurze  $200\mu s$  dauert; also fünf mal länger.

## 6.3 2D-Wegpunktregler

### 6.3.1 Höhenregelung

Dieses Kapitel beschäftigt sich mit der für den automatischen Flug notwendigen Höhenregelung.

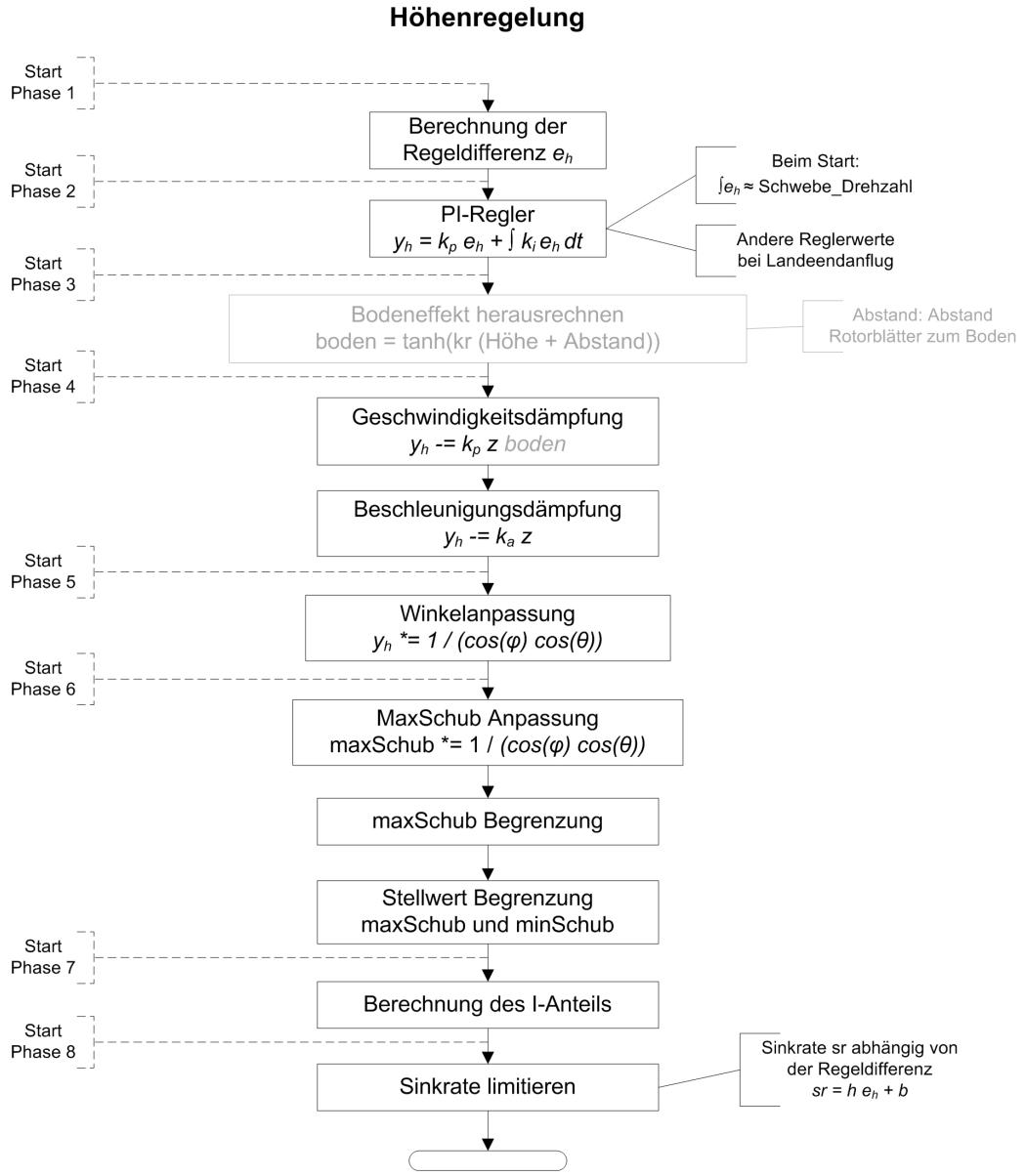


Abb. 6.2: Höhenregler

Der Höhenregler hat die Aufgabe, sowohl während des stationären Fluges als auch während des Abfliegens von Wegpunkten eine vorgegebene Sollhöhe  $z_c$  zu halten oder bei Höhenänderungen die neue Zielhöhe  $z_d$  „schnellstmöglich“ zu erreichen. Dabei ist aber

darauf zu achten, dass vor allem beim Sinkflug die vorgegebene Sollhöhe nicht unterschritten werden darf.

Das Grundprinzip der Höhenregelung besteht darin, dass zur Ausregelung der Höhendifferenz  $e_h$  immer alle vier Rotoren simultan und gleichverteilt angesteuert werden (siehe 2.3). Das bedeutet, dass der am Ende der Höhenregelung berechnete Stellwert  $y_h$  auf alle vier Rotoren gelegt wird.

In Abb. 6.2 ist das Flussdiagramm des Höhenreglers dargestellt. Der Regelalgorithmus kann in acht Phasen aufgeteilt werden, die im weiteren genauer beschrieben werden.

- **Phase 1: Berechnung der Regeldifferenz**

Im ersten Schritt wird die Höhendifferenz  $e_h$  berechnet:

$$e_h = z_d - z_c, \quad (6.16)$$

wobei  $z_d$  die Soll- und  $z_c$  die Isthöhe ist.

- **Phase 2: PI-Regler**

Der Wert  $e_h$  wird als Eingangswert für einen modifizierten PI-Regler verwendet. Der Unterschied zu einem klassischen PI-Regler liegt in der Art, wie der integrale Anteil des Reglers berechnet wird (siehe Phase 7). Da in dem Wert  $\int k_i^h \cdot e_h dt$  der Grundschatz enthalten ist, der notwendig ist, den Flugroboter in der Schwebe zu halten, muss er zu Beginn mit einem Wert initialisieren werden, der nahe dem benötigten Grundschatz liegt, damit das Integral  $e_h$  nicht komplett von Null aufgebaut werden muss. Dieser Wert muss experimentell oder simulativ ermittelt werden. Im Detail wird der Stellwert  $y_h$  wie folgt berechnet:

$$y_h^{PI} = k_p^h \cdot e_h + \int k_i^h \cdot e_h dt, \quad (6.17)$$

wobei  $k_p^h$  der Regelparameter für den proportionalen und  $k_i^h$  der Regelparameter für den integralen Anteil des Reglers sind.

- **Phase 3: Dämpfungen**

In dieser Phase werden zwei unterschiedliche Arten von Dämpfungen zu dem zuvor berechneten Wert  $y_h^{PI}$  addiert. Dabei handelt es sich um eine Geschwindigkeits- und um eine Beschleunigungsdämpfung.

- *Geschwindigkeitsdämpfung*

Die Geschwindigkeitsdämpfung ist essentiell, um ein Oszillieren um  $e_d$  zu ver-

hindern und wird wie folgt berechnet.

$$y_h^{PId} = y_h^{PI} - (k_d^h \cdot \dot{z}_c \cdot h_{boden}), \quad (6.18)$$

wobei  $k_d^h$  den Regelparameter für die Geschwindigkeitsdämpfung darstellt.  $h_{boden}$  ist ein Term zum Überwinden des Bodeneffekts (siehe Phase 4).

- *Beschleunigungsdämpfung*

Die Beschleunigungsdämpfung hat die Aufgabe, abrupt auftretenden Höhen-schwankungen entgegenzuwirken, die zum Beispiel durch Windböen auftreten können und wird berechnet durch

$$y_h^{PId^2} = y_h^{PId} - (k_a^h \cdot \ddot{z}_c), \quad (6.19)$$

wobei  $k_a^h$  den Regelparameter für die Beschleunigungsdämpfung darstellt.

- **Phase 4: Bodeneffekt**

Um den Bodeneffekt zu überwinden, werden die Ergebnisse aus der Arbeit von [Kem07] verwendet. Es wird der Korrekturfaktor  $h_{boden}$  eingeführt.

$$h_{boden} = \tanh(p_g \cdot (z + d_{gr})) = \frac{2}{\exp(-2 \cdot p_g \cdot (z + d_{gr}) + 1)} - 1 \quad (6.20)$$

mit  $d_{gr} > 0$  Abstand zwischen Rotorblätter und Boden und  $p_g > 0$ . Der Term  $z + d_{gr}$  ist abhängig von der aktuellen Höhe und verhindert, dass die Funktion  $\tanh$  Null wird. Dieser Term wird mit dem Geschwindigkeitsdämpfungsanteil multipliziert (siehe Gl. (6.18)).

- **Phase 5: Winkelanpassung**

In Phase 5 werden die aktuellen Lagewinkel  $\phi$  und  $\theta$  mit berücksichtigt. Sobald  $\phi$  und/oder  $\theta$  ungleich null sind, zeigt der Schubvektor  $F = F1 + F2 + F3 + F4$  (siehe Abb. 2.3) nicht mehr exakt nach oben, so dass  $y_h$  erhöht werden muss, um den gleichen Auftrieb zu erzeugen. Aus diesem Grund wird der Wert  $y_h PId^2$  wie folgt angepasst.

$$y_h^W = y_h^{PId^2} \cdot \frac{1}{\cos \phi \cdot \cos \theta}, \quad (6.21)$$

mit  $(\cos \phi \cdot \cos \theta) \neq 0$

- **Phase 6: Schubbegrenzung und Schubanpassung**

- *Schubbegrenzung*

In diesem Abschnitt wird der vorher berechnete Stellwert  $y_h^w$  sowohl durch

ein Maximum ( $thrust_{max}^{height}$ ) wie auch durch ein Minimum ( $thrust_{min}^{height}$ ) begrenzt. Dieses ist aus zwei Gründen notwendig. Zum einen ist es essentiell, dass  $thrust_{max}^{height} < thrust_{max}$  und  $thrust_{min}^{height} > thrust_{min}$  ist, damit für die Lageregelung noch genug Reserveschub vorhanden ist. Zum zweiten ist die untere Grenze nützlich, um ein zu schnelles Sinken des Flugroboters zu verhindern. Dieses ist essentiell, da ein zu schnelles Absinken dazu führen kann, dass der Flugroboter in seinen eigenen Luftstrom gerät, der durch die Rotoren erzeugt wird und dadurch instabil werden kann.

- *Schubanpassung*

Wie in Phase 5 beschrieben, wird der Stellwert  $y_h^w$  der aktuellen Lage des Flugroboters angepasst. Damit nun der Flugroboter auch bei großen Werten von  $\phi$  und/oder  $\theta$  nicht absinkt, kann auch der Wert  $thrust_{max}^{height}$  in der selben Art und Weise der aktuellen Lage angepasst werden.

$$thrust_{max}^{height} = thrust_{max}^{height} \cdot \frac{1}{\cos \phi \cdot \cos \theta}, \quad (6.22)$$

mit  $(\cos \phi \cdot \cos \theta) \neq 0$  und  $thrust_{max}^{height} < thrust_{max}$ .

- **Phase 7: I-Anteil berechnen**

In dieser Phase wird der integrale Teil  $\int e_h dt$  des PI-Reglers bestimmt. Aber im Vergleich zu einem klassischen I-Regler ist der Reglerparameter  $k_i^h$  nicht konstant, sondern abhängig von der Bewegungsrichtung.

$$k_i^h = \begin{cases} k_{ih}^h, & e_h \cdot \dot{z}_c < 0 \\ k_{il}^h, & \text{sonst} \end{cases} \quad (6.23)$$

mit  $k_{ih}^h > k_{il}^h$ . Das führt dazu, dass Windänderungen oder Gewichtsänderungen schnell entgegengewirkt werden kann, ohne die Gefahr, ein instabiles Verhalten zu bekommen.

- **Phase 8: Sinkrate limitieren**

Als letztes wird die Sinkgeschwindigkeit  $\dot{z}_c$  ( $\dot{z}_c < 0$ ) limitiert. Ziel ist es, ein Unterschreiten der Zielhöhe zu verhindern. Aber trotzdem soll eine Änderung von  $z_d$  so schnell wie möglich ausgeregelt werden können. Aus diesem Grund ist die zulässige

Sinkrate  $\dot{z}_{sr}$  ( $\dot{z}_{sr} > 0$ ) variabel in Abhängigkeit von  $e_h$ .

$$\dot{z}_{sr} = \begin{cases} \dot{z}_{sr}^{MIN}, & e_h > OBERE\_GRENZE \\ \dot{z}_{sr}^{MAX}, & e_h < UNTERE\_GRENZE \\ h \cdot e_h + b, & sonst \end{cases} \quad (6.24)$$

wobei  $0 > OBERE\_GRENZE > UNTERE\_GRENZE$ ,

$$h = (\dot{z}_{sr}^{MIN} - \dot{z}_{sr}^{MAX}) / (OBERE\_GRENZE - UNTERE\_GRENZE) \quad (6.25)$$

$$b = \dot{z}_{sr}^{MIN} - OBERE\_GRENZE \cdot h \quad (6.26)$$

Der Stellwert  $y_h^w$  wird dann wie folgt angepasst:

$$y_h = y_h^w - (\dot{z} + \dot{z}_{sr}) \cdot k_{sr}, \quad (6.27)$$

falls  $\dot{z} < -\dot{z}_{sr}$ , wobei  $k_{sr}$  der Regelparameter ist.

Am Ende muss überprüft werden, ob durch diese Berechnung  $y_h > thrust_{max}^{height}$  ist.  
Ist dieses der Fall muss  $y_h = thrust_{max}^{height}$  gesetzt werden.

### 6.3.2 Positionsregelung

In Abb. 6.3 ist ein Flussdiagramm des 2D-Wegpunktreglers zu sehen, der in der Lage ist, eine der in 1.2 erwähnten Anforderungen zu erfüllen, nämlich ohne Geschwindigkeitsbegrenzung in das Einsatzgebiet zu fliegen. In der Abb. sind einige Beschreibungen ausgegraut. Diese Teile werden in dem Kapitel 9 erklärt.

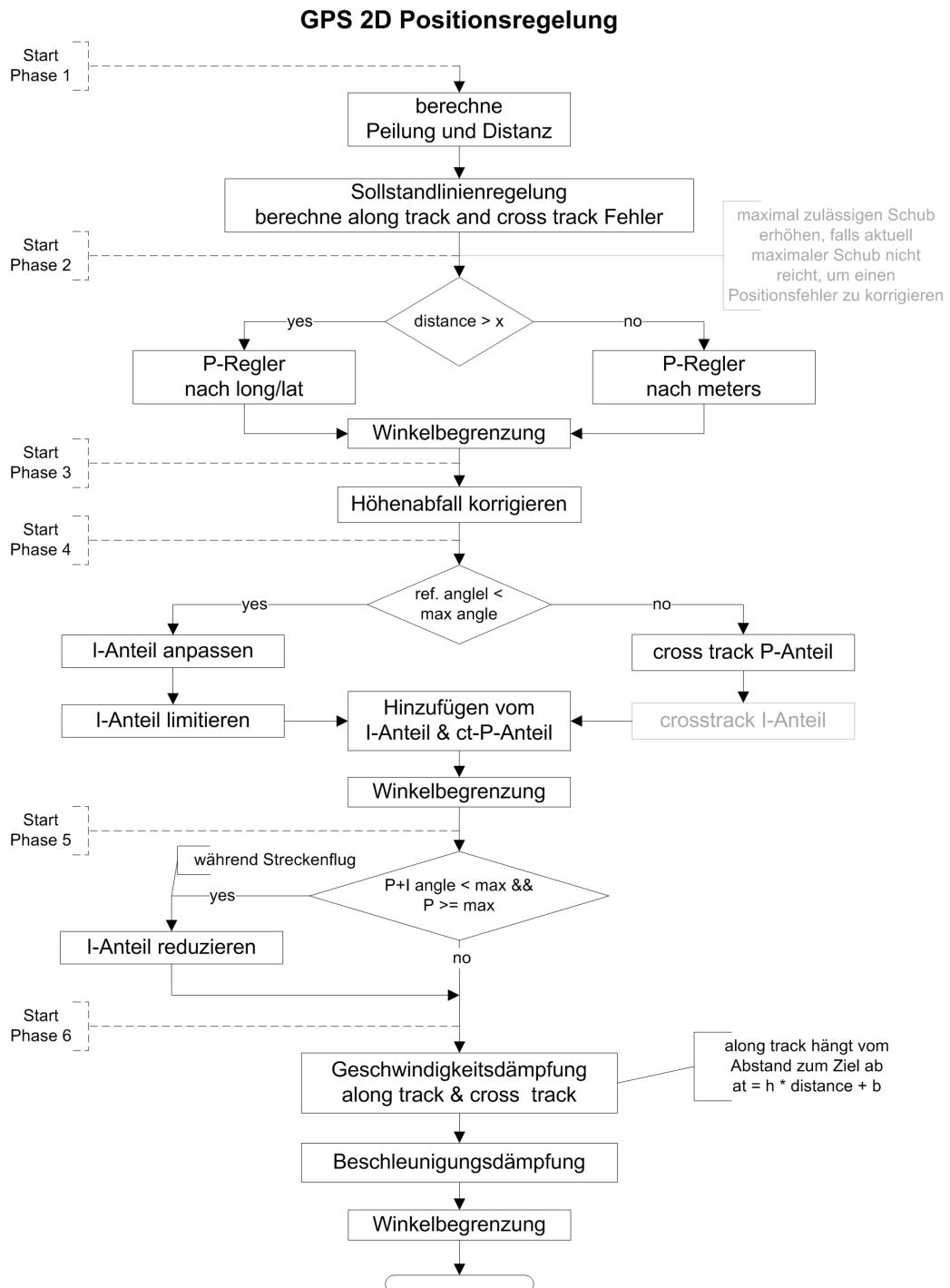


Abb. 6.3: 2D-Wegpunktregler

- **Phase 1: Berechnung von Peilung und Distanz**

Im ersten Teil der Regelung werden die Peilung  $\omega$  (bezüglich geographisch Nord) und die Distanz  $d$  (in Meter) von der aktuellen Position des UASs zum Ziel berechnet (siehe 6.2). Diese Werte werden direkt aus den Längengraden und Breitengraden bestimmt, die vom GPS-Empfänger ermittelt werden. Das Ziel dieses Reglers ist es, das UAS auf einer definierten Trajektorie zu halten, die durch eine Startposition  $sp$  und eine Zielkoordinate  $tp$  bestimmt wird.

Sobald eine neue Zielposition definiert wird, werden die neuen Werte  $\omega$  und  $d$ , die an der Startposition berechnet wurden, in den Variablen  $\omega_{start}$  und  $d_{start}$  gespeichert. Zusätzlich werden die Steigung  $m_b$  und die Verschiebungskonstante  $b_b$  einer linearen Funktion bestimmt, durch die der Punkt  $x_b$  (siehe Abb. 6.4) in Abhängigkeit von  $POT$  (siehe Gl. (6.31)) auf der Trajektorie verschoben wird, auf den das UAS zufliegen soll.

$$m_b = \frac{POT - 1}{d_{start} - d_0}, \quad (6.28)$$

mit  $0 < POT \leq 1$  und  $0 < d_0 < d_{start}$

$$b_b = POT - m_b \cdot d_{start} \quad (6.29)$$

wobei  $POT$  (POT: point on target) den maximal erlaubten Korrekturwinkel definiert, das heißt, wie stark das UAS auf die Trajektorie zuhält. Für  $POT = 0$  fliegt das UAS immer direkt aufs Ziel. Ist z.B.  $POT = \frac{1}{3}$  würde das UAS bei  $d \geq d_{start}$  auf den Punkt  $x_b$  zufliegen, der sich auf der Trajektorie  $\frac{1}{3}$  Richtung Ziel befindet. Mit kleiner werdendem  $d$  ( $d < d_{start}$ ) wandert der Punkt  $x_b$  immer weiter Richtung Zielpunkt.  $d_0$  bestimmt die Distanz, unter der keine Winkelkorrektur mehr stattfindet und das UAS direkt auf die Zielposition zuhält.

## Sollstandlinienregelung

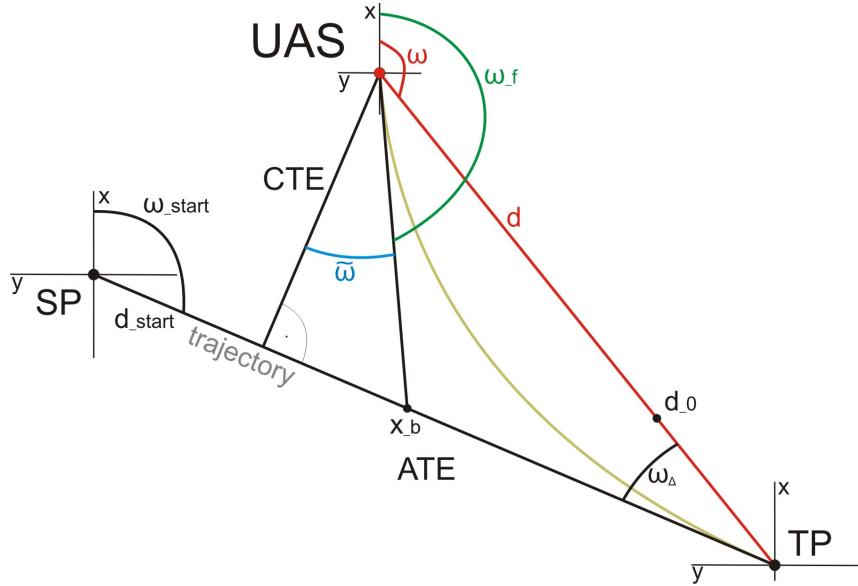


Abb. 6.4: Sollstandlinienregelung

Tritt nun der Fall auf, dass das UAS die definierte Trajektorie verlässt und die Distanz  $d$  den vorher definierten Wert  $d_0$  überschreitet, wird  $\omega$  unter Berücksichtigung von  $d$  angepasst. Dazu wird die Winkeldifferenz  $\omega_\Delta$  zwischen dem Startwinkel  $\omega_{start}$  und dem aktuellen Peilungswinkel  $\omega$  berechnet.

$$\omega_\Delta = \omega_{start} - \omega \quad (6.30)$$

Des Weiteren wird der „cross-track“-Fehler  $CTE$  und der „along-track“-Fehler  $ATE$  bestimmt, wobei der „cross-track“-Fehler der Abstand zwischen UAS und Trajektorie und der „along-track“-Fehler der Abstand zum Ziel auf der Trajektorie ist.

$$ATE = d \cdot \cos \omega_\Delta \quad (6.31)$$

$$CTE = d \cdot \sin \omega_\Delta \quad (6.32)$$

Aus 6.31 und 6.32 wird der Korrekturwinkel  $\tilde{\omega}$  berechnet.

$$\tilde{\omega} = \text{atan2}(ATE \cdot x_b, CTE) \quad (6.33)$$

$x_b$  definiert den Punkt auf  $ATE$ , auf den das UAS zuhalten soll und wird wie folgt berechnet:

$$x_b = b_b + m_b \cdot ATE, \quad (6.34)$$

mit  $POT < x_b \leq 1$ .

Am Ende wird dann der finale Peilwinkel  $\omega_f$  berechnet:

$$\omega_f = \omega - ((\pi - \omega_\Delta) - \tilde{\omega}) \quad (6.35)$$

Da der aktuelle Gierwinkel  $\psi_c$  bei der Regelung berücksichtigt werden muss, wird vorab einmalig der Peilwinkel korrigiert durch

$$\omega_\psi = \psi_c - \omega_f \quad (6.36)$$

### • Phase 2: P-Regler

Im zweiten Teil der Regelung werden die Solllagewinkel Roll  $\phi_d^P$  und Nick  $\theta_d^P$  aus den Werten  $\omega_\psi$  und  $d$  berechnet.

Es sei der Vektor<sup>1</sup>  $\underline{W}_d$  mit  $\underline{W}_d = \begin{pmatrix} \phi_d \\ \theta_d \end{pmatrix}$ .

Als Grundlage zur Berechnung dient ein klassischer P-Regler. Der Stellwert  $y_d$  wird berechnet durch

$$y_d = k_p^p \cdot d, \quad (6.37)$$

wobei  $k_p^p$  der Reglerparameter ist. Wie beim Höhenregler wird der Stellwert  $y_d$  nach oben begrenzt durch  $y_{max}^p$ . Dieser Wert beschreibt, welchen maximalen Lagewinkel der Flugroboter bei einer Positionsänderung oder -korrektur einnehmen darf.

Der Stellwert  $y_d$  wird dann auf die Sollwinkel  $\phi_d^P$  und  $\theta_d^P$  transformiert durch

$$\phi_d^P = \sin(\omega_\psi) \cdot y_d \quad (6.38)$$

$$\theta_d^P = \cos(\omega_\psi) \cdot y_d \quad (6.39)$$

### • Phase 3: Höhenabfall korrigieren

Um einen möglichen Höhenabfall zu korrigieren, der auftreten kann, wenn bei großen Roll- bzw. Nickwinkeln und der daraus resultierenden hohen Geschwindigkeit des Flugroboters die Höhenregelung nicht in der Lage ist, die Höhe zu halten, muss der Positionsregler eingreifen. Dieses geschieht durch folgende Maßnahme. Überschreitet  $e_h$  einen gewissen Wert  $e_h^{err}$  werden die Winkel  $\phi_d^P$  und  $\theta_d^P$  proportional zur

---

<sup>1</sup> $\widetilde{\underline{W}}_d$ ,  $\underline{W}_d^P$  usw. entsprechend

Höhendifferenz gleichmäßig verringert.

$$\tilde{\phi}_d^P = \phi_d^P \cdot h_{loss} \quad (6.40)$$

$$\tilde{\theta}_d^P = \theta_d^P \cdot h_{loss}, \quad (6.41)$$

mit  $h_{loss} = \frac{1}{(e_h - (e_h^{err} - 1))}$  falls  $e_h > e_h^{err}$ , wobei  $e_h^{err} > 1$

Die Korrektur von  $\underline{W}_d^P$  findet vor dem Hinzufügen der integralen Werte statt, damit bei Höhenänderungen ohne Positionsänderungen der I-Anteil trotzdem weiterhin Einfluss nehmen kann, um während der Steigphase nicht die Position zu verlieren.

- **Phase 4: I-Regler**

In dieser Phase werden zwei unterschiedliche Fälle unterschieden. Ist  $y_d = y_{max}^p$  befindet sich der Flugroboter im so genannten Streckenflug, ist  $y_d < y_{max}^p$ , befindet er sich in der Nähe des Ziels und muss die Zielposition  $zp$  halten.

- *Streckenflug*

Während des Streckenflugs ( $y_d = y_{max}^p$ ) ist darauf zu achten, dass der Flugroboter seine Trajektorie nicht verlässt. Zwar wird durch die Sollstandlinienregelung veranlasst, dass der Flugroboter wieder auf die Trajektorie zurückfliegt, er sollte aber vorher schon daran gehindert werden, die Trajektorie überhaupt zu verlassen. Aus diesem Grund wird während des Streckenflugs korrigierend auf den *CTE*-Wert eingewirkt. Zu diesem Zweck werden zwei Korrigierstellwerte  $y_y^{cross}$  und  $y_x^{cross}$  berechnet durch:

$$y_x^{cross} = \sin(\omega_{start}) \cdot k_p^{cross} \cdot CTE \quad (6.42)$$

$$y_y^{cross} = \cos(\omega_{start}) \cdot k_p^{cross} \cdot CTE \quad (6.43)$$

falls  $y_d = y_{max}^p$ , wobei  $k_p^{cross}$  der Regelparameter ist, mit  $k_p^{cross} < k_p^p$ .

Diese beiden Werte gehen in Formeln 6.48 und 6.49 ein.

- *I-Anteil*

Während des Positionshalten ( $y_d < y_{max}^p$ ) wird aus dem P-Regler ein PI-Regler. Dieser hat die Aufgabe, auch bei stetigen Winden die Zielposition zu erreichen und zu halten.

Zu diesem Zweck werden zu Beginn  $\omega_f$  und  $d$  umgerechnet in

$$d_x = \cos(\omega_f) \cdot d \quad (6.44)$$

$$d_y = \sin(\omega_f) \cdot d \quad (6.45)$$

und in dem Vektor  $\underline{d} = \begin{pmatrix} d_x \\ d_y \end{pmatrix}$  zusammengefasst. Diese Werte werden verwendet, um die Integrale  $\int k_i^p \cdot d_x dt$  und  $\int k_i^p \cdot d_y dt$  zu berechnen.

Wie beim Höhenregler ist der Reglerparameter  $k_i^p$  nicht konstant sondern abhängig von der Bewegungsrichtung. Die Berechnung ist für die x- und y-Richtung gleich und ist hier für die x-Richtung aufgeschrieben.

$$k_{ix}^p = \begin{cases} k_{ih}^p, & d_x \cdot \dot{x}_c < 0 \\ k_{il}^p, & \text{sonst} \end{cases} \quad (6.46)$$

mit  $k_{ih}^p > k_{il}^p$ .

Wie beim P-Regler müssen auch die Integrale nach oben hin begrenzt werden und zwar ebenfalls auf  $y_{max}^p$ . Es ist darauf zu achten, dass sich die Richtung des Vektors  $\underline{\int ddt} = \begin{pmatrix} \int k_{ix}^p \cdot d_x dt \\ \int k_{iy}^p \cdot d_y dt \end{pmatrix}$  durch die Begrenzung nicht ändert. Zu diesem Zweck findet die Begrenzung wie folgt statt:

$$\int k_{ix}^p \cdot d_x dt = \int k_{ix}^p \cdot d_x dt \cdot \left( y_{max}^p / \left| \underline{\int ddt} \right| \right), \quad (6.47)$$

wenn  $\left| \underline{\int ddt} \right| > y_{max}^p$ .  $\int k_{iy}^p \cdot d_y dt$  wird analog bestimmt.

Diese Werte werden dann zu dem vorher berechneten Vektor  $\widetilde{W}_d^P$  addiert, mit

$$\phi_d^{PI} = \widetilde{\phi}_d^P - \left( -\sin(\psi_c) \cdot \left( \int k_{ix}^p \cdot d_x dt + y_x^{cross} \right) + \cos(\psi_c) \cdot \left( \int k_{iy}^p \cdot d_y dt + y_y^{cross} \right) \right) \quad (6.48)$$

$$\theta_d^{PI} = \widetilde{\theta}_d^P + \left( \cos(\psi_c) \cdot \left( \int k_{ix}^p \cdot d_x dt + y_x^{cross} \right) + \sin(\psi_c) \cdot \left( \int k_{iy}^p \cdot d_y dt + y_y^{cross} \right) \right) \quad (6.49)$$

Zuletzt wird der Vektor  $\underline{W}_d^{PI}$  auf  $y_{max}^p$  unter Beibehaltung der Richtung begrenzt.

$$\widetilde{\phi}_d^{PI} = \phi_d^{PI} \cdot \left( y_{max}^p / \left| \underline{W}_d^{PI} \right| \right), \quad (6.50)$$

wenn  $\left| \underline{W}_d^{PI} \right| > y_{max}^p$ .  $\widetilde{\theta}_d^{PI}$  wird analog bestimmt.

### • Phase 5: I-Anteil reduzieren

In dieser Phase wird der vorher aufgebaute Integralvektor  $\underline{\int ddt}$  schrittweise wieder verkleinert durch

$$\underline{\int ddt} = \underline{\int ddt} \cdot 0.99, \quad (6.51)$$

falls  $\left| \underline{W}_d \right| < y_{max}^p$  und  $y_d = y_{max}^p$ .

Dies ist notwendig, da der I-Anteil des Reglers nur dann geändert wird, wenn  $y_d < y_{max}^p$  ist, sich der Flugroboter also in der Nähe des Ziels befindet. Jetzt kann es passieren, dass sich während eines Streckenflugs die Windverhältnisse so ändern, dass eine Reduzierung des I-Anteils notwendig ist, um das Ziel zu erreichen. Allerdings bedeutet dies, dass jedes mal, wenn der Flugroboter seine Zielposition erreicht, der I-Anteil von neuem wieder aufgebaut werden muss, was sich in einem anfänglichen Positionsfehler niederschlägt.

- **Phase 6: Dämpfungen**

Die Dämpfung ist das wichtigste Instrument, um den Flugroboter zu stabilisieren. Ohne sie würde der Flugroboter ständig um die Zielposition oszillieren. Das Ziel der Dämpfungen ist es also, die Dynamik und die Geschwindigkeiten des Systems zu reduzieren.

In diesem Abschnitt werden drei unterschiedliche Arten von Dämpfungen beschrieben. Die „Along-Track-Geschwindigkeits-Dämpfung“  $ATSD$ , die „Cross-Track-Geschwindigkeits-Dämpfung“  $CTSD$  und die „Beschleunigungsdämpfung“  $AD$ , wobei die  $ATSD$  dynamisch ist und die  $CTSD$  und  $AD$  konstant sind.

- *Geschwindigkeitsdämpfung*

Es wird zwischen den beiden Dämpfungen  $ATSD$  und  $CTSD$  unterschieden, da eine Dämpfung in Richtung des Ziels nicht wünschenswert ist, solange sich der Flugroboter auf dem Streckenflug befindet. Die Dämpfung orthogonal zur Trajektorie sollte ständig vorhanden sein, um ein Oszillieren um die Trajektorie zu vermeiden.

Aus diesem Grund ist der Regelparameter  $k_{ATSD}^p$  für  $ATSD$  dynamisch und zwar in Abhängigkeit von  $d$ . Es werden zwei Grenzen eingeführt und zwar  $d_{min}$  und  $d_{max}$ , wobei  $d_{max}$  die Entfernung ist, ab der keine Dämpfung mehr vorhanden ist und  $d_{min}$  die Entfernung, ab der die Dämpfung ihren maximalen Wert hat. Ab diesem Wert gilt also  $k_{ATSD}^p = k_{CTSD}^p$ , wobei  $k_{CTSD}^p$  der Regelparameter für  $CTSD$  ist.  $k_{ATSD}^p$  wird wie folgt berechnet.

$$k_{ATSD}^p = \begin{cases} 0, & d > d_{max} \\ CTSD, & d < d_{min} \\ m_c^p \cdot d + b_c^p, & sonst \end{cases} \quad (6.52)$$

mit

$$m_c^p = \frac{CTSD}{d_{min} - d_{max}} \quad (6.53)$$

mit  $d_{max} > d_{min}$  und

$$b_c^p = CTSD - d_{min} \cdot m_c^p \quad (6.54)$$

Das bedeutet, dass in der Nähe der Zielposition die volle Dämpfung in jede Richtung vorhanden ist.

Diese beiden Dämpfungen gehen nun in die folgenden Berechnungen zur Ermittlung der Lagewinkel  $\phi_d$  und  $\theta_d$  ein.

Es sei  $\underline{v}$  ein Vektor, der die aktuellen Geschwindigkeiten in x- und y-Richtung  $v_x$  und  $v_y$  enthält mit  $\underline{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$

Als erstes müssen die beiden Geschwindigkeitsvektoren bestimmt werden, die in Richtung der Zielposition und orthogonal zur Zielposition zeigen.

Als erstes wird der Winkel  $\omega_v^t$  zwischen dem Vektor  $\underline{v}$  und dem Zielvektor  $\omega_f$  bestimmt.

$$\omega_v^t = \omega_f - atan2(v_y, v_x) \quad (6.55)$$

Aus diesem Winkel werden dann die beiden Dämpfungen  $ATSC$  und  $CTSC$  berechnet.

$$ATSC = k_{ATSD}^p \cdot \cos(\omega_v^t) \cdot |\underline{v}| \quad (6.56)$$

$$CTSC = k_{CTSD}^p \cdot \sin(\omega_v^t) \cdot |\underline{v}| \quad (6.57)$$

Die neuen Sollwinkel  $\phi_d$  und  $\theta_d$  berechnen sich wie folgt:

$$\phi_d^{PId} = \tilde{\phi}_d^{PI} - (\sin(\omega_\psi) \cdot ATSD + \cos(\omega_\psi) \cdot CTSD) \quad (6.58)$$

$$\theta_d^{PId} = \tilde{\theta}_d^{PI} - (\cos(\omega_\psi) \cdot ATSD - \sin(\omega_\psi) \cdot CTSD) \quad (6.59)$$

### – Beschleunigungsdämpfung

Die Beschleunigungsdämpfung  $AD$  ist dafür zuständig, schnell auf Störungen zu reagieren und Beschleunigungen abzubremsen. Dabei soll diese Dämpfung nur eingreifen, wenn der Flugroboter beschleunigt, nicht wenn er schon langsammer wird.

Es sei  $\dot{\underline{v}}$  der Beschleunigungsvektor mit  $\dot{\underline{v}} = \begin{pmatrix} \dot{v}_x \\ \dot{v}_y \end{pmatrix}$  und  $\underline{\dot{v}}^{AD}$  der Dämpfungsvektor mit  $\underline{\dot{v}}^{AD} = \begin{pmatrix} \dot{v}_x^{AD} \\ \dot{v}_y^{AD} \end{pmatrix}$

Dann ist

$$\dot{v}_x^{AD} = \begin{cases} \dot{v}_x \cdot k_a^p, & (v_x > 0 \wedge \dot{v}_x > 0) \vee (v_x < 0 \wedge \dot{v}_x < 0) \\ 0, & \text{sonst} \end{cases} \quad (6.60)$$

wobei  $k_a^p$  der Regelparameter ist.  $\dot{v}_y^{AD}$  wird analog berechnet.

Die neuen Sollwinkel  $\phi_d$  und  $\theta_d$  berechnen sich wie folgt:

$$\phi_d^{PIdd} = \phi_d^{PID} - (\sin(\omega_\psi) \cdot \dot{v}_x^{AD} + \cos(\omega_\psi) \cdot \dot{v}_y^{AD}) \quad (6.61)$$

$$\theta_d^{PIdd} = \theta_d^{PID} - (\cos(\omega_\psi) \cdot \dot{v}_x^{AD} - \sin(\omega_\psi) \cdot \dot{v}_y^{AD}) \quad (6.62)$$

Als letztes wird noch einmal der Vektor  $\underline{W}_d^{PIdd}$  auf  $y_{max}^p$  unter Beibehaltung der Richtung begrenzt.

$$\phi_d = \phi_d^{PIdd} \cdot (y_{max}^p / |\underline{W}_d^{PIdd}|), \quad (6.63)$$

wenn  $|\underline{W}_d^{PIdd}| > y_{max}^p$ .  $\theta_d$  wird analog bestimmt.

### 6.3.3 Diskussion

Durch den vorgestellten 2D-Wegpunktregler ist es möglich, ohne Geschwindigkeitsbegrenzung zwischen zwei Wegpunkten zu fliegen. Während des Fluges wird sowohl die Sollstandlinie als auch die Höhe eingehalten. Dieser Regler wird eingesetzt, um die im Anwendungsfall (siehe Abschnitt 1.1) genannte Anforderung, „schnell“ ins Einsatzgebiet zu fliegen, zu erfüllen. Es gibt aber auch Nachteile bei dieser Regelung.

- Es gibt keine Kontrolle über die Dynamik des Systems während des Streckenfluges.
- Bei starken Beschleunigungen (vor allem beim Abbremsen) können Abweichungen im Gierwinkel (z.B. durch Kompassabweichungen oder durch Filterungen in der IMU) zum Ausbrechen des Systems führen.
- Während des Fluges gibt es keine Kontrolle über die tatsächliche Abweichung von der Trajektorie.
- Das Verfolgen von Objekten und damit das Landen auf beweglichen Oberflächen ist mit dieser Regelung nur schlecht möglich.

Aus diesen Gründen wurde ein weiterer Regler (siehe Trajektorienregler 6.4) entwickelt.

## 6.4 Trajektorienregler

In diesem Abschnitt wird der 3D-Trajektorienregler beschrieben. Dieser Regler ist eine Weiterentwicklung des Wegpunktreglers (6.3), um die in Abschnitt 6.3.3 erwähnten Nachteile zu vermeiden. Der Trajektorienregler wird mit 100Hz berechnet, der Lage- und Höhenregler mit 500Hz.

Grundvoraussetzung für diesen Regler ist die Fähigkeit, einen vorgegebenen Zustand  $S_d$  des UASs einzuhalten, der wie folgt definiert ist.

$$S_d = \{x_d, y_d, z_d, \dot{x}_d, \dot{y}_d, \dot{z}_d, \ddot{x}_d, \ddot{y}_d, \ddot{z}_d\} \quad (6.64)$$

Aus diesem Grund werden die im folgenden beschriebenen Positions-, Geschwindigkeits- und Beschleunigungsregler verwendet. Im Anschluss wird an zwei Beispielen beschrieben, wie einzelne Trajektorien berechnet werden. Zum Abschluss wird eine Untersuchung bezüglich der optimalen Übergänge zwischen den einzelnen Zuständen durchgeführt.

### 6.4.1 Positionsregelung

Ähnlich wie in 6.3.2 Phase 2 werden aus dem aktuellen Zustand  $S_c$  des UASs und  $S_d$  sowohl die Ausrichtung  $\omega$  als auch der Abstand  $d$  zwischen den beiden Positionen berechnet. Diese beiden Werte werden direkt aus GPS-Signalen bestimmt. Da die beiden Punkte niemals „weit“<sup>2</sup> auseinander liegen können (siehe 6.4.4), wird für die Berechnung der Algorithmus für „kurze Distanzen“ (siehe 6.2.1) verwendet.

Der Abstand zwischen zwei Punkten kann in seine Komponenten aufgeteilt werden:

$$d_x = \cos(\omega) \cdot d, d_y = \sin(\omega) \cdot d \quad (6.65)$$

Auf Grund der Tatsache, dass der aktuelle Gierwinkel  $\psi_c$  mit berücksichtigt werden muss, wird  $\omega$  korrigiert durch:  $\omega_\psi = \psi_c - \omega$ .

Als erstes werden sowohl die Sollwinkel  $\theta_d^{PI}$  und  $\phi_d^{PI}$  für die Position als auch  $thr^{PI}$  für die Höhe berechnet. Dazu wird ein veränderter PI-Regler (vgl. Abschnitt 6.3.2 Phase 2 und 4) mit den Parametern  $k_p^{xy}$  und  $k_p^z$  sowie  $k_i^x$ ,  $k_i^y$  und  $k_i^z$  verwendet.

$$\phi_d^{PI} = \sin(\omega_\psi) \cdot k_p^{xy} \cdot d + \sin(\psi_c) \cdot I_x - \cos(\psi_c) \cdot I_y \quad (6.66)$$

$$\theta_d^{PI} = \cos(\omega_\psi) \cdot k_p^{xy} \cdot d + \sin(\psi_c) \cdot I_x + \cos(\psi_c) \cdot I_y \quad (6.67)$$

---

<sup>2</sup>maximal:  $\frac{1}{100Hz} \cdot v_{cr}$  (1/Updaterate x Reisegeschwindigkeit)

$$thr^{PI} = k_p^z \cdot e_z + I_z \quad (6.68)$$

mit  $e_z = z_d - z_c$ .

Der integrale Anteil des Reglers wird verwendet, um konstante Abweichungen von der gewünschten Sollposition auszugleichen. Im Gegensatz zu einem klassischen I-Regler, werden zwei unterschiedliche Parameter  $k_{ih}$  und  $k_{il}$  abhängig von der aktuellen Bewegungsrichtung des UASs verwendet (vgl. 6.3.2 Phase 4). Der integrale Anteil  $I_x$  berechnet sich daher wie folgt:

$$I_x = \int g(t) dt, g(t) = \begin{cases} k_{ih} \cdot d_x, d_x \cdot \dot{x}_c < 0 \\ k_{il} \cdot d_x, \text{sonst} \end{cases} \quad (6.69)$$

$I_y$  und  $I_z$  werden analog berechnet. Der Parameter  $k_{ih}$  wird größer gewählt als  $k_{il}$ , so dass das Integral schnell auf Positionsänderungen in Folge von Winden reagieren kann. Der kleinere Parameter  $k_{il}$  sorgt dann dafür, dass der aufgetretene Fehler langsam mit limitiertem Überschwingen ausgeglichen wird. Des Weiteren werden alle Integrale ( $I_x$ ,  $I_y$  und  $I_z$ ) limitiert (vgl. 6.3.2 Phase 4), um Verzögerungen der Stellgröße zu vermeiden („integral windup“).

## 6.4.2 Geschwindigkeits- und Beschleunigungsregelung

Die Geschwindigkeits- und Beschleunigungsdämpfung sind die wichtigsten Terme, um den Quadrokopter zu stabilisieren, da es sich um ein unteraktuiertes System mit fast keiner internen Dämpfung handelt. Deshalb werden zwei unterschiedliche Dämpfungen zum Regler hinzugefügt (vgl. 6.3.2 Phase 6): Die Geschwindigkeitssdämpfung  $VD$  und die Beschleunigungsdämpfung  $AD$  (Gl. (6.70)-(6.72))

Aus diesem Grund werden  $\phi_d^{PI}$ ,  $\theta_d^{PI}$  und  $thr^{PI}$  verändert durch:

$$\phi_d^{PIdd} = \phi_d^{PI} + (\sin(\psi_c) \cdot VD_x - \cos(\psi_c) \cdot VD_y) + (\sin(\psi_c) \cdot AD_x - \cos(\psi_c) \cdot AD_y) \quad (6.70)$$

$$\theta_d^{PIdd} = \theta_d^{PI} + (\cos(\psi_c) \cdot VD_x + \sin(\psi_c) \cdot VD_y) + (\cos(\psi_c) \cdot AD_x + \sin(\psi_c) \cdot AD_y) \quad (6.71)$$

$$thr^{PIdd} = thr^{PI} + VD_z + AD_z \quad (6.72)$$

mit  $VD_x = k_v(\dot{x}_d - \dot{x}_c)$ .  $VD_y$  und  $VD_z$  werden analog berechnet.

Wie bei dem 2D-Wegpunktregler ist es sinnvoll, dass die Beschleunigungsdämpfung nur dann aktiv ist, wenn das System beschleunigt, nicht wenn es abbremst, da sie sonst auch dem Abbremsen entgegenwirken würde. Deswegen gilt z.B. für  $AD_x$ :

$$AD_x = \begin{cases} k_a(\ddot{x}_d - \ddot{x}_c), (\dot{x}_c > \dot{x}_d \wedge \ddot{x}_c > \ddot{x}_d) \vee (\dot{x}_c < \dot{x}_d \wedge \ddot{x}_c < \ddot{x}_d) \\ 0, sonst \end{cases} \quad (6.73)$$

$AD_y$  und  $AD_z$  werden analog berechnet.

Sobald  $\phi_c$  und  $\theta_c$  ungleich Null sind, zeigt der Schubvektor nicht mehr exakt entgegen dem Erdschwerevektor  $g$ , so dass  $thr^{PIdd}$  erhöht werden muss, um den gleichen Schub zu erreichen. Deshalb wird  $thr^{PIdd}$  durch folgenden Term verändert:

$$thr = thr^{PIdd} \cdot \frac{1}{\cos \phi_c \cdot \cos \theta_c} \quad (6.74)$$

für  $\cos \phi_c \cdot \cos \theta_c \neq 0$

Auf Grund der physikalischen Grenzen muss der komplette Verkippungswinkel  $\alpha = \left| \begin{pmatrix} \phi_d \\ \theta_d \end{pmatrix} \right|$  zu einem maximalen Winkel  $\alpha_{max}$  begrenzt werden, unter Berücksichtigung der Richtung.

$$\tilde{\phi}_d = \phi_d^{PIdd} \cdot \min(1, \alpha_{max}/\alpha), \quad (6.75)$$

$$\tilde{\theta}_d = \theta_d^{PIdd} \cdot \min(1, \alpha_{max}/\alpha), \quad (6.76)$$

Des Weiteren muss der Schubvektor  $F$  begrenzt werden, um im Intervall  $[thr^{min}, thr^{max}]$  zu bleiben.

Im Falle eines unerwünschten Höhenabfalls des Quadroopters, z.B. durch Windturbulenzen, muss sichergestellt werden, dass die Höhenreglung bevorzugt wird. Daher werden die Sollwinkel wie folgt verändert:

$$\begin{pmatrix} \phi_d \\ \theta_d \end{pmatrix} = \begin{pmatrix} \tilde{\phi}_d \\ \tilde{\theta}_d \end{pmatrix} \cdot z_{loss} \quad (6.77)$$

wobei  $z_{loss} = \begin{cases} \frac{1}{e_z - (e_z^{err} - 1)), e_z > e_z^{err}} \\ 1, sonst \end{cases}$  eine Gleichung ist, um dem Höhenabfall (vgl. 6.3.2 Phase 3) entgegen zu wirken, mit  $e_z^{err} > 0$ .

### 6.4.3 Stabilitätsanalyse

#### 6.4.3.1 Methode nach Lyapunov

Der Begriff Stabilität kann wie folgt beschrieben werden: „Eine Lösung einer Differentialgleichung, beschrieben durch eine Trajektorie  $\underline{x}(t)$  ist stabil, wenn jede andere Lösung,

die in der Nähe von  $\underline{x}(t)$  beginnt, für alle Zeiten in der Nähe von  $\underline{x}(t)$  bleibt.“ [Unb07]

Um die Methode von Lyapunov anwenden zu können, wird eine zeit-invariante gewöhnliche Differentialgleichung benötigt. [FPEN06]:

$$\dot{\underline{x}}(t) = f(\underline{x}(t), \underline{u}(t), t), \underline{x} \in \mathbb{R}^n; \underline{u} \in \mathbb{R}^m, \underline{x}(t_0) = \underline{x}_0, \quad (6.78)$$

mit dem  $n$ -dimensionalen Zustandsvektor  $\underline{x}(t)$  und  $m$  Eingangssignalen  $\underline{u}(t)$ .

### Stabilitätsdefinition nach Lyapunov:

Es sei  $\underline{x}(x_0, t)$  eine Lösung des gewöhnlichen Differentialgleichungssystems (6.78) und  $E$  der Arbeitspunkt. Dann heißt die Lösung  $\underline{x}(x_0, t)$  Lyapunov stabil, wenn es zu jeder reellen Zahl  $\epsilon > 0$  eine andere reelle Zahl  $\delta = \delta(\epsilon) > 0$  gibt, so dass gilt:

$$\forall \underline{x}_0 : |\underline{x}_0 - E| < \delta \Rightarrow |\underline{x}(x_0, t) - E| < \epsilon, \forall t \geq 0. \quad (6.79)$$

Die Lösung  $\underline{x}(x_0, t)$  heißt lokal asymptotisch stabil, wenn zusätzlich der Grenzwert

$$\lim_{t \rightarrow +\infty} |\underline{x}(x_0, t) - E| = 0 \quad (6.80)$$

gegeben ist.

Im folgenden wird definiert:

$$\underline{\xi}(\xi_0, t) = \underline{x}(x_0, t) - E \quad (6.81)$$

Um die Stabilität eines Systems in Ruhelage nachzuweisen, wird die direkte Methode von Lyapunov verwendet: (siehe [Unb07]). Das Grundprinzip dieser Methode liegt in der Annahme, dass eine stabile Ruhelage dadurch gekennzeichnet sein muss, dass die zeitliche Änderung der Gesamtenergie des Systems in der Umgebung der Ruhelage nie positiv wird. Um die Stabilität der Ruhelage nachzuweisen, ist es nötig, eine Funktion zu finden, die das Energieniveau des Systems darstellt. Für diese skalare Funktion  $V(\underline{\xi})$  muss gezeigt werden, dass

$$V(\underline{\xi}) > 0, \forall \underline{\xi} \neq 0 \quad (6.82)$$

$$V(\underline{\xi}) = 0, \underline{\xi} = 0 \quad (6.83)$$

$$\dot{V}(\underline{\xi}) \leq 0. \quad (6.84)$$

Lyapunov hat gezeigt, dass diese Energiebetrachtung verallgemeinert und für nichtlineare dynamische Systeme angewendet werden kann. Dabei ist die physikalische Bedeutung der Energiefunktion  $V(\underline{\xi})$  untergeordnet. Ziel ist es, eine generalistische Energiefunktion zu

finden, die die genannten Bedingungen (6.82)-(6.84) erfüllt. Sobald eine Funktion  $V(\underline{\xi})$  gefunden worden ist, ist die Stabilität des Systems nachgewiesen.

### Definition:

- Eine skalare Funktion  $V(\underline{\xi})$  heißt positiv (negativ) *definit*, wenn ein  $h > 0$  existiert, so dass gilt:  $V(\underline{\xi}) > 0$  ( $V(\underline{\xi}) < 0$ ) für  $0 < |\underline{\xi}| \leq h$ .
- Eine skalare Funktion  $V(\underline{\xi})$  heißt positiv (negativ) *semidefinit*, wenn ein  $h > 0$  existiert, so dass gilt:  $V(\underline{\xi}) \geq 0$  ( $V(\underline{\xi}) \leq 0$ ) für  $0 < |\underline{\xi}| \leq h$ .
- Nicht *definite* und nicht *semidefinite* Funktionen heißen *indefinit*.

Lyapunov hat folgende drei Theoreme aufgestellt:

- **Erstes Theorem von Lyapunov**

Existiert eine positiv definite Lyapunovfunktion  $V(\underline{\xi})$ , deren Ableitung  $\dot{V}(\underline{\xi})$  negativ *semidefinit* ist, dann ist die Gleichgewichtslage  $\underline{\xi}_0$  Lyapunov stabil.

- **Zweites Theorem von Lyapunov**

Existiert eine positiv definite Lyapunovfunktion  $V(\underline{\xi})$ , deren Ableitung  $\dot{V}(\underline{\xi})$  negativ *definit* ist, dann ist die Gleichgewichtslage  $\underline{\xi}_0$  lokal asymptotisch stabil.

- **Drittes Theorem von Lyapunov**

Existiert eine positiv definite Lyapunovfunktion  $V(\underline{\xi})$ , deren Ableitung  $\dot{V}(\underline{\xi})$  positiv definit ist und  $V(\underline{\xi})$  positiv sein kann für kleine  $\underline{\xi}$ , dann ist die Gleichgewichtslage  $\underline{\xi}_0$  instabil.

In [Unb07] werden Methoden vorgestellt, mit denen es möglich ist, geeignete Lyapunov-Funktionen zu finden. Eine wichtige Klasse positiv definiter Lyapunov-Funktionen hat die quadratische Form

$$V(\underline{\xi}) = \underline{\xi}^T \mathbf{P} \underline{\xi} \quad (6.85)$$

wobei  $\mathbf{P}$  eine symmetrische Matrix sei. Diese Funktionsklasse wird häufig verwendet, da schon einfache Systeme häufig aus der Summe der Quadrate seiner Variablen bestehen [Kem07].

#### 6.4.3.2 Lyapunov-Analyse

Die Stabilitätsanalyse der Regelung wird exemplarisch an der Differentialgleichung der  $x$ -Richtung durchgeführt. Als Grundlage wird die Differentialgleichung für die  $x$ -Bewegung (Gl. (2.4) aus Abschnitt 2.3) benötigt:

$$m \cdot \ddot{x}_g = u_1 \cdot (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - c_{D1} \cdot \dot{x}_g^2 - (w_y \dot{z}_g - w_z \dot{y}_g) \quad (6.86)$$

Für die Analyse wird angenommen, dass Geschwindigkeiten klein sind und der Widerstandsterm linear anwächst. Des Weiteren wird für die Analyse der Coriolis-Term vernachlässigt, um die Komplexität zu verringern. Das führt zu der vereinfachten Differentigleichung:

$$m \cdot \ddot{x}_g = u_1 \cdot (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - c_{D1} \cdot \dot{x}_g \quad (6.87)$$

Mit  $\xi = \dot{x}_g$  ergibt sich:

$$m \cdot \dot{\xi} = u_1 \cdot (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - c_{D1} \cdot \xi \quad (6.88)$$

Der Energiegehalt soll der Energiefunktion

$$V = \frac{1}{2} \xi^T m \xi + \frac{1}{2} e^T k_p e + \frac{1}{2} \dot{e}^T k_a \dot{e} \quad (6.89)$$

entsprechen. Die Ableitung ist

$$\dot{V} = \xi^T m \dot{\xi} + \frac{1}{2} \xi^T \dot{m} \xi + e^T k_p \dot{e} + \dot{e}^T k_a \ddot{e}. \quad (6.90)$$

Die Gl. (6.88) wird in die Gl. (6.90) eingesetzt und man erhält:

$$\dot{V} = \xi^T [u_1 \cdot (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) - c_{D1} \cdot \xi] + \frac{1}{2} \xi^T \dot{m} \xi + e^T k_p \dot{e} + \dot{e}^T k_a \ddot{e} \quad (6.91)$$

Der betrachtete Positionsregler ist:

$$u^* = k_p e + k_e \dot{e} + k_a \ddot{e} \quad (6.92)$$

mit  $e = x_d - x_c$ ,  $\dot{e} = \frac{d}{dt}(x_d - x_c) = -\xi$  und  $\ddot{e} = 0$ . Bei der Analyse wird der Integralanteil der Regelung zunächst nicht berücksichtigt. Des Weiteren beeinflusst eine Transformation mit sin- und cos-Anteilen nicht das Energieniveau des Systems und kann deshalb bei der Analyse vernachlässigt werden.

Für den Eingangswert  $u_1$  ergibt sich:

$$u_1 = \frac{1}{(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)} (k_p e - k_v \xi + k_a \ddot{e}) \quad (6.93)$$

wobei  $1/(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)$  die Rücktransformation von (6.86) ist. Durch Einsetzen (6.93) in Gl. (6.91) ergibt sich:

$$\dot{V} = \xi^T (k_p e - k_v \xi + k_a \ddot{e} - c_{D1} \xi) - e^T k_p \xi - \xi^T k_a \ddot{e} \quad (6.94)$$

$$\Leftrightarrow \dot{V} = -\xi^T c_{D1} \xi - \xi^T k_v \xi \leq 0 \quad (6.95)$$

für  $c_{D1} > 0$  und  $k_v > 0$ .

An den quadratischen Formen der Terme  $\xi^T c_{D1} \xi$  und  $\xi^T k_v \xi$  und an den Bedingung  $c_{D1} > 0$  und  $k_v > 0$  ist zu erkennen, dass die Gl. (6.95) *semidefinit* ist. Damit ist das System stabil im Sinne von Lyapunov.

Um die Stabilität des kompletten Systems inklusive Integralanteil zu analysieren, müssen zusätzlich noch zwei weitere Überlegungen durchgeführt werden.

- **quadratischer Widerstandsterm:**

Für die Analyse wurde der Widerstandsterm  $-C_D \dot{x}_g^2$  aus Gl. (6.86) linearisiert. Im realen System hat die Dämpfung aber eine quadratische Form. Zum einen lässt sich argumentieren, dass eine quadratische Dämpfung eher einen noch positiveren Einfluss auf die Stabilität des Gesamtsystems hat. Zum zweiten lässt sich aus (6.95) erkennen, dass sich für Widerstandsterme mit ungeraden Potenzen immer ein stabiles System nachweisen lässt. Es ist nicht ersichtlich, warum Widerstandsterme mit geraden Potenzen das System instabil werden lassen sollten.

- **I-Anteil:**

Für stetig differenzierbare Funktionen kann das Integral zwischen zwei Punkten nicht unendlich groß werden. Da es sich bei der Fehlerfunktion  $e$  um eine solche Funktion handelt, ist das Integral nach oben hin beschränkt. Des Weiteren wird, wie in 6.4.1 beschrieben, das Integral durch den Regelalgorithmus begrenzt. In der LTI-Theorie sind Integrale im offenen Regelkreis grenzstabil, werden bei der Schließung des Regelkreises aber stabil. Aus diesen Gründen lässt sich schließen, dass der Integralanteil keinen negativen Einfluss auf die Stabilität des Systems hat.

Grundlage der stabilen Positionsregelung ist ein stabiler Lageregler, wie er in [Kem07] vorgestellt wird. Es wurde gezeigt, dass der Lageregler global asymptotisch stabil ist. Die in dieser Arbeit vorgenommene Modifikation mit einem I-Anteil hat aus den selben Gründen keinen negativen Einfluss auf die Stabilität der Lageregelung.

#### 6.4.4 Trajektorienberechnung

Für den beschriebenen Regler ist es notwendig, quasi-kontinuierlich (in den Experimenten mit 100Hz) den nächsten Sollzustand  $S_d$  zu berechnen. Das bedeutet, dass, wenn das UAS von Position A nach B fliegen soll, der Regler alle Sollzustände während des Fluges Schritt für Schritt berechnen muss.

Für diese Berechnungen benötigt der Algorithmus eine vorgegebene Reisegeschwindigkeit  $v_{cr}$  und eine Steig- bzw. Sinkrate  $\dot{z}_d$  zu jedem Zeitpunkt. Aus diesen Geschwindigkeiten wird jeder Wert des nächsten Sollzustandes  $S_d$  berechnet durch:

$$\begin{aligned}\dot{x}_d &= v_{cr} \cdot \cos(\omega_d) & x_d &= \int \dot{x}_d dt & \ddot{x}_d &= \frac{d\dot{x}_d}{dt} \\ \dot{y}_d &= v_{cr} \cdot \sin(\omega_d) & y_d &= \int \dot{y}_d dt & \ddot{y}_d &= \frac{d\dot{y}_d}{dt} \\ \dot{z}_d &= \dot{z}_d & z_d &= \int \dot{z}_d dt & \ddot{z}_d &= \frac{d\dot{z}_d}{dt}\end{aligned}$$

wobei  $\omega_d$  die Richtung der Trajektorie zum aktuellen Zeitpunkt ist.

Zum Beispiel ist bei einer linearen Trajektorie (Flug von Position A nach B, gleiche Höhe)  $\omega_d$  immer gleich der Peilung  $\omega_{start}$  zu Beginn der Trajektorie. Der Geschwindigkeitsvektor  $v_{cr}$  liegt dabei immer auf der Trajektorie.

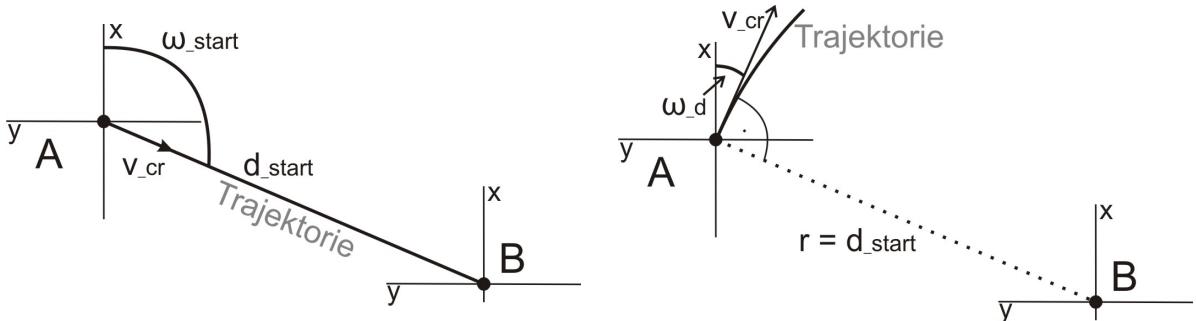


Abb. 6.5: Verschiedene Trajektorien - links: linear und rechts: Kreis

Für einen Kreis mit B als Mittelpunkt ist  $\omega_d = \omega_{circle} + (\frac{\pi}{2} \cdot dir)$ , wobei  $dir$  die Richtung des Kreises<sup>3</sup> angibt.

$$\omega_{circle} = \pi + \omega_{start} + \int \frac{|v_{cr}|}{r} dt \cdot dir \quad (6.96)$$

Hier liegt der Geschwindigkeitsvektor  $v_{cr}$  nicht exakt auf der Trajektorie, sondern bildet die Tangente vom Kreis an der aktuellen Position.

So lange sich das System in einem konstanten, also nicht beschleunigten Zustand befindet, ist der Regler in der Lage, den Sollzustand exakt einzunehmen. Kritische Momente treten

<sup>3</sup>dir = 1 im Uhrzeigersinn, dir = -1 gegen den Uhrzeigersinn

auf, wenn das System beschleunigt wird, also Änderungen in den Sollgeschwindigkeiten auftreten. Die Frage ist, unter welchen Geschwindigkeitsverläufen (im folgenden als „Rampen“ bezeichnet) die Trajektorie optimal ist. Optimal heißt in diesem Fall, dass während des gesamten Fluges der Positionsfehler zwischen Soll- und Istposition minimal ist.

Abb. 6.6 gibt einen Überblick über verschiedene Rampenarten, die bei einer Änderung der Reisegeschwindigkeit  $v_{cr}$  von hier  $0\text{m/s}$  auf  $5\text{m/s}$  verwendet werden können. Der Zeitraum, in dem die Änderung stattfindet, wird mit  $t_{alter}$  bezeichnet und ist hier auf  $6\text{s}$  gesetzt.

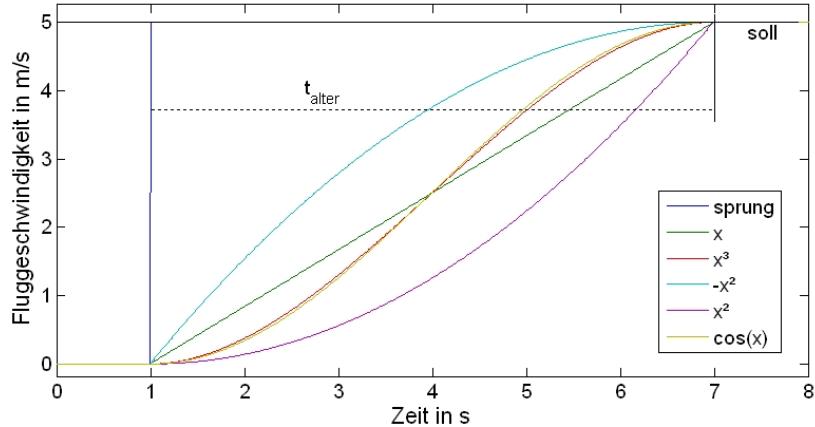


Abb. 6.6: Verschiedene Geschwindigkeitsverläufe / Rampen

Die Ergebnisse dieser Untersuchung werden in dem Simulationskapitel 7.3 präsentiert.

#### 6.4.5 Diskussion

Dieser Regelalgorithmus wird eingesetzt, um kontrolliert und exakt einer vordefinierten Trajektorie zu folgen. Mit dieser Methode lässt sich voraussagen, zu welchem Zeitpunkt  $t$  sich das System an einer bestimmten Position  $p$  befindet.

Für diese Regelung wird aber im Vergleich zum 2D-Wegpunktregler ein höherer Rechenaufwand benötigt. Der Sollzustand  $S_d$  muss zu jedem Zeitpunkt  $t$  berechnet werden. Des Weiteren muss ein Algorithmus vorhanden sein, der die Geschwindigkeitsprofile zwischen zwei Wegpunkten berechnet. Der Nachteil gegenüber dem 2D-Wegpunktregler liegt darin, dass dieser Regler es nicht ermöglicht, ohne Geschwindigkeitsbegrenzung zu fliegen. Aus diesem Grund ist es sinnvoll, beide Regler zu verwenden, um den Anwendungsfall (siehe Abschnitt 1.1) zu erfüllen.

Zusammenfassend kann festgehalten werden, dass der 2D-Wegpunktregler verwendet wird, um schnell ins Einsatzgebiet zu fliegen, und der Trajektorienregler, um exakt Trajektorien zu folgen.

## 6.5 Reglerentwurf

Für die vorgestellten Regler müssen die Parameter optimal eingestellt werden. Dabei wurden in erster Annäherung die von Ziegler und Nichols 1942 publizierten Einstellregeln für PID-Regler verwendet. Um diese in Tabelle 6.1 dargestellten Regeln anwenden zu können, ist es vorher notwendig, die Werte  $K_{R_{krit}}$  und  $T_{krit}$  zu ermitteln. Der Faktor  $K_{R_{krit}}$  ist derjenige Verstärkungsfaktor, bei dem das System anfängt, instabil zu werden. Man versucht den P-Anteil des Reglers so zu wählen, dass sich ein grenzstabiles Regelverhalten einstellt. Der Wert  $T_{krit}$  ist dann die Periodendauer der sich einstellenden Dauerschwingung.

Tab. 6.1: Ziegler und Nichols Einstellregeln [Kem02]

Regler	Reglerverstärkung $K_R$	Nachstellzeit $T_N$	Vorhaltezeit $T_V$
P	$0.5 \cdot K_{R_{krit}}$	-	-
PI	$0.45 \cdot K_{R_{krit}}$	$0.85 \cdot T_{krit}$	-
PID	$0.6 \cdot K_{R_{krit}}$	$0.5 \cdot T_{krit}$	$0.12 \cdot T_{krit}$

Nachdem die einzelnen Freiheitsgrade der Regler auf diese Weise bestimmt wurden, müssen sie im Anschluss durch experimentelle Versuche noch optimiert werden. Dies ist notwendig, da es sich bei den Modellen um eine vereinfachte Darstellung der Realität handelt und Abweichungen auftreten können. Größte Herausforderung hierbei sind die fehlerbehafteten Sensordaten bei MEMS-Sensoren. Vor allem bei dem zusätzlichen Parameter  $k_a$  für die Beschleunigungsdämpfung muss das Signal-Rausch-Verhältnis der realen Sensordaten mit berücksichtigt werden. Der Parameter darf nicht so groß gewählt werden, dass dadurch das System in Folge des Sensorrauschens anfängt, unkontrolliert zu oszillieren.

In der Tabelle 6.2 sind die Reglerparameter für die verwendeten Regler, die sowohl in den Simulationen als auch in den Experimenten verwendet werden, aufgelistet.

Die Reglerparameter sind so gewählt, dass die Regler direkt die Stellwerte für die Motorsteller ausgeben. Da für den Guard S und den Guard M unterschiedliche Motorsteller und Motoren verwendet werden, sind die Reglerwerte in Bezug auf den Höhen- und Lageregler unterschiedlich. Der Regler wird mit einer Frequenz von  $1\text{kHz}$  aufgerufen. Das bedeutet, dass auch die Integrale mit  $1\text{kHz}$  berechnet werden. Aus diesem Grund sind die Parameter  $k_i$  (mit  $k_i = \frac{T}{T_1}$  siehe Abschnitt 2.4.1) relativ klein.

Tab. 6.2: Durch Experimente optimierte Reglerparameter:

Parameter	Guard S	Guard M
Lageregler ( $\phi/\theta$ )		
$k_p$	2.5	28.5
$k_d$	25	285.7
$k_{ih}$	0.025	0.014
$k_{il}$	0.0125	0.007
Lageregler ( $\psi$ )		
$k_p$	5	100
$k_d$	6.25	128.5
Höhenregler		
$k_p$	13	530
$k_d$	40	830
$k_a$	4	100
$k_{ih}$	0.03	0.04
$k_{il}$	0.015	0.02
Positionsregler		
$k_p$	4	4
$k_d(CTSD)$	18	12
$k_a$	2	2
$k_{ih}$	0.0015	0.001
$k_{il}$	0.0005	0.0005

## 6.6 Objekt (Fahrzeug) verfolgen

Zum Verfolgen eines Objektes wird der Trajektorienregler verwendet, da dieser nicht nur die Möglichkeit bietet, die Position des Fahrzeuges einzunehmen, sondern auch die Geschwindigkeit und die Beschleunigung anpassen.

Sind die Geschwindigkeit und die Position des Fahrzeuges bekannt (z.B. über einen GPS-Empfänger) können diese Daten verwendet werden, um die einzelnen Schritte der Trajektorie zu berechnen (vgl. 6.4.4). Die GPS-Geschwindigkeit des Fahrzeuges entspricht dann der Sollgeschwindigkeit für den Regler zu jedem Zeitpunkt. Durch diese Geschwindigkeit werden dann alle anderen Sollwerte des Zustands  $S_d$  berechnet. Zusätzlich wird die berechnete Sollposition mit Hilfe eines Kalman-Filters mit der tatsächlichen GPS-Position des Fahrzeuges korrigiert.

## 6.7 Automatischer Start

Beim automatischen Start ist darauf zu achten, dass der Quadrokopter möglichst schnell seine Starthöhe  $z_{start}$  erreicht. Diese Anforderung ergibt sich aus der Anforderung, aus einer sich bewegenden Box zu starten. Ein langsames Abheben könnte zu einer Kollision mit der Box führen. Ein schlagartiges sprunghaftes Erhöhen der Motordrehzahlen  $n_c$  kann aber zu Instabilitäten führen, da nicht alle Motoren das gleiche Ansprechverhalten auf sich ändernde Stellwerte haben. Aus diesem Grund erfolgt der automatische Start nach folgendem Schema.

1. Die Motoren werden eingeschaltet und drehen sich mit einer konstanten minimalen Drehzahl  $n_c = n_{start}$
2. Danach werden die Drehzahlen  $n_c$  aller Motoren exponentiell erhöht und der Lageregelungsalgorithmus aktiviert.  $n_c = n_{start} + k * x^2$ , wobei  $x$  ein Millisekunden-Zähler und  $k$  ein Faktor ist, mit dem die Anstiegsgeschwindigkeit der Drehzahlen eingestellt wird.
3. Erst nachdem  $n_c$  die maximale Startdrehzahl  $n_{max}$  mit  $n_{max} > n_{schweben}$  erreicht hat, wird der Positions- und Höhenregler aktiviert. Da die maximale Startdrehzahl  $n_{max}$  größer ist als die Drehzahl  $n_{schweben}$ , die zum Schwebeflug benötigt wird, findet diese Aktivierung erst statt, nachdem der Quadrokopter schon in der Luft ist.
4. Die erste Sollhöhe der Höhenregelung nach dem Start  $z_{start}$  ist variabel.

## 6.8 Automatische Landung

In diesem Abschnitt wird der entwickelte Landealgorithmus beschrieben. Durch diesen Algorithmus ist der Quadrokopter in der Lage, aus jeglicher Position punktgenau in einem vorher definierten Bereich zu landen, und er schließt damit die Lücke zwischen der GPS-basierten Lokalisation bzw. Wegpunktnavigation und der Zentimeter genauen Nahfeldlokalisation. Als Positionsregler können sowohl der 2D-Wegpunktregler (siehe 6.3) als auch der Trajektorienregler (6.4) verwendet werden. Das Schema in Abb. 6.7 gibt einen Überblick über die unterschiedlichen Messverfahren, die während des gesamten Landevorgangs verwendet werden.

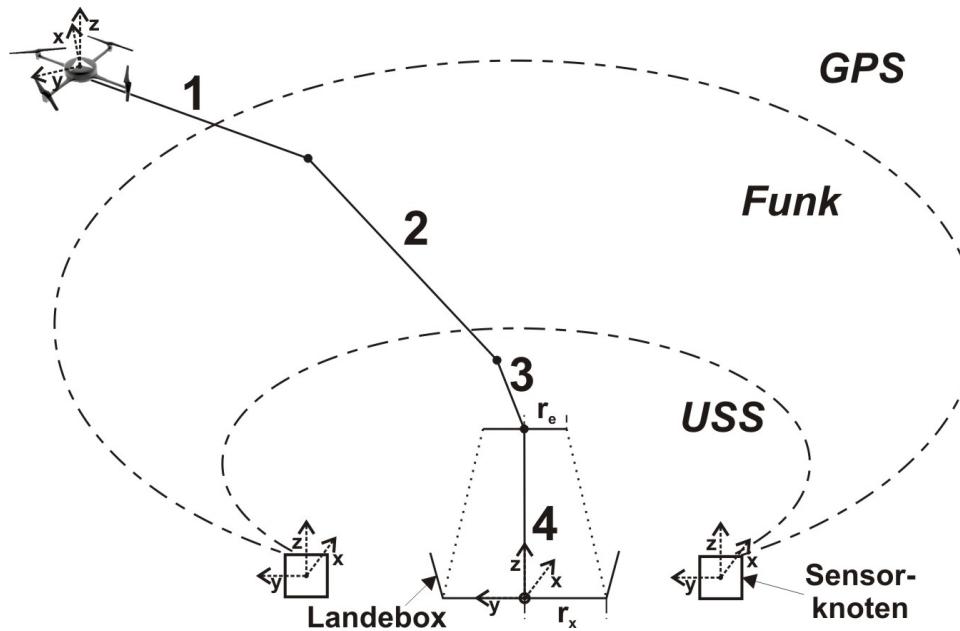


Abb. 6.7: Schematische Darstellung der Landung inkl. der verwendeten Messverfahren und der daraus resultierenden Phasen

Als Messverfahren kommen das GPS, das funkbasierte und das ultraschallbasierte Messsystem zum Einsatz, welche in dem Abschnitt 5.2 beschrieben werden.

Das Landeverfahren wird in vier Phasen eingeteilt, die sich aus den einzelnen Messverfahren ergeben. Während eines Landevorgangs werden immer alle vier Phasen durchlaufen. Treten während des Landevorgangs Störungen in Folge von starken Windeinflüssen oder Messfehlern auf, können einzelne Phasen auch häufiger als einmal durchlaufen werden. Abb. 6.8 zeigt das Zustandsdiagramm des entwickelten Landealgorithmus.

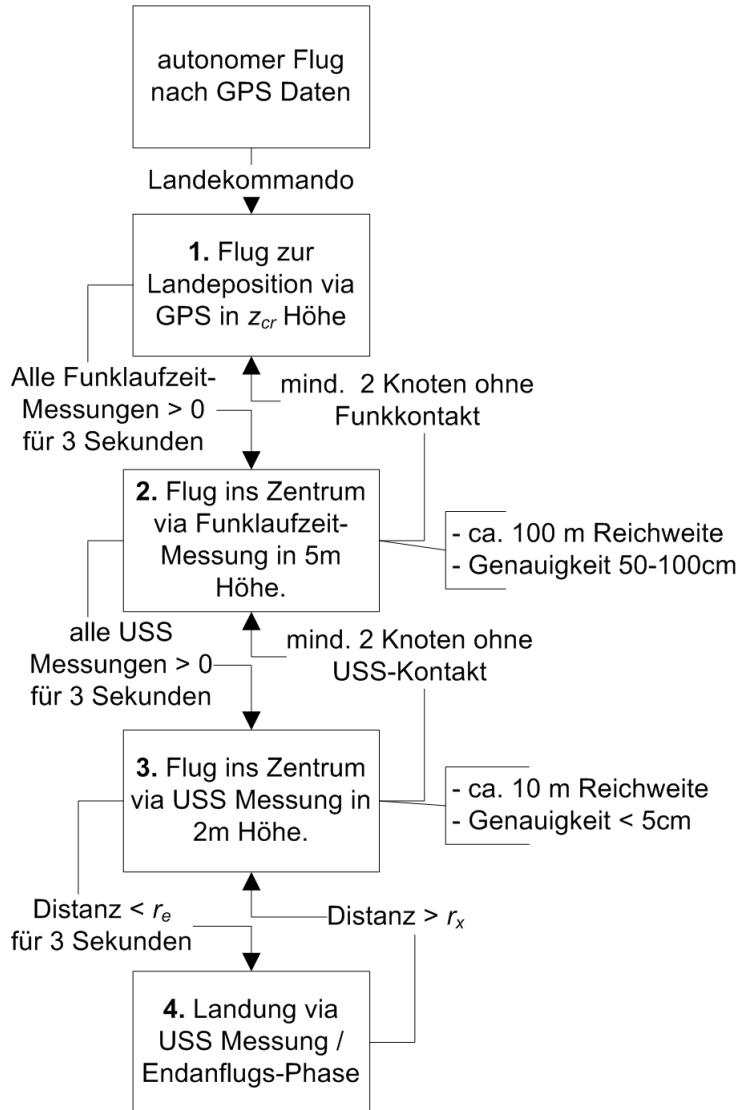


Abb. 6.8: Zustandsdiagramm des Landealgoritmus

## 1. Annäherung über GPS Daten

Sobald das Landekommando zum UAS gesendet wurde, fliegt das UAS mit einer vorgegebenen Flughöhe  $z_c$  mit Hilfe des GPS-Signals in die Nähe des Landepunktes. Die Genauigkeit des GPS (ca. 10m) ist dabei ausreichend, um zuverlässig in den Bereich des Funklaufzeitmesssystems (ca. 100m Reichweite, siehe Abschnitt 5.1) zu gelangen.

## 2. Annäherung über Funklaufzeitmessung

Sobald alle vier Sensorknoten für mindestens drei Sekunden Kontakt zum UAS haben (Messungen > 0), ändert sich der Landezustand. Drei Sekunden wurde gewählt, um stabile Messungen auch an der maximalen Reichweite der Funkmodule sicherzustellen, bevor umgeschaltet wird. Das UAS wird dann mit Hilfe der Funklaufzeitmessung ins Zentrum der vier Sensorknoten geleitet (Genauigkeit ca. 1m).

Die Flughöhe wird dabei weiter reduziert. Die Höhe muss dabei so gering sein, dass sichergestellt werden kann, dass das UAS in den Bereich der Ultraschallsensoren (ca. 10m Reichweite, siehe Abschnitt 5.1) gelangt.

Tritt der Fall auf, dass während dieser Phase die Funklaufzeitmessung ausfällt, also mehr als ein Knoten kein Kontakt mehr hat, wird die vorherige Phase wieder aktiv.

### 3. Positionierung über Ultraschall Messung

Befindet sich das UAS für mindestens drei Sekunden im Bereich der Ultraschallsensoren, übernimmt das Ultraschallmesssystem die Lokalisation. Diese Messung ist genau genug, um das UAS innerhalb eines Kreises von weniger als  $0.3m$  Radius zu landen. Für die Zeit wurde drei Sekunden gewählt, damit sichergestellt ist, dass sich der Quadrokopter auch in Ruhelage befindet, bevor er in den Endanflug übergeht.

Tritt der Fall auf, dass während dieser Phase die Ultraschallmessung ausfällt, also mehr als ein Knoten kein Kontakt mehr hat, wird die vorherige Phase wieder aktiv.

### 4. Endanflug

Zu diesem Zweck muss sich das UAS für mindestens  $t_{landing}$  Sekunden (hier  $3s$ ) innerhalb des „Enter-Endanflug-Radius“  $r_e$  (hier  $0.2m$ ) befinden, bevor der „Endanflug“-Zustand aktiviert wird. Verlässt das UAS während des Landevorgangs den so genannten „Exit-Endanflug-Radius“  $r_x$  (hier  $0.3m$ ) wird der vorherige Zustand wieder aktiv. Andernfalls werden die Motoren des UASs ausgeschaltet, sobald das UAS den Boden berührt.

Im Vergleich zu [VN09] und [Voo09b] leiten sich die einzelnen Phasen aus dem Einsatz der einzelnen Messmethoden ab. Es hat sich zudem herausgestellt, dass nach dem Eintritt in den „Endanflug“-Zustand der  $r_e$  größer sein muss als der  $r_x$ . Das begründet sich daran, dass sonst kleinste Positionsabweichungen während des „Endanflug“-Vorgangs häufig sofort zu einem Verlassen des Endanflugs führen.

## 6.9 Landung auf fahrendem Fahrzeug

Beim Landen auf einem fahrenden Fahrzeug wird im Prinzip derselbe Landealgorithmus verwendet, wie in Abschnitt 6.8 beschrieben wurde. Es gibt aber zusätzliche Voraussetzungen und Anforderungen, die erfüllt werden müssen, damit das Landen auf einer sich bewegenden Plattform funktionieren kann. Im folgenden werden die einzelnen Anforderungen beschrieben:

- **Geschwindigkeitsanpassung:**

Während des Landevorgangs muss sich das UAS an die Geschwindigkeit des Fahrzeugs anpassen. Das bedeutet, dass neben der Position auch die Geschwindigkeit des Fahrzeugs regelmäßig zum UAS übermittelt werden muss. Aus diesem Grund wurde während der Experimente (siehe 8.5) ein GPS-Sensor am Fahrzeug angebracht, der mit ca.  $5\text{ Hz}$  seine Positions- und Geschwindigkeitsdaten zum UAS übermittelt hat. Die Geschwindigkeitsanpassung bedeutet weiterhin, dass nur der Trajektorienregler (siehe 6.4) als Positionsregler verwendet werden kann, da nur dieser es ermöglicht, neben der Position auch die Geschwindigkeit des UASs zu regeln (vgl. 6.6).

- **Ausrichtung der Landeplattform muss bekannt sein:**

Neben der Position und der Geschwindigkeit muss auch die Ausrichtung des Fahrzeugs zu jedem Zeitpunkt bekannt sein. Das begründet sich dadurch, dass das verwendete Messverfahren, wie in Abschnitt 5.2 beschrieben, nicht die Möglichkeit bietet, die Ausrichtung zu ermitteln. Wenn sich das Fahrzeug bewegt, ist es möglich, aus dem GPS-Sensor die Ausrichtung zu bestimmen. Steht das Fahrzeug, muss ein zusätzlicher Kompass am Fahrzeug installiert werden.

- **Messfehler bei sich bewegender Plattform:**

Bei der Berechnung der Position mit Hilfe der Ultraschallsensoren muss ein Aspekt näher betrachtet werden. Die Ausbreitungsgeschwindigkeit  $c_{us}$  des Schalls beträgt ca.  $340 \frac{\text{m}}{\text{s}}$ . So lange sich die Messvorrichtung nicht bewegt ( $v = 0$ ), tritt kein Messfehler auf. Bei sich bewegender Messvorrichtung entsteht aber eine Messabweichung, die sowohl von der Geschwindigkeit der Landeplattform  $v$  als auch von der Entfernung des UASs zum Sensorknoten  $d$  abhängt. In Gl. (6.97) ist der Zusammenhang dargestellt.

$$e_d = d \cdot \left( \frac{c_{us}}{c_{us} - v} - 1 \right) \quad (6.97)$$

wobei  $e_d$  der entstandene Messfehler in Metern ist. Abb. 6.9 zeigt den Verlauf des Fehlers.

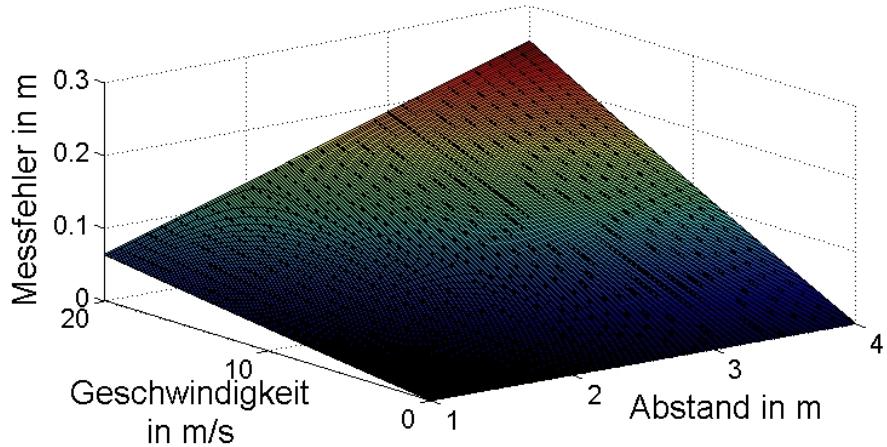


Abb. 6.9: Messfehler bei sich bewegender Messvorrichtung

Wenn man davon ausgeht, dass bei der Landung auf einem fahrenden Fahrzeug (siehe Kapitel 8.4) die Sensorknoten z.B. 2 Meter voneinander entfernt stehen, ist  $d$  während des Landevorgangs ca.  $1 - 2\text{m}$ . Das bedeutet, dass bei einer Fahrzeugschwindigkeit von  $10 \frac{\text{m}}{\text{s}}$  (ca.  $36 \frac{\text{km}}{\text{h}}$ ) ein Messfehler von maximal  $0.04 - 0.07\text{m}$  auftritt. Dieser Fehler liegt nur leicht über dem Bereich des Messrauschen der Ultraschallmessung und wurde deshalb während der Experimente vernachlässigt. Würde aber die Geschwindigkeit des Fahrzeuges weiter erhöht werden, müsste dieser Fehler bei der Lokalisation mit berücksichtigt werden.

# Kapitel 7

## Simulation

In diesem Kapitel werden verschiedene Simulationen bezüglich der in Kapitel 6 beschriebenen Regelalgorithmen vorgestellt. Alle Simulationen wurden in Matlab/Simulink mit den Differentialgleichungen (2.1) bis (2.6) aus Kapitel 2.3 als Grundlage durchgeführt.

### 7.1 Lageregelung inklusive I-Anteil

Wie schon im Kapitel 2 beschrieben, gibt es viele Arbeiten in Bezug auf die Lageregelung eines Quadroopters. Deshalb soll hier nur auf den I-Anteil der Regelung eingegangen werden.

In Abb. 7.1 ist der Verlauf des  $\theta$ - und  $\phi$ -Winkels bei Auftriebsverlust eines Motors von 25% zu erkennen. Die Auswirkung auf das System ist ähnlich, als wenn sich der Schwerpunkt in Richtung dieses Motors verschieben oder bei einem Flugsystem mit sechs oder acht Propellern ein Motor ausfallen würde.

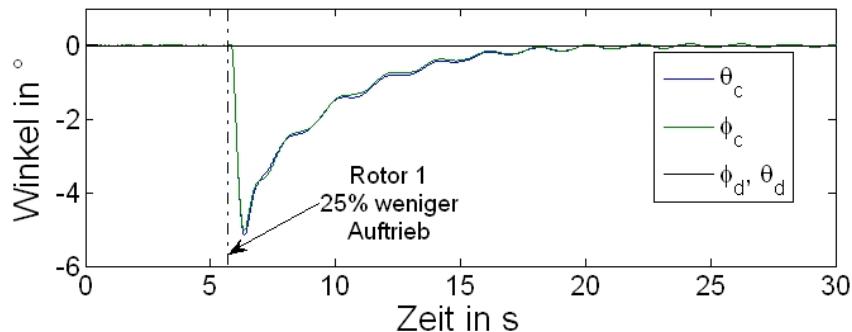


Abb. 7.1: Simulation: Lageregelung inklusive I-Anteil

Es ist zu erkennen, dass der Regler in der Lage ist, solche Störungen im System schnell abzufangen (nur ca.  $5^\circ$  Abweichung) und wieder auszugleichen, ohne dass das System

instabil wird. Dieses Verhalten ist Folge des nicht konstanten Parameters für den I-Regler (vgl.6.1, Phase 4) und der Dämpfung (vgl.6.1, Phase 3).

## 7.2 2D-Wegpunktregler

In diesem Abschnitt werden die Ergebnisse der Simulationen bezüglich des Wegpunktreglers aus Abschnitt 6.3 vorgestellt. Als Plattform für diese Simulationen wurde ausschließlich der Guard M als Grundlage (siehe 4.1.1) verwendet.

### 7.2.1 Höhenregelung

In Abb. 7.2 ist das Ergebnis des Höhenreglers zu erkennen. Es wurde sowohl ein Steigflug auf 100m als auch ein Sinkflug zurück auf 20m durchgeführt. Die Position wurde während der Simulation nicht verändert.

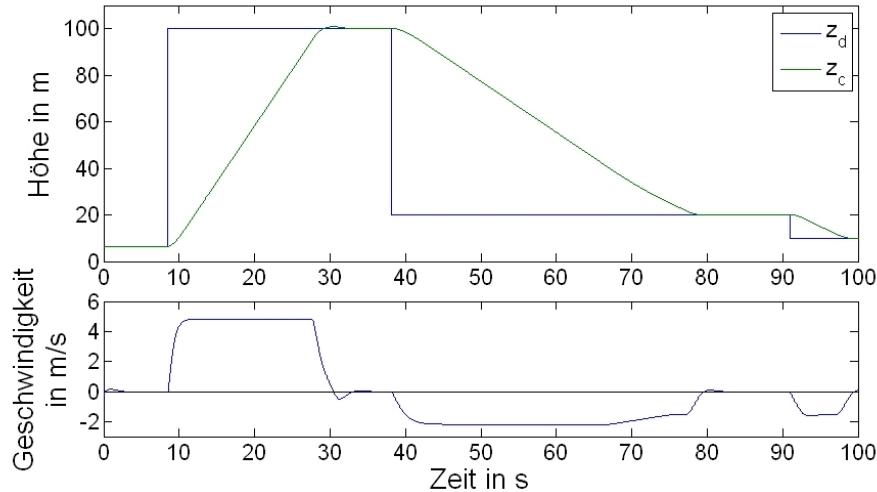


Abb. 7.2: Simulation: Höhenregler

Es ist zu sehen, dass auf Grund der limitierten Sinkrate der Abstieg länger dauert als der Aufstieg. Des Weiteren ist zu erkennen, dass die Sinkgeschwindigkeit am Ende des Abstiegs reduziert wird und deshalb kein Überschwingen zu erkennen ist.

### 7.2.2 Positionsaltung

In Abb. 7.3 ist das Verhalten des Wegpunktreglers bei Auftreten von statischem Seitenwind dargestellt. Der Seitenwind ist in der Art simuliert worden, dass das UAS mit einem bestimmten Wert (hier  $2\frac{m}{s^2}$ ) beschleunigt wird.

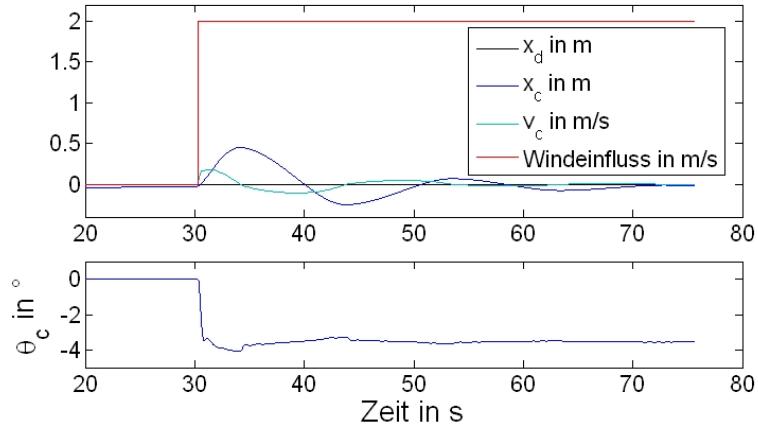


Abb. 7.3: Simulation: Positionshaltung mit Wind

Es ist zu erkennen, dass der Regler in der Lage ist, die Positionsabweichung schnell abzufangen (ca. 0.5m) und den Positionsfehler wieder auszugleichen, ohne dass das System instabil wird. In der unteren Grafik ist der korrespondierende  $\theta$ -Winkel dargestellt. Es ist zu sehen, dass sich der Quadrokopter um ca. 4° geneigt hat, um den statischen Seitenwind auszugleichen.

Wie in Abschnitt 6.3.2 (Phase 6) und in 6.4.2 beschrieben, wirkt die Beschleunigungsdämpfung nur dann, wenn das System auch beschleunigt wird, nicht wenn es abbremst. In Abb. 7.4 ist das Verhalten unterschiedlicher Beschleunigungsdämpfungen grafisch dargestellt. In der Simulation wurde das System während der Positionshaltung mit einer Störung von einer Sekunde belegt. Die Störung wurde in der Art simuliert, dass das System mit einer Kraft von 1g in eine Richtung beschleunigt wird.

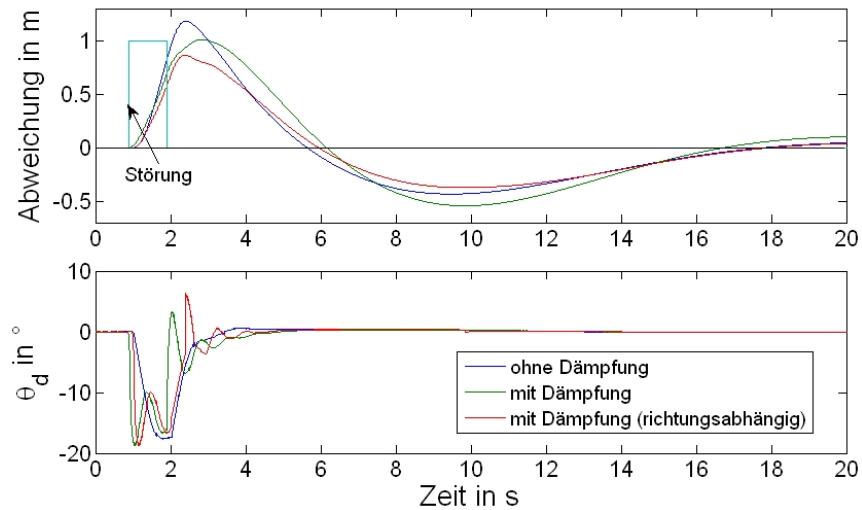


Abb. 7.4: Simulation: Positionshaltung mit Wind und unterschiedlichen Beschleunigungsdämpfungen

Tab. 7.1: Durchschnittliche und maximale Positionsabweichung über die Flugzeit nach Störeinfluss: 1) keine Beschleunigungsdämpfung vorhanden 2) mit Beschleunigungsdämpfung 3) nur Beschleunigungen werden gedämpft, das Abbremsen nicht

Abweichung	1) ohne Dämpfung	2) Dämpfung	3) richt.abh. Dämpfung
Durchschnitt	0.13m	0.15m	0.12m
Maximal	1.18m	1.01m	0.86m

Die Simulation hat gezeigt, dass die vorgestellte richtungsabhängige Beschleunigungsdämpfung die Qualität der Regelung erhöht. Sowohl die durchschnittliche als auch maximale Abweichung während des Fluges werden mit dieser Regelung minimiert. In Bezug auf die maximale Abweichung ist in dieser Simulation<sup>1</sup> die Güte um fast 30% besser als wenn keine Dämpfung verwendet wird. Im Vergleich zur „normalen“ Beschleunigungsdämpfung liegt die Verbesserung bei 15%. Auch die durchschnittliche Abweichung konnte geringfügig verbessert werden.

### 7.2.3 Streckenflug

Die Abb. 7.5 zeigt das Ergebnis eines 320m Streckenfluges inklusive Höhenänderung von ca. 15m.

Diese Simulation wurde einmal vor und einmal nach dem dazugehörigen Experiment aus Abschnitt 8.1.2 durchgeführt. Vor dem Experiment diente die Simulation dazu, herauszufinden, ob der Regler funktioniert. Bei der Simulation des Fluges nach dem Experiment wurden die am Boden parallel zum Flug gemessenen Windbedingungen bei der Simulation berücksichtigt, um einen besseren Vergleich zwischen Simulation und Experiment zu erhalten.

Die Abb. zeigt die Ergebnisse inklusive der gemessenen Windbedingungen. Das UAS wurde programmiert, sowohl die Position als auch die Höhe simultan zu ändern, um 320m Richtung Süden und wieder zurück zu fliegen mit einem konstanten Wind aus Norden von  $2 - 3 \frac{m}{s}$ . Wie erwartet führt der „Rückenwind“ beim Flug Richtung Süden zu einer höheren Reisegeschwindigkeit. Die Folge der höheren Geschwindigkeit und des Windes ist ein größeres Überschwingen beim Erreichen der Endposition im Süden. Des Weiteren ist der korrespondierende Höhenverlauf dargestellt. Es ist zu sehen, dass der integrale Anteil des Reglers funktioniert und sogar bei Sollwinkeländerungen die Höhe fast konstant bleibt.

---

<sup>1</sup>Parameter für die Dämpfung:  $k_a = 5$

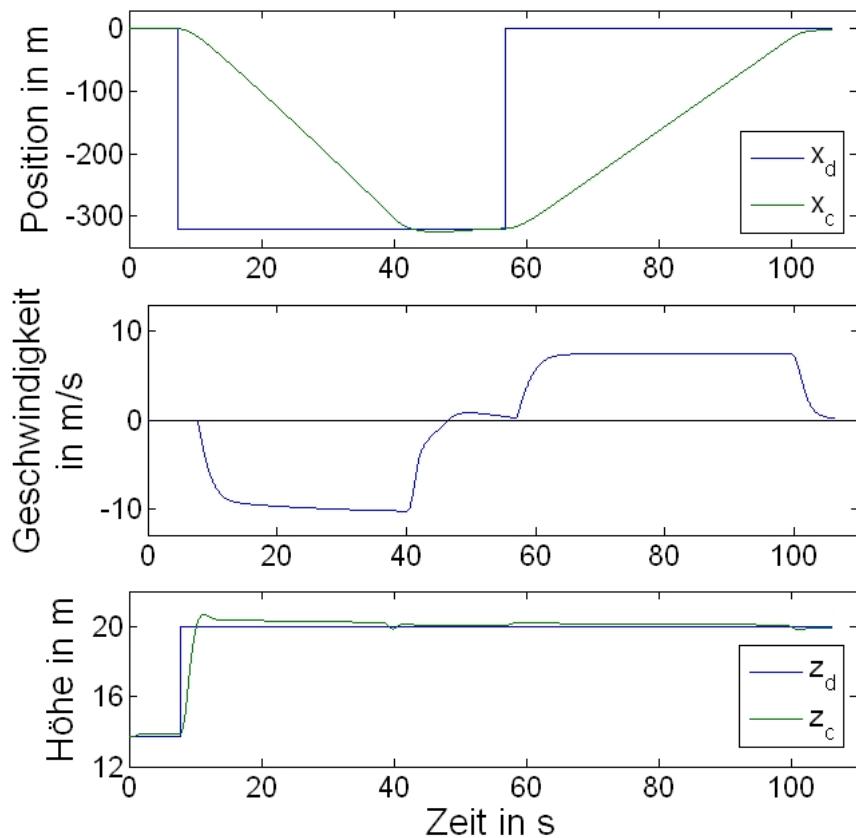


Abb. 7.5: Simulation: Streckenflug

#### 7.2.4 Sollstandlinienregelung

In Abb.<sup>2</sup> 7.6 ist das resultierende Verhalten der Sollstandlinienregelung (vgl. 6.3.2 Phase 2) zu erkennen. Der simulierte Flug wurde manuell gestartet am Wegpunkt (wp) 1. Das UAS wurde entlang der Trajektorie mit einer Geschwindigkeit von  $10 \frac{m}{s}$  zu wp 2 manövriert. Sobald das UAS diesen Wegpunkt erreicht hatte, wurde der Wegpunktregler aktiviert mit wp 1 als Zielposition und wp 2 als Startposition.

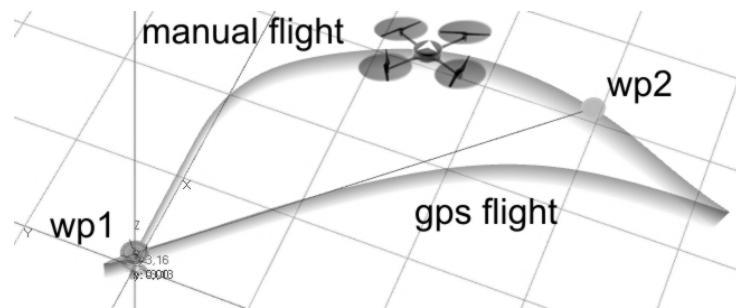


Abb. 7.6: Simulation: Sollstandlinienregelung

<sup>2</sup>Die Abmaße des Quadroopters innerhalb der Grafik sind nicht korrekt

Es ist zu erkennen, wie das UAS versucht, wieder zurück auf seine eigentliche Trajektorie zu gelangen. Je weiter das UAS von der Trajektorie entfernt ist, desto direkter fliegt es wieder auf sie zu.

### 7.2.5 Simultane Höhen- und Positionsänderung

Die folgenden beiden Abbildungen zeigen das Zusammenspiel der Positions- und Höhenregelung des Wegpunktreglers.

In Abb. 7.7 ist eine resultierende Trajektorie einer simultanen Positions- und Höhenänderung dargestellt. Das UAS hatte die Aufgabe, von der Startposition *sp* (Höhe 20m) bis zur Zielposition *tp* (Höhe 50m und 70m entfernt von *sp*) zu fliegen.

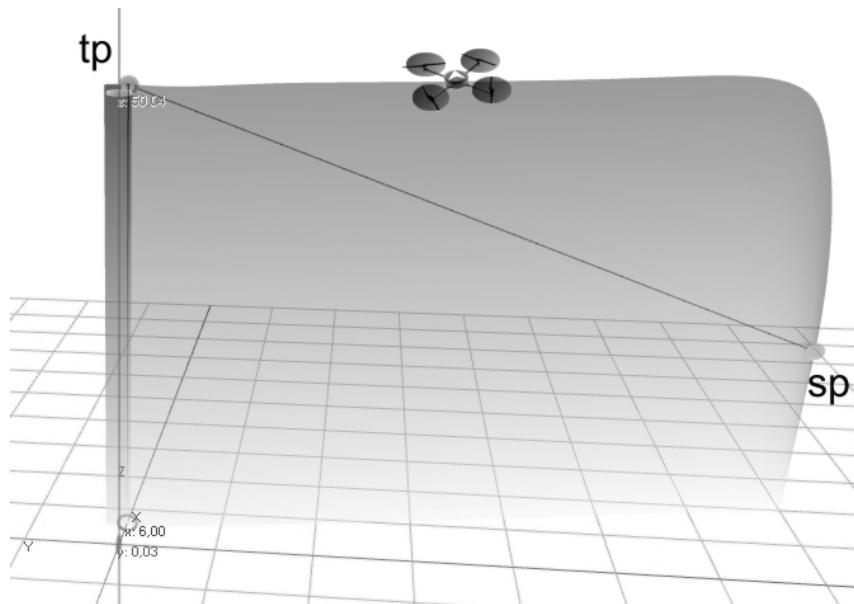


Abb. 7.7: Simulation: simultane Höhen- und Positionsänderung

Das Verhalten, dass das UAS nicht direkt (schwarze Linie) zum Ziel fliegt, ist durch die Phase 3 des Wegpunktreglers begründet (siehe 6.3.2). An dieser Simulation ist zu sehen, dass die bevorzugte Aufgabe des Reglers darin besteht, eine vorgegebene Höhe zu erreichen bzw. zu halten, bevor es Richtung *tp* fliegt.

Das gleiche Verhalten ist auch in Abb. 7.8 am wp 1 zu erkennen. Ziel dieser Simulation war es, entlang eines Rechtecks zu fliegen und gleichzeitig die Sollhöhe zu ändern. Während dieser Simulation wurde der Wind nicht berücksichtigt.

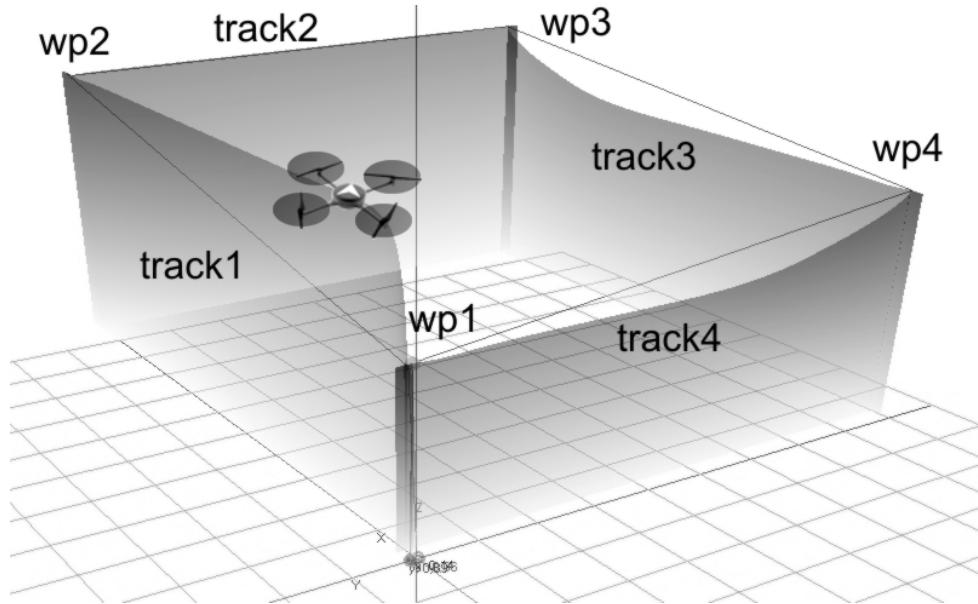


Abb. 7.8: Simulation: Abflug eines Vierecks

Der erste Wegpunkt (wp 1) befindet sich in 30m Höhe und ist die Startposition des Fluges. Nachdem die Startposition erreicht wurde, wurde eine simultane Höhen und Positionsänderung kommandiert. Wie auch in Abb. 7.7 zu sehen, beginnt das UAS zuerst, den Höhenunterschied von 20m auszugleichen, bevor es 100m nördlich Richtung wp 2 fliegt. Das begründet sich wiederum durch die Phase 3 des Reglers (speziell dem  $h_{loss}$ -Term). Der nächste Wegpunkt 3 befindet sich in der selben Höhe aber 90m weiter östlich. Richtung wp 4 wurde wieder eine simultane Positions- und Höhenänderung kommandiert, wobei die neue Sollhöhe in diesem Fall 10m unterhalb der alten Höhe liegt. Dieses Mal werden der Positions- und Höhenunterschied gleichzeitig reduziert, da der  $h_{loss}$ -Term nur einen Effekt hat, befindet sich das UAS unterhalb der Sollhöhe. Der letzte Teil des Fluges ist ähnlich dem vorherigen. Vom wp 4 fliegt das UAS wieder zur Startposition, welche sich 10m unterhalb von wp 4 befindet.

## 7.3 Trajektorienregler

In diesem Abschnitt werden Simulationen bezüglich des Trajektorienreglers vorgestellt. Hauptaugenmerk dabei liegt auf der Untersuchung der verschiedenen Rampenverläufe aus Abschnitt 6.4.4. Dabei werden sowohl die Auswirkungen verschiedener Rampenfunktionen als auch verschiedener Rampenzeiten  $t_{alter}$  betrachtet. Die Reglerparameter für den Regler waren bei allen Simulationen die selben, um die Resultate miteinander vergleichen zu können. Als Plattform wurde für diese und folgende Experimente ausschließlich der Guard S (siehe 4.1.2) verwendet.

### 7.3.1 Rampenfunktionen

In Abb. 7.9 ist die einfache 1D-Trajektorie eines 100m-Fluges zu sehen. Als Rampe wurde die  $\cos(x)$ -Funktion verwendet mit einer Änderungszeit  $t_{alter}$  von 6s und einer Reisegeschwindigkeit  $v_{cr}$  von  $5 \frac{m}{s}$ .

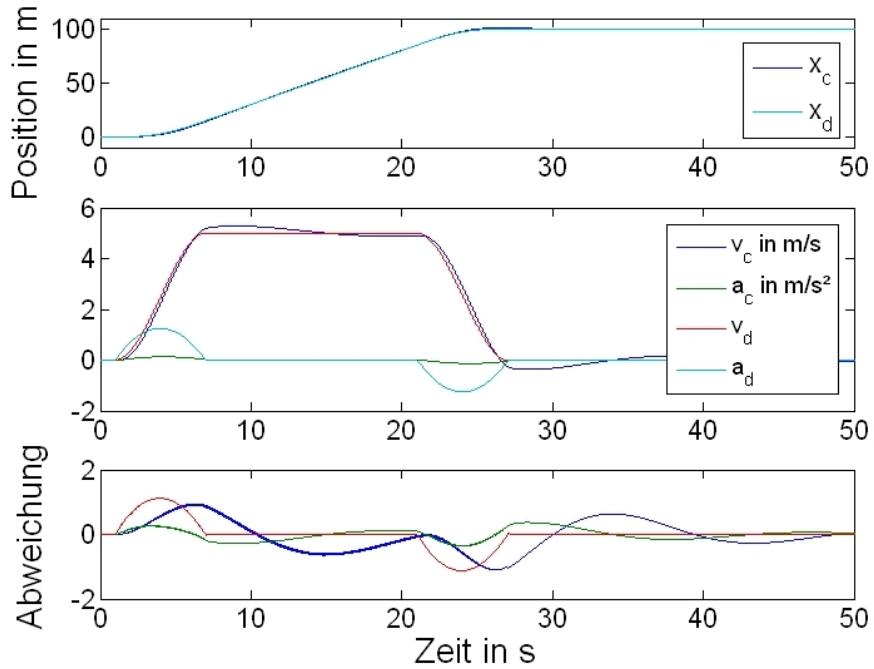


Abb. 7.9: Simulation: 100m-Flug mit 6s-cos(x)-Rampe,  $v_{cr} = 5 \frac{m}{s}$

Während dieser Simulation betrug die durchschnittliche Positionsabweichung  $0.33m$  und die durchschnittliche Geschwindigkeitsabweichung  $0.12 \frac{m}{s}$ .

Abb. 7.10 zeigt den Positionsfehler des selben 100m-Fluges aus Abb. 7.9, allerdings mit den unterschiedlichen Rampenfunktionen aus Abb. 6.6.

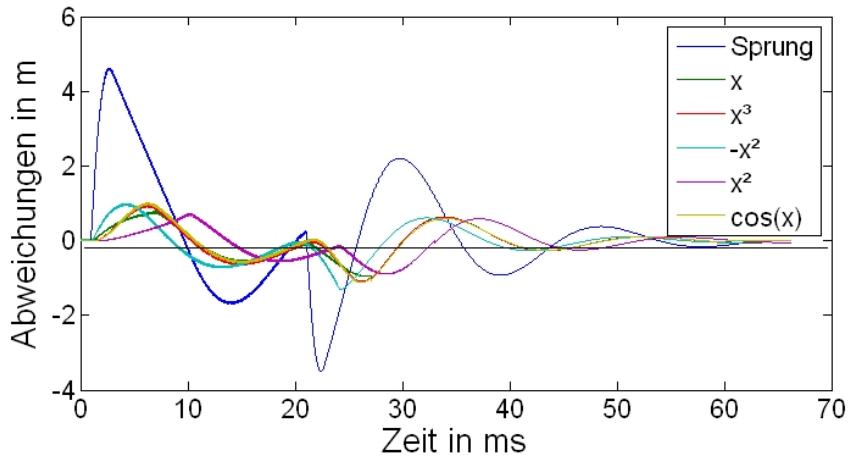


Abb. 7.10: Simulation: Positionsfehler während des 100m-Fluges aus Abb. 7.9 mit den verschiedenen Rampenfunktionen aus Abb. 6.6.

Tab. 7.2: Durchschnittliche Positionsabweichung während des 100m-Fluges mit unterschiedlichen Funktionen

<i>Sprung</i>	$x$	$x^3$	$-x^2$	$x^2$	$\cos(x)$
1.06m	0.32m	0.33m	0.33m	0.33m	0.32m

Es ist zu erkennen, dass die unterschiedlichen Rampenfunktionen, mit Ausnahme der Sprungfunktion, keine signifikanten Unterschiede bezüglich der Positionsabweichung verursachen und demzufolge alle diese Rampenfunktionen gleichwertig verwendet werden können. Der Sachverhalt, der in Abb. 7.11 dargestellt ist, relativiert diese Aussage aber. Zu sehen ist in dieser Abb. der korrespondierende Sollwinkel  $\theta_d$ , der sich während der Anstiegszeit  $t_{alter}$  aus dem Regler ergibt. Es ist zu erkennen, dass die nicht differenzierbaren Rampenverläufe von „Sprung“, „ $x$ “, „ $-x^2$ “ und „ $x^2$ “ zu unerwünschten Sprüngen im korrespondierenden Sollwinkel  $\theta_d$  führen. Aus diesem Grund sollte ein differenzierbarer Rampenverlauf wie z.B. bei der „ $\cos(x)$ “- oder der „ $x^3$ “-Rampe verwendet werden.

Im Gegensatz zu den Rampenfunktionen hat eine Änderung der Rampenzeiten  $t_{alter}$  eine nennenswerte Auswirkung auf die Positionsgenauigkeit während des Fluges. In Abb. 7.12 ist die durchschnittliche Positionsabweichung während des 100m-Fluges mit den verschiedenen Rampenzeiten zu sehen. Es ist eindeutig zu erkennen, dass längere Rampenzeiten zu geringen Positionsabweichungen führen. Daher sollten, falls möglich, längere Rampenzeiten gewählt werden.

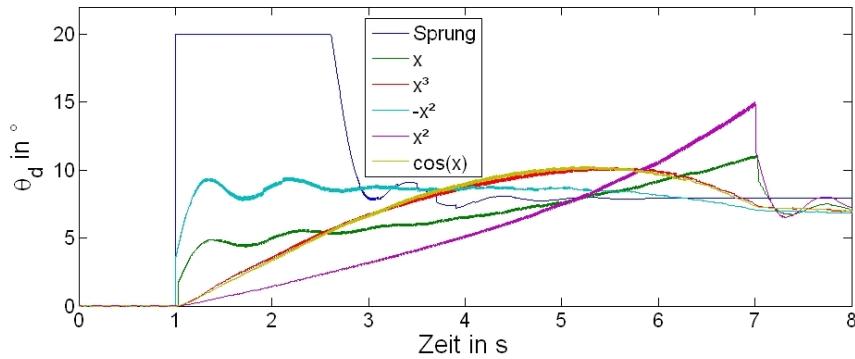


Abb. 7.11: Simulation: korrespondierende berechnete Winkel  $\theta_d$  von Abb. 6.6

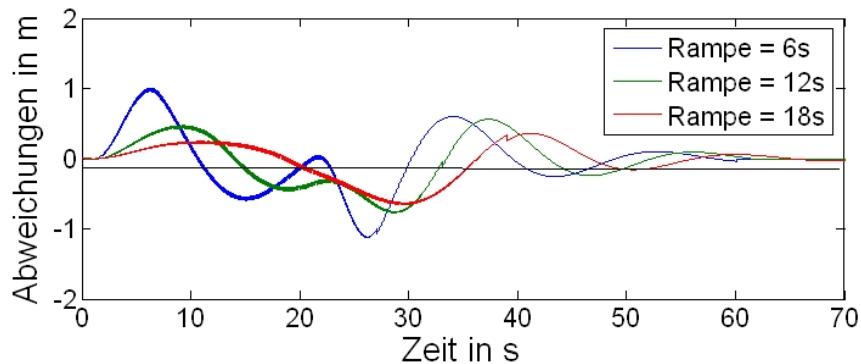


Abb. 7.12: Simulation: Positionsabweichung während des 100m-Fluges mit unterschiedlichen Rampenzeiten  $t_{alter}$

### 7.3.2 Geozentrischer Kreis

Wie im Anwendungsfall (siehe Abschnitt 1.1) beschrieben, soll das UAS in der Lage sein, so genannte geozentrische Kreise zu fliegen. Das bedeutet, das UAS fliegt auf einer kreisförmigen Trajektorie um ein definiertes Zentrum mit sich kontinuierlich änderndem  $\psi$ -Winkel, um das UAS ständig auf das Zentrum auszurichten.

Die Abbildungen 7.13 und 7.14 zeigen die Ergebnisse einer geozentrischen Kreistrajektorie mit einer  $\cos(x)$ -Rampe und einer Reisegeschwindigkeit von  $v_{cr} = 5 \frac{m}{s}$ . Der Radius des Kreises beträgt 25m. Zur Berechnung der Kreistrajektorie wurde die Formel (6.96) aus Abschnitt 6.4.4 verwendet.

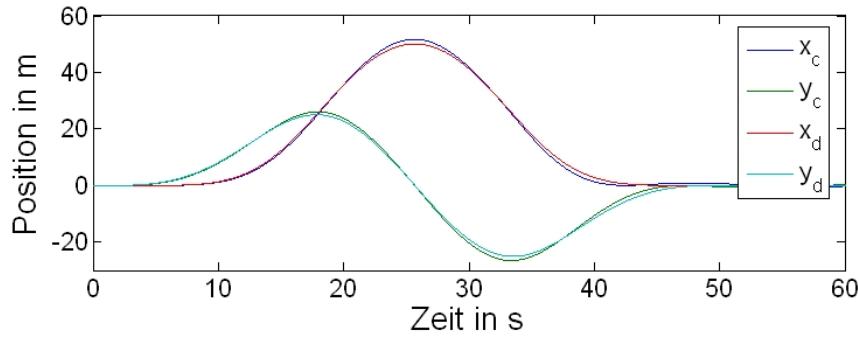


Abb. 7.13: Simulation: Geozentrischer Kreis mit einem Radius von  $25\text{m}$ ,  $18\text{s} \cdot \cos(x)$ -Rampe,  $v_{cr} = 5 \frac{\text{m}}{\text{s}}$

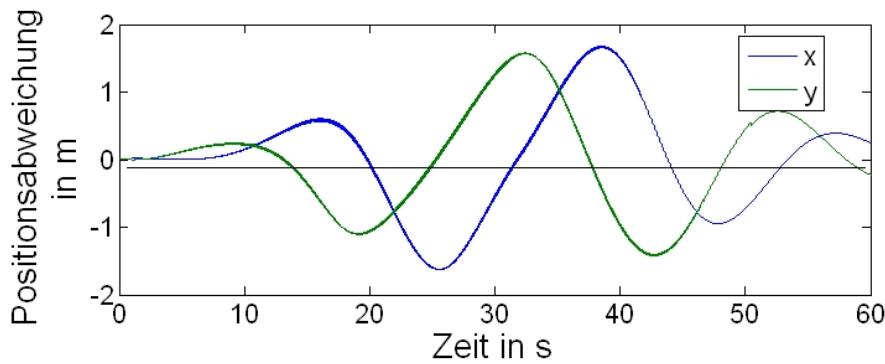


Abb. 7.14: Simulation: Positionsabweichung während des Kreisfluges

Es ist zu erkennen, dass der Positionsfehler größer ist als der Positionsfehler der linearen Trajektorie. Zum einen ergibt er sich durch den sich kontinuierlich ändernden  $\psi$ -Winkel und der Trägheit im System. Den größten Einfluss hat aber die Tatsache, dass sich die Flugrichtung ständig ändert und der tangentiale Geschwindigkeitsvektor nicht exakt auf die nächste Position auf dem Kreis zeigt. Ist die Trajektorie von vornherein bekannt, kann diese Abweichung herausgerechnet werden, um die Genauigkeit zu erhöhen. Zu diesem Zweck wird der Geschwindigkeitsvektor nicht als Tangente  $v_{cr}^T$  sondern als Sekante  $v_{cr}^S$  gewählt (siehe Abb. 7.15).

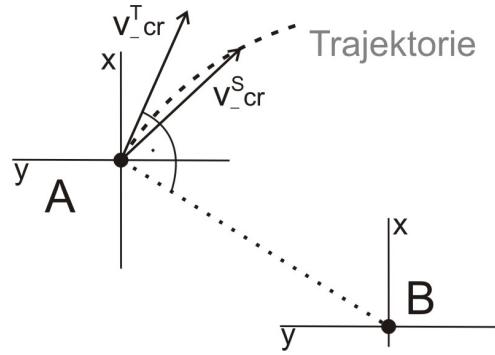


Abb. 7.15: Tangentialvektor  $v_{cr}^T$  und Sekantenvektor  $v_{cr}^T$

Die Berechnung für den Winkel  $\omega_d$  (vgl. Gl.(6.96) für den Tangentialvektor) verändert sich zu:

$$\omega_d = \omega_{circle} + \left[ \pi - \arccos \left( \frac{|v_{cr}|}{2 \cdot d_{start}} \right) \right] \cdot dir \quad (7.1)$$

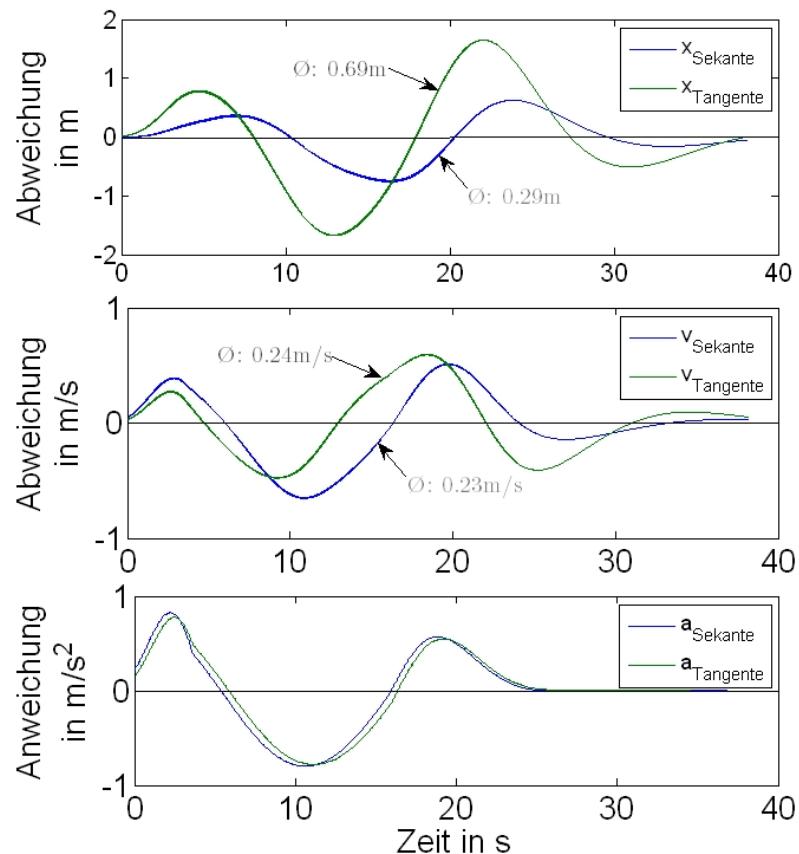


Abb. 7.16: Simulation: Abweichungen während des Kreisfluges mit Tangential- und Sekantenvektor

Das Ergebnis des veränderten Geschwindigkeitsvektors ist in Abb. 7.16 am Beispiel der

$x$ -Richtung zu sehen. Es wurde ein Kreisflug simuliert mit einem Radius von  $r = 10m$  und einer Geschwindigkeit von  $v_{cr} = 3\frac{m}{s}$ . Als Funktion wurde die  $6s\cos(x)$ -Rampe gewählt.

Der durchschnittliche Positionsfehler während des Kreisfluges sinkt mit dem veränderten Geschwindigkeitsvektor von  $0.69m$  auf  $0.29m$  (um ca. 60%) und ist damit fast identisch mit dem der linearen Trajektorie. Die Abweichungen in der Geschwindigkeit und der Beschleunigung dagegen bleiben fast konstant.

### 7.3.3 Höhenregelung

Neben der  $x$ - $y$ -Position gehört zum Trajektorienregler auch die  $z$ -Position. Abb. 7.17 zeigt das Ergebnis einer Höhenänderung. Als Rampenfunktion wurde die „ $x^2$ “-Funktion gewählt mit unterschiedlichen Rampenzeiten beim Beschleunigen und Abbremsen. Die Steig- bzw. Sinkrate betrug  $3\frac{m}{s}$ .

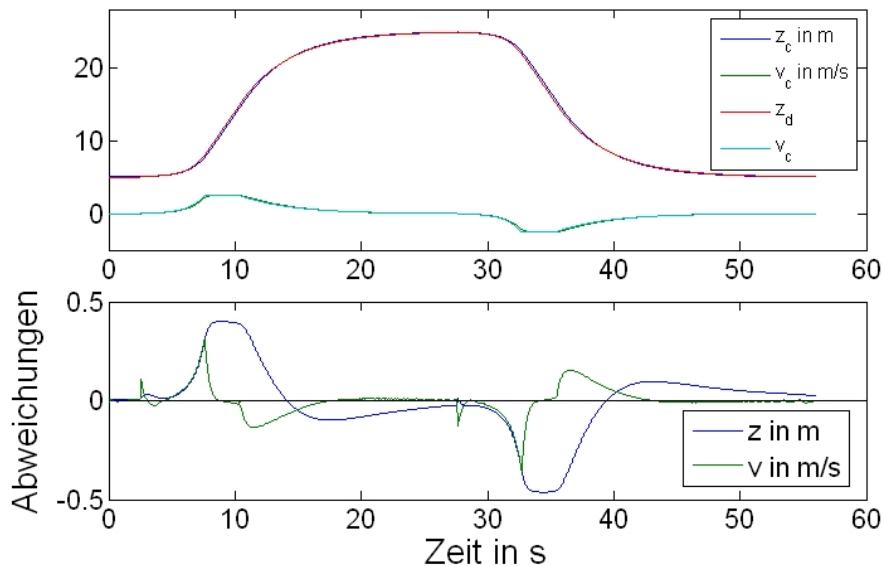


Abb. 7.17: Simulation: Höhenänderung mit  $x^2$ -Funktion und verschiedenen Rampenzeiten,  $v_{cr} = 3\frac{m}{s}$

Die durchschnittliche Höhenabweichung beträgt während der Simulation dabei  $0.17m$  und die Geschwindigkeitsabweichung liegt bei  $0.04\frac{m}{s}$ .

Diese Simulationen haben gezeigt, dass es mit diesem Regler möglich ist, jegliche Art von 3D-Trajektorie in  $x$ ,  $y$  und  $z$  zu fliegen. Zudem ermöglicht dieser Regler das Verfolgen von Objekten, wenn die Position und die Geschwindigkeit des Objektes bekannt sind. (siehe Abschnitt 7.4 und 8.3)

## 7.4 Objekt (Fahrzeug) verfolgen

Abb. 7.18 zeigt die Simulationsergebnisse einer Fahrzeugverfolgung.

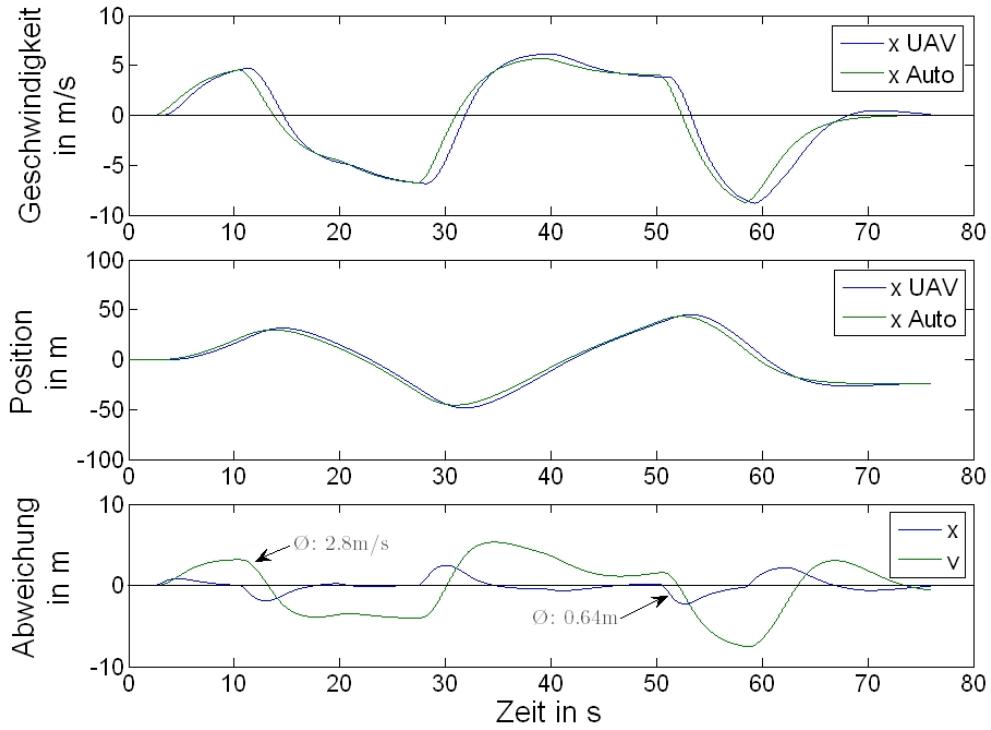


Abb. 7.18: Simulation: Fahrzeugverfolgung

Die durchschnittliche Positionsabweichung lag bei dieser Simulation bei  $0.64\text{m}$ , die der Geschwindigkeit bei  $2.8\text{m/s}$ . Abweichungen treten immer dann auf, wenn sich die Geschwindigkeit des Fahrzeuges ändert. Das liegt daran, dass die Regelung nur auf Änderungen reagieren kann, nachdem sie aufgetreten sind. Dieses Ergebnis entspricht den Untersuchungen in Abschnitt 7.3.

## 7.5 Automatischer Start

In Abb. 7.19 ist die Simulation eines automatischen Starts, wie er in Abschnitt 6.7 beschrieben wird, dargestellt. Die Startdrehzahl  $n_{start}$  beträgt  $1,000\text{rpm}$ , Enddrehzahl  $n_{max} = 4,800\text{rpm}$  und die initiale Schwebedrehzahl  $n_{schweben} = 4,050\text{rpm}$ .

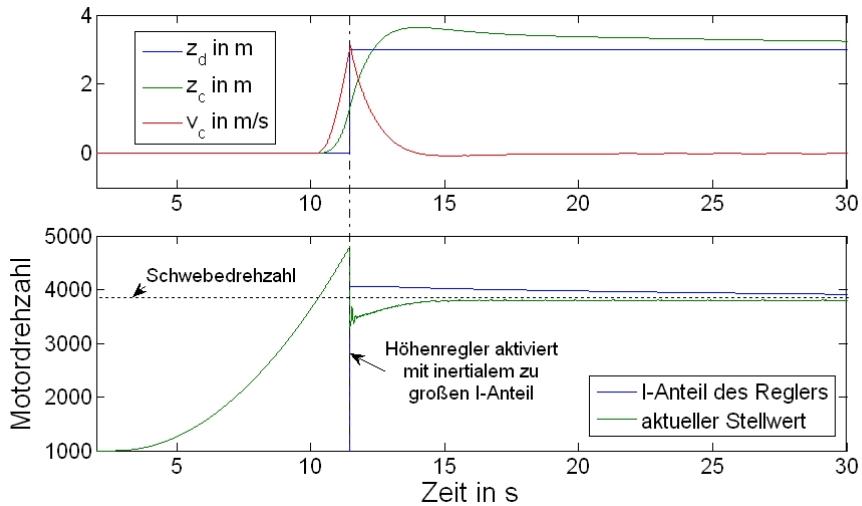


Abb. 7.19: Simulation: automatischer Start

Es ist zu erkennen, dass der Quadrokopter schon ca.  $1\text{m}$  in der Luft ist, als der Höhenregler aktiviert wird. Beim Aktivieren des Höhenreglers wird der I-Anteil des Höhenreglers auf die initiale Schwebedrehzahl (entspricht dem Grundschatz)  $n_{schweben} = 4050$  gesetzt (vgl. Abschnitt 6.3.1, Phase 2). Es ist zu sehen, dass die initiale Schwebedrehzahl höher ist, als für den Schwebeflug notwendig ist. Aus diesem Grund steigt der Quadrokopter nach der Aktivierung der Höhenregelung noch etwas weiter. Der I-Regler sorgt dann dafür, dass die Drehzahl soweit verringert wird, bis die Drehzahl dem Schub für den Schwebeflug entspricht.

## 7.6 Automatische Landung

Die Abb. 7.20 zeigt das Ergebnis einer Simulation des in Abschnitt 6.8 beschriebenen Landealgorithmus. Die Anflughöhe  $z_{cr}$  während der Simulation beträgt  $7.5\text{m}$ . Während aller vier Phasen (Annäherung über GPS, Annäherung über Funklaufzeitmessung, USS und Endanflug) wurde die aktuelle Position des UASs durch die Simulationsumgebung Matlab/Simulink berechnet, so dass es sich immer um ideale und fehlerfreie Daten handelt.

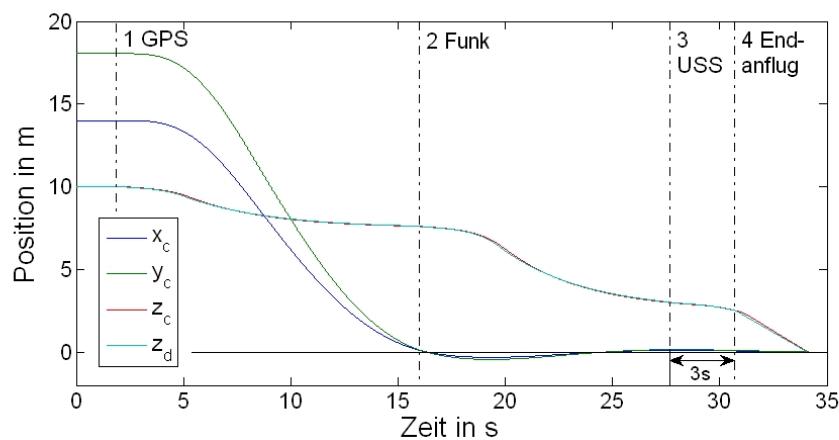


Abb. 7.20: Simulation: automatische Landung

In der Simulation konnte gezeigt werden, dass der entwickelte Landeargorithmus funktioniert und alle Phasen durchlaufen werden. Das UAS konnte mit einer Genauigkeit von  $0.03m$  gelandet werden.

Auf Grund der idealen Daten dauert in der Simulation die USS-Phase exakt drei Sekunden. Das liegt daran, dass sich das UAS schon vor Eintritt in diese Phase in dem „Enter-Endanflug-Radius“  $r_e$  befindet (vgl. 6.8). Bei den experimentellen Ergebnissen ist dieses anders (siehe Abschnitt 8.4).

## 7.7 Landung auf fahrendem Fahrzeug

In Abb. 7.21 ist die Simulation einer Landung auf einem beweglichen Ziel zu sehen. Es werden die gleichen Bedingungen verwendet wie bei der Simulation der automatischen Landung (vgl. 7.6).

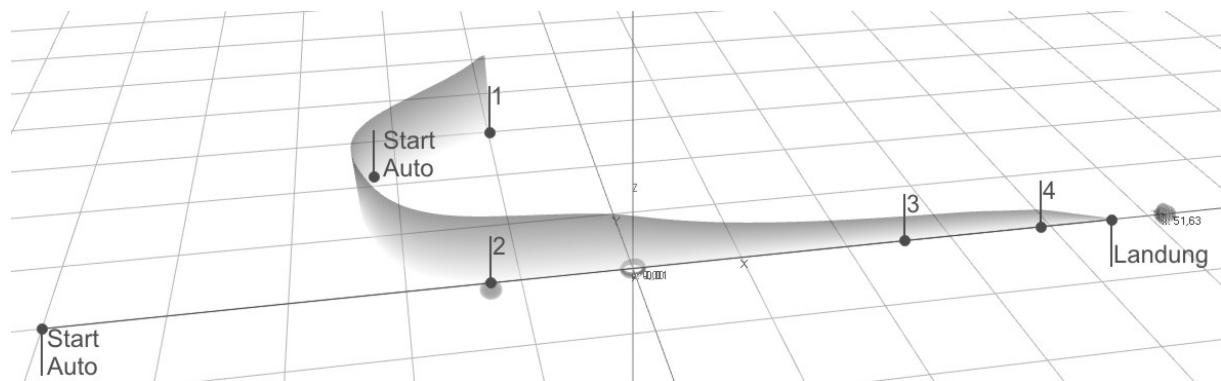


Abb. 7.21: Simulation: automatische Landung auf Fahrzeug, 3D-Trajektorie des UAs: 1) GPS-Modus 2) Funkmessung Modus 3) USS-Modus 4) Endanflug Modus

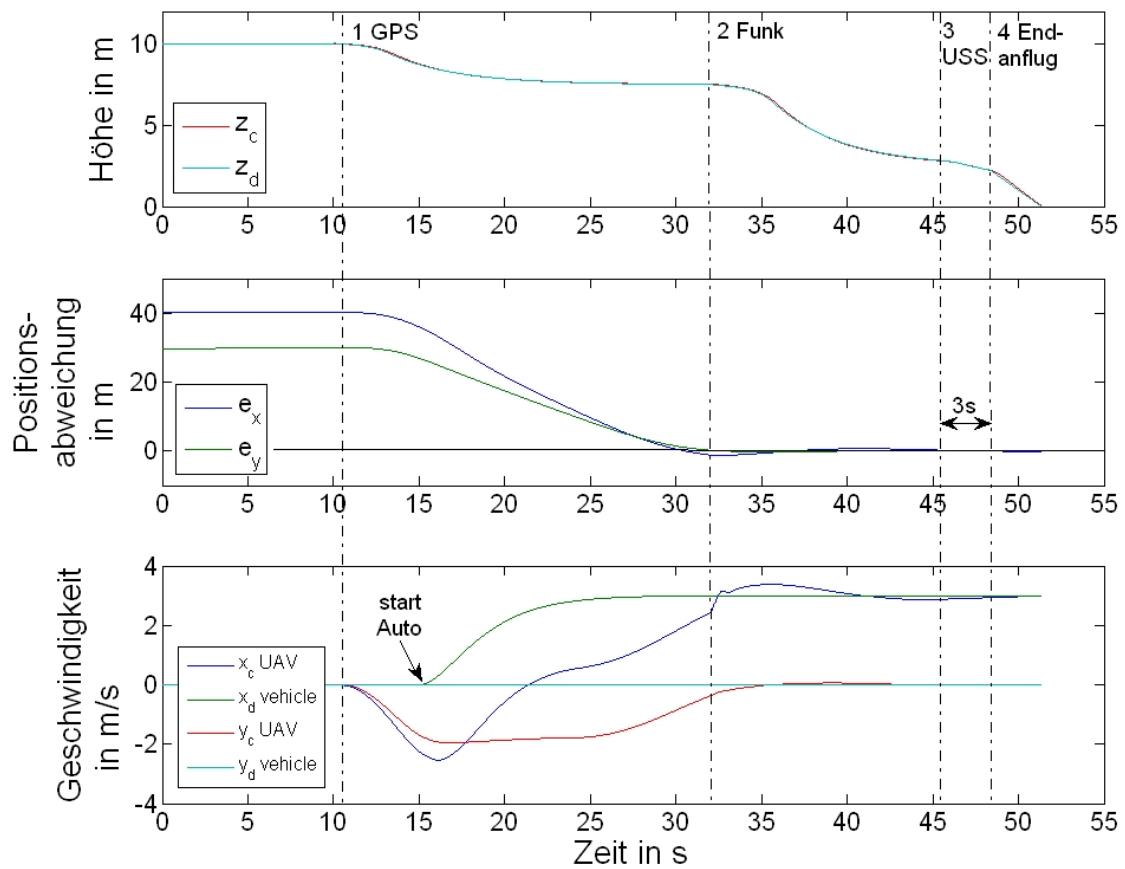


Abb. 7.22: Simulation: automatische Landung auf Fahrzeug

Die Landesequenz wird an Punkt 1 eingeleitet und das UAS beginnt, in Richtung des Fahrzeuges zu fliegen. Nach einer kurzen Strecke beginnt das Fahrzeug zu beschleunigen und fährt mit einer Geschwindigkeit von  $3\text{m/s}$  entlang der x-Achse Richtung Landepunkt. Es ist zu erkennen, dass das UAS sowohl weiter auf das Fahrzeug zufliest, als sich auch an die Bewegung des Fahrzeugs anpasst. Im Punkt 2 ist die Annäherung vollendet und der nächste Zustand im Landeargorithmus wird aktiviert (Funkmessung Modus). Jetzt laufen der Reihe nach die restlichen Phasen des Landeargorithmus ab (siehe auch Abb. 7.22). Am Ende landet dann das UAS mit einer Genauigkeit von  $0.14\text{m}$ .



# Kapitel 8

## Experiment

In diesem Kapitel werden die experimentellen Ergebnisse vorgestellt. Die Flugversuche orientieren sich an denen, die auch in den Simulationen durchgeführt wurden. Alle Experimente fanden im Freien statt. Abweichungen zwischen Simulation und Experiment können meistens durch vorhandene Windbedingungen erklärt werden. Da die Experimente fast alle in größeren Höhen stattfanden, sind die aktuellen Windbedingungen nur abschätzbar. Zu diesem Zweck wurden Windmesser am Boden aufgestellt, um einen Anhaltspunkt für die vorhandenen Windbedingungen zu erhalten.

### 8.1 2D-Wegpunktregler

#### 8.1.1 Höhenregelung

In Abb. 8.1 ist das experimentelle Ergebnis der in 7.2.1 beschriebenen Simulation dargestellt.

Es ist zu erkennen, dass das Ergebnis des Experiments fast genau dem der Simulation entspricht. Im Experiment konnte nachgewiesen werden, dass die Sollhöhe beim Absteigen nicht nennenswert unterschritten wurde, was eine Kernanforderung an den Höhenregler ist. Der Fehler von  $0.3 - 0.4m$  liegt im Rahmen des normalen Schwingverhaltens im Freien (ca.  $40cm$ , vgl. Abb. 8.10) um die Sollhöhe bei der automatischen Höhenhaltung.

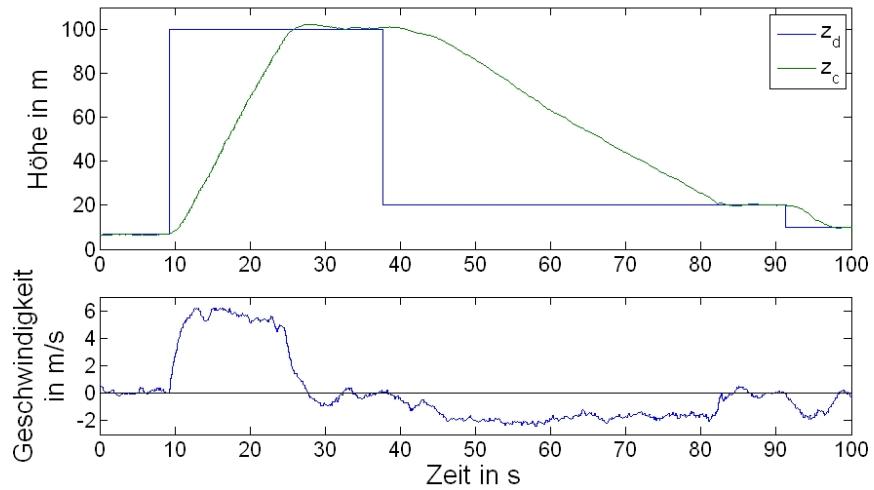


Abb. 8.1: Experiment: Höhenregler

### 8.1.2 Streckenflug

Abb. 8.2 zeigt den experimentellen Ablauf des 320m-Streckenflugs aus 7.2.3.

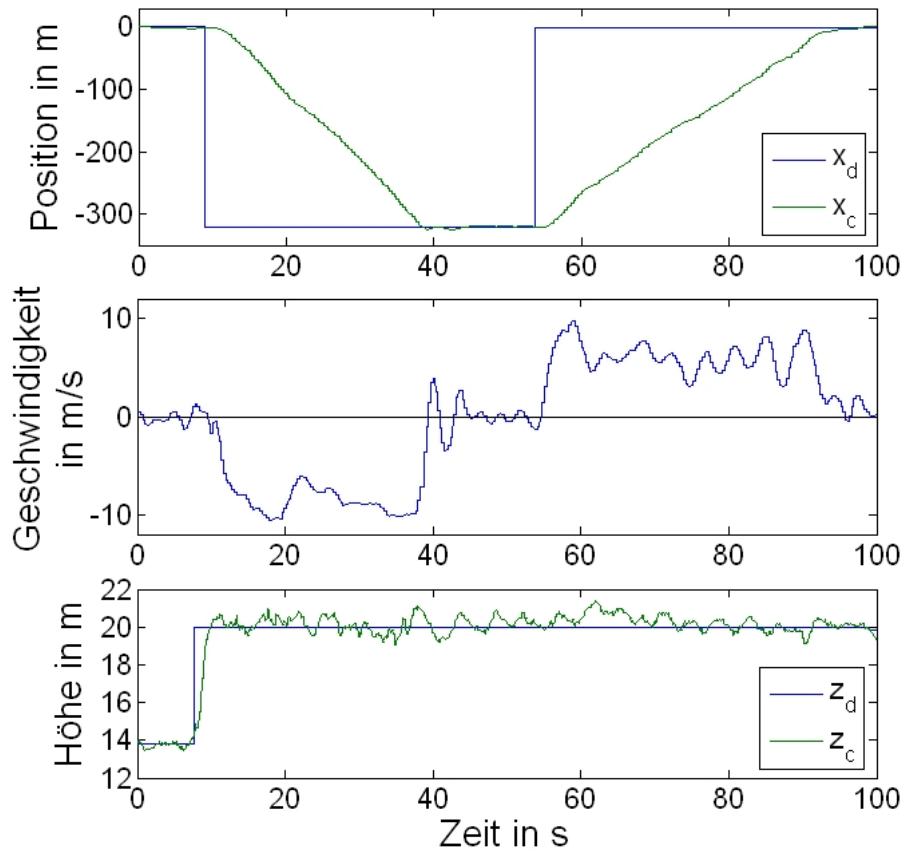


Abb. 8.2: Experiment: Streckenflug

Der chronologische Ablauf des Experimentes ist im Vergleich zur Simulation fast identisch. Sowohl die durchschnittlichen Fluggeschwindigkeiten beim Hin- und Rückflug als auch das Überschwingverhalten entspricht denen in der Simulation. Ein Unterschied ist aber in der fluktuierenden Geschwindigkeit und Höhe während des Streckenflugs zu erkennen. Zum einen resultieren diese durch Windböen während des Flugs und zum anderen durch Modifikationen in der Flugrichtung in Folge der Sollstandlinienregelung.

### 8.1.3 Simultane Höhen- und Positionsänderung

In Abb. 8.3 ist die experimentelle Trajektorie der gleichzeitigen Positions- und Höhenänderung aus 7.2.5 dargestellt.

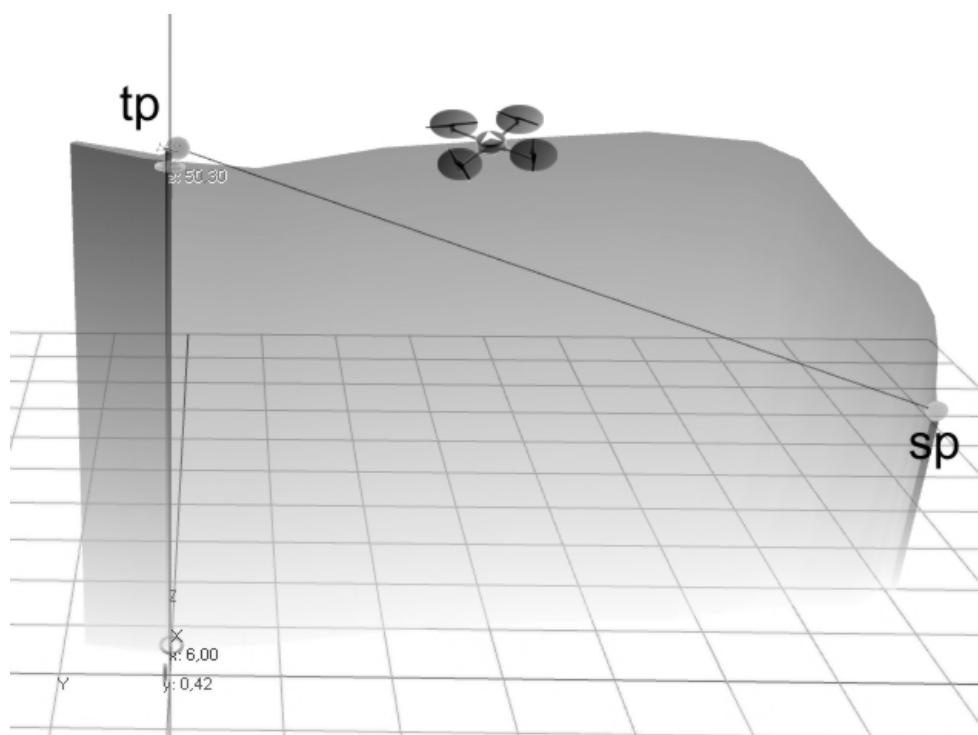


Abb. 8.3: Experiment: simultane Höhen- und Positionsänderung

Wie in der Simulation versucht das UAS zuerst, seine Sollhöhe zu erreichen, bevor es Richtung Zielposition fliegt. Auf Grund des gemessenen Rückenwinds (am Boden gemessen) von  $3 - 5 \frac{m}{s}$  ist sowohl die Delle in der Trajektorie beim Aufsteigen, als auch das größere Überschwingen an der Zielposition im Vergleich zur Simulation zu erklären.

Auch das in Abb. 8.4 dargestellte Experiment des Viereckfluges (vgl. Abb. 7.8) stimmt mit den Simulationsergebnissen fast überein.

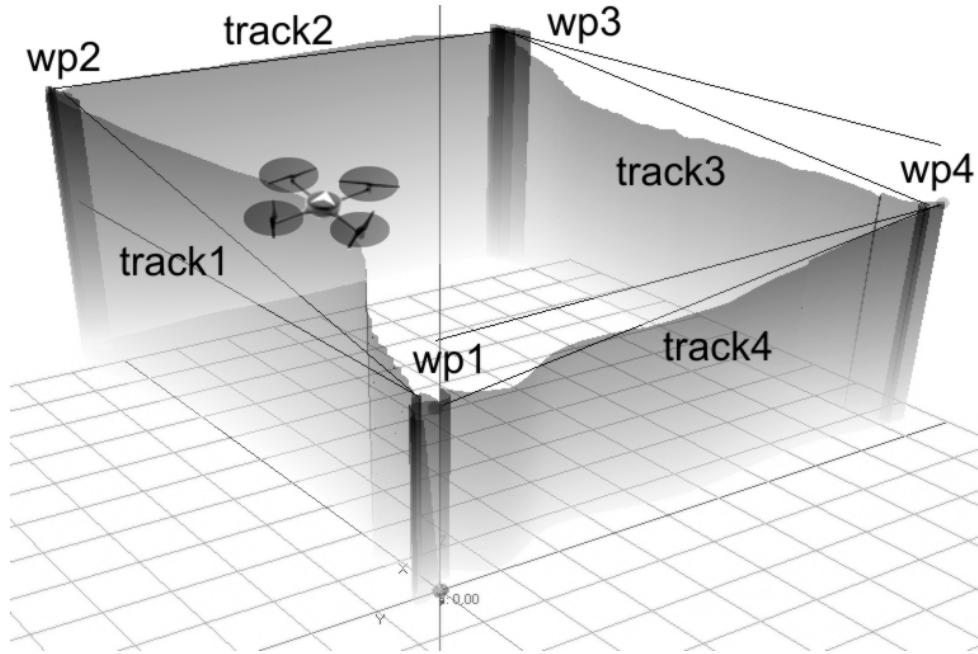


Abb. 8.4: Experiment: Abflug eines Vierecks

Die Unterschiede basieren auch hier wiederum auf der Anwesenheit von Wind (bis zu  $5 \frac{m}{s}$  auf dem Boden) und zusätzlichen Windböen.

Zum Beispiel ist die Trajektorie während des Fluges von wp 1 zu wp 2 im ersten Teil unterschiedlich. Das hat zwei Ursachen, die beide durch den Wind bedingt sind. Zum einen wird das UAS während der Aufstiegsphase durch den Wind weggedrückt, da in Folge des  $h_{loss}$ -Terms der P-Anteil des Reglers unterdrückt wird. Zum zweiten baut sich in Folge des statischen Winds während des Positionshaltens an wp 1 der integrale Anteil des Reglers auf. Das bedeutet, das UAS befindet sich trotz Positionshaltung in einer geneigten Lage. Wird jetzt der Schub in Folge der Höhenregelung erhöht, verlässt das UAS seine Position, da der Schubvektor nicht mehr parallel zum Erdschwerevektor  $g$  verläuft.

Des Weiteren ist während des Fluges von wp 4 nach wp 1 die Auswirkung der Sollstandlinienregelung zu erkennen. Das UAS wird zwar durch den Wind von seiner Trajektorie gedrückt, fliegt aber immer wieder zurück.

## 8.2 Trajektorienregler

In diesem Abschnitt werden die experimentellen Ergebnisse des Trajektorienreglers vorgestellt.

### 8.2.1 Lineare 3D-Trajektorie

In Abb. 8.5 ist das Experiment einer linearen 3D-Trajektorie dargestellt. Das UAS sollte zwischen zwei Wegpunkten hin- und zurück fliegen. Die beiden Wegpunkte lagen ca. 150m voneinander entfernt mit einem Höhenunterschied von ca. 90m. Als Geschwindigkeitsfunktion wurde eine  $6s - x^2$ -Rampe verwendet mit einer Reisegeschwindigkeit  $v_{cr} = 3 \frac{m}{s}$ .

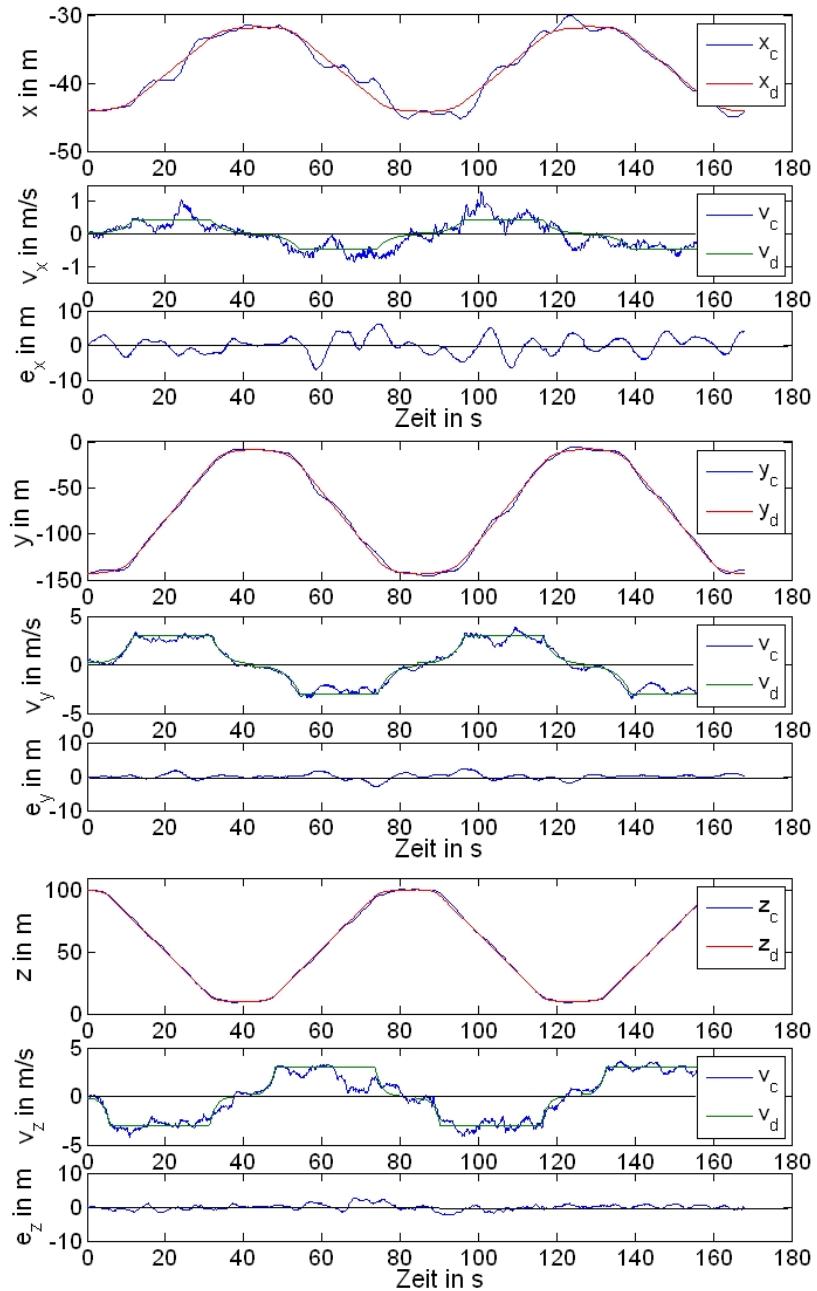


Abb. 8.5: Experiment: lineare 3D Trajektorie,  $6s - x^2$ -Rampe,  $v_{cr} = 3 \frac{m}{s}$

Es ist zu sehen, dass die experimentellen Ergebnisse ähnlich denen der Simulationen sind. Nur in der  $x$ -Richtung ist eine nennenswerte höhere Positionsabweichung zu erkennen

(siehe Tabelle 8.1). Diese Abweichung ist eine Folge der windigen Wetterbedingungen im Freien. Die Windgeschwindigkeit am Boden betrug entlang der  $x$ -Richtung  $2 - 4 \frac{m}{s}$ .

Tab. 8.1: Über die Flugzeit durchschnittliche Positions- und Geschwindigkeitsabweichungen von der linearen 3D-Trajektorie

	$x$	$y$	$z$
Positionsabweichung	1.99m	0.6m	0.63m
Geschwindigkeitsabweichung	0.27m/s	0.15m/s	0.47m/s

### 8.2.2 Geozentrischer Kreis

Auch die experimentellen Ergebnisse eines geozentrischen Kreisfluges im Freien stimmen fast mit denen der Simulation überein. Die Abbildungen 8.6 und 8.7 zeigen die Ergebnisse des Experimentes im Detail. Der Radius des Kreises betrug 30m mit einer Reisegeschwindigkeit des UASs von  $v_{cr} = 3 \frac{m}{s}$ , einer  $6s - x^2$ -Rampe als Geschwindigkeitsfunktion und dem Tangentialvektor als Geschwindigkeitsvektor. Die Windgeschwindigkeit betrug am Boden bis zu  $5 - 6m/s$

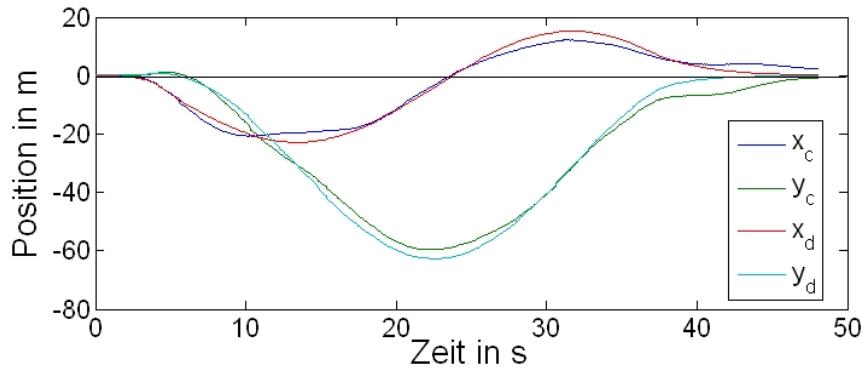


Abb. 8.6: Experiment: Geozentrischer Kreis mit 30m-Radius und 6s- $x^2$ -Rampe,  $v_{cr} = 3 \frac{m}{s}$

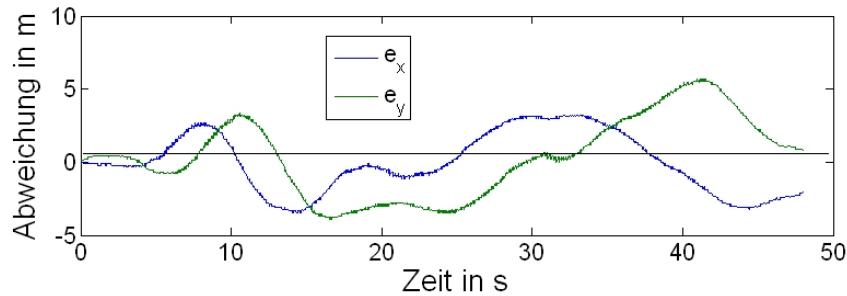


Abb. 8.7: Experiment: Positionsabweichung während des geozentrischer Kreises

Sowohl die Simulationen als auch die Experimente haben die Fähigkeiten des Trajektorienreglers gezeigt, nicht linearen Trajektorien wie z.B. Kreisen zu folgen.

### 8.3 Objekt (Fahrzeug) verfolgen

In diesem Abschnitt wird die Fähigkeit des Trajektorienreglers experimentell gezeigt, Fahrzeuge zu verfolgen. Zu diesem Zweck wurde am Fahrzeug ein GPS-Empfänger montiert, der sowohl seine Position als auch seine Geschwindigkeit mit 5Hz zum UAS schickt (vgl. Abschnitt 6.9).

In Abb. 8.8 ist das Ergebnis des Experimentes zu sehen. Dargestellt sind die Positions- und Geschwindigkeitsdaten von der  $x$ -Achse (Latitude), sowie deren Abweichungen.

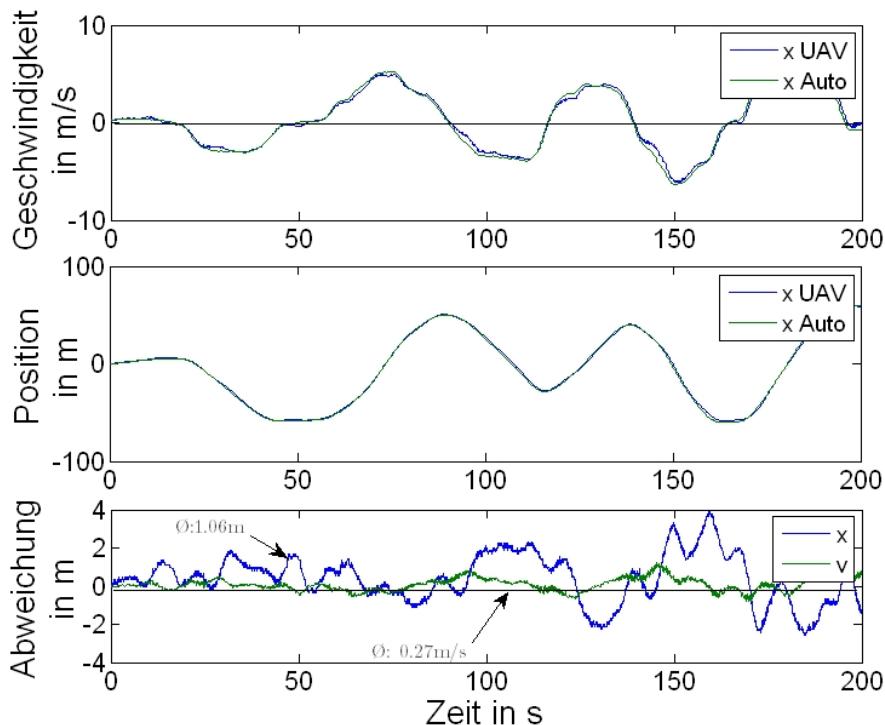


Abb. 8.8: Experiment: Verfolgen eines Fahrzeuges

Es ist zu erkennen, dass das UAS in der Lage ist, dem Fahrzeug mit einer durchschnittlichen Genauigkeit von  $1.06\text{m}$  zu folgen. Die Geschwindigkeitsabweichung betrug dabei  $0.27\frac{\text{m}}{\text{s}}$ . Die Genauigkeit im Experiment ist besser als die in der Simulation (vgl. 7.4). Das begründet sich darin, dass in der Simulation größere Geschwindigkeitsänderungen angenommen wurden als im Experiment aufgetreten sind.

Die exklusive Verwendung eines GPS-Empfängers am Fahrzeug zur Bestimmung der Position und der Geschwindigkeit hat einen Nachteil, der auch schon im Kapitel 4.3.2 adressiert

wurde. Die Signale des GPS-Empfängers haben eine Verzögerung von grob  $\frac{1}{2}s$ . Das bedeutet, dass die gemessene Position des Fahrzeugs von der realen immer um  $\frac{v_{Fahrzeug}}{2}m$  falsch ist. Um diesen Fehler zu korrigieren, müsste man theoretisch auch im Fahrzeug eine Avionik einbauen, die diesen Versatz wieder kompensiert. Zur reinen Validierung des Reglers wurde darauf aber verzichtet.

## 8.4 Automatische Landung

Für die im folgenden beschriebenen Experimente wurde folgender Versuchsaufbau verwendet (siehe Abb. 8.9). Der Abstand der einzelnen Sensorknoten voneinander betrug  $5m$  (also  $d = 2.5m$  vgl. Abb. 5.2).



Abb. 8.9: Versuchsaufbau

Abb. 8.10 zeigt eine 3D-Trajektorie eines zweiminütigen Fluges im Freien unter variablen Windbedingungen ( $0 - 1.5 \frac{m}{s}$ ). Die Position wurde mit Hilfe des Ultraschallmesssystems (siehe Abschnitt 5.2) ermittelt. Das UAS hatte die Aufgabe, im Zentrum der vier Sensorknoten in einer bestimmten Höhe (hier ca.  $1.2m$ ) zu schweben.

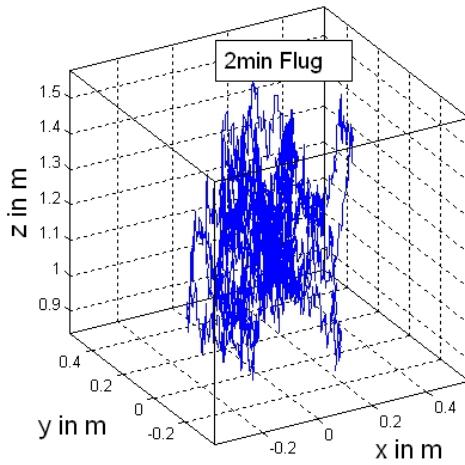


Abb. 8.10: Experiment: Trajektorie während Positions haltung

Es ist zu erkennen, dass der Trajektorienregler trotz variabler Windbedingungen in der Lage ist, das UAS innerhalb eines Würfels mit ca.  $0.4m$  Kantenlänge zu halten. In Abb. 8.11 ist der Abstand zwischen UAS und Zentrum während des zweiminütigen Fluges zu sehen. Die durchschnittliche Abweichung lag bei  $0.17m$ .

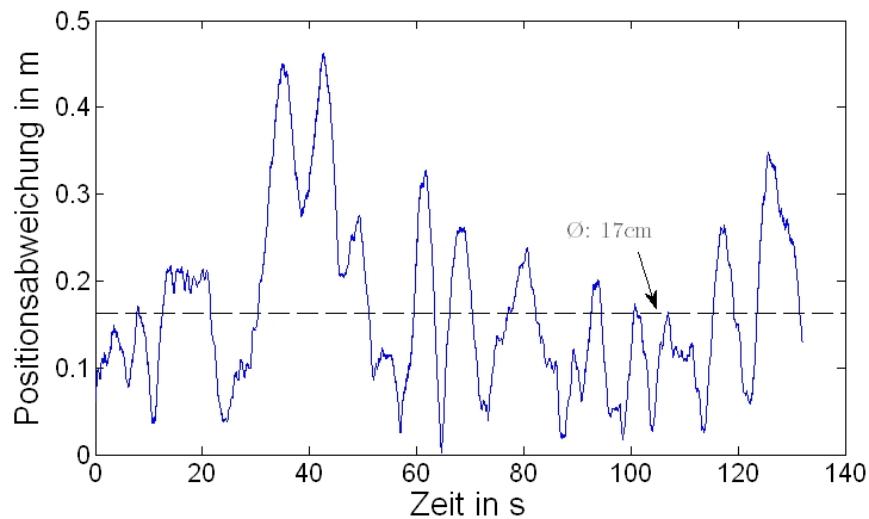


Abb. 8.11: Experiment: Abstand zur Mitte während Positions haltung

Das bedeutet, dass neben der Fähigkeit, die Position des UASs mit einer Genauigkeit von wenigen Zentimetern zu lokalisieren, auch die zweite Vorbedingung zum punktgenauen Landen erfüllt werden kann. Das UAS kann durch den Trajektorienregler (oder auch Positionsregler) für mindestens drei Sekunden im „Enter-Endanflug-Radius“  $r_e$  platziert werden kann.

In Abb. 8.12 ist eine typische Landesequenz dargestellt, die während eines Landevorgangs jedesmal durchlaufen wird. Die Anflughöhe  $z_{cr}$  während des Experimentes betrug  $7.5m$ . In diesem und folgenden Experimenten war der „Enter-Endanflug-Radius“  $r_e = 0.3m$  und der „Enter-Endanflug-Radius“  $r_x = 0.5m$ . Zur Visualisierung der Positionsdaten, die in Abb. 8.12 dargestellt sind, wurde während der Phasen 1 (Annäherung über GPS) und 2 (Annäherung über Funklaufzeitmessung) die Position des UASs über die Funklaufzeitmessung berechnet. Während der Phasen 3 (USS) und 4 (Endanflug) wurde die Ultraschallmessung verwendet. Während des Experimentes wurde in Phase 1 die Position durch den GPS-Sensor bestimmt.

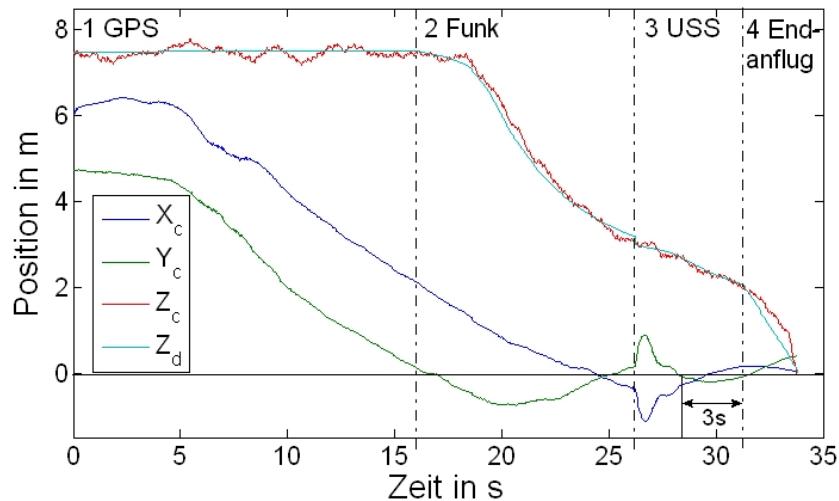


Abb. 8.12: Experiment: Landeanflug

Beim Übergang zwischen Funklaufzeit- und Ultraschallmessung ist eine plötzlich auftretende Änderung der aktuellen Positionsberechnung von ca.  $1m$  zu erkennen. Das begründet sich dadurch, dass, sobald die Ultraschall-Phase aktiv ist, die Positions berechnung genauer ist und vorherige Messungenauigkeiten eliminiert werden. Die Landegenauigkeit bei diesem Experiment lag bei  $0.39m$ .

In Abb. 8.13 sind die Ergebnisse von 15 aufeinanderfolgenden Landungen grafisch dargestellt.

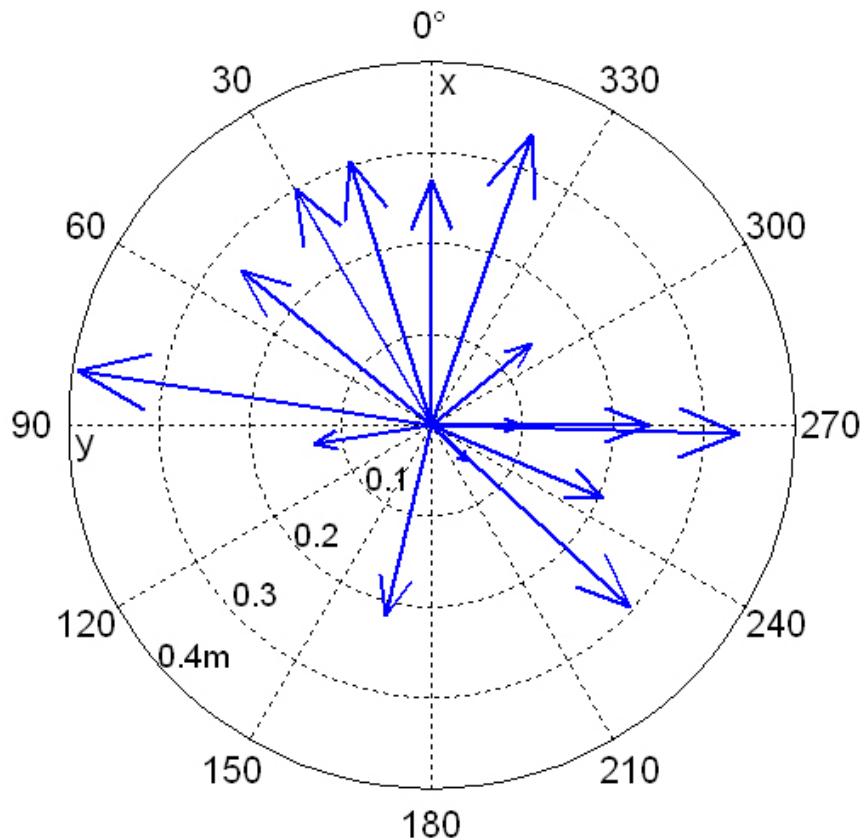


Abb. 8.13: Experiment: Landungen

Der durchschnittliche Positionsfehler nach den Landungen lag bei ca.  $0.24m$ , bei einem maximalen Fehler von  $0.39m$ . Es ist zu erkennen, dass das UAS bei diesen 15 Landungen immer innerhalb des maximal zugelassenen Radius von  $r_x = 0.5m$  gelandet ist.

In zwei Fällen kam es vor, dass das UAS während des Landeendanfluges den  $r_x$ -Radius verließ, dadurch die Endanflug-Phase unterbrochen und wieder die USS-Phase aktiviert wurde. In Abb. 8.14 ist so ein Fall dargestellt.

Die Experimente bezüglich der Landungen haben gezeigt, dass durch dieses Verfahren ein UAS „zuverlässig“ innerhalb eines vordefinierten Bereiches landen kann. Diese Experimente sind zwar kein Beweis, dass es immer funktioniert, aber während sämtlicher Versuche ist das UAS nicht einmal außerhalb der vorgegebenen Landezone gelandet. Die untere Grenze bezüglich der Genauigkeit wird nur durch die Fähigkeit des UASs festgelegt, wie gut es trotz Windeinflüsse die Position halten kann. Sobald es in der Lage ist, seine Position für mehr als  $t_{landing}$  Sekunden innerhalb des Radius  $r_e$  zu halten, kann die Genauigkeit der Landung falls erforderlich erhöht werden.

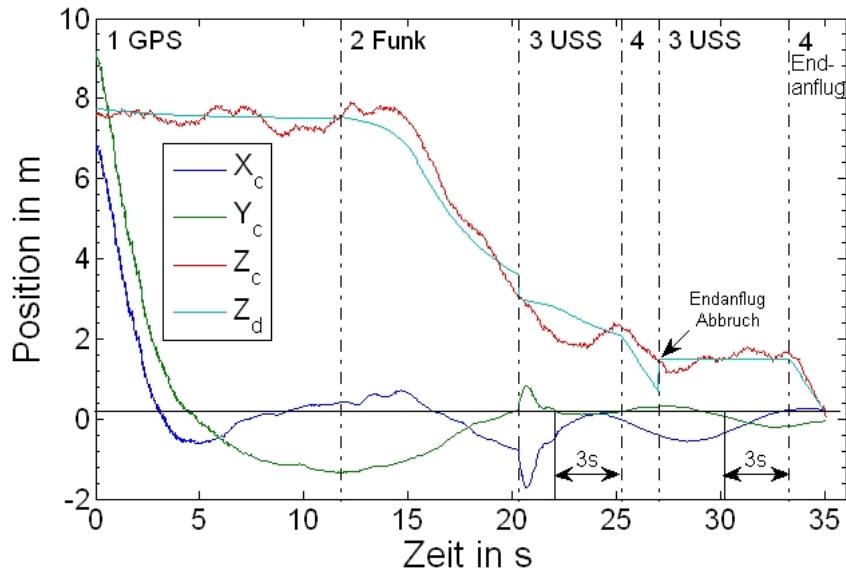


Abb. 8.14: Experiment: Landeanflug mit Endanflug-Abbruch

## 8.5 Landung auf fahrendem Fahrzeug

Um die Landung auf einem fahrenden Fahrzeug zu testen, wurde das komplette Landesystem inklusive Bodenstation und GPS-Empfänger auf einem Anhänger montiert. Der Versuchsaufbau ist in Abb. 8.15 zu sehen. Der Basisabstand zwischen den vier Sensorknoten wurde dazu von 5m auf 2m ( $d = 1m$ ) reduziert.



Abb. 8.15: Fahrzeug mit Landevorrichtung

In Abb. 8.16 ist der sequenzielle Ablauf einer Landung dargestellt. Die Geschwindigkeit des Fahrzeuges betrug im Schnitt ca.  $8km/h(\approx 2.2\frac{m}{s})$

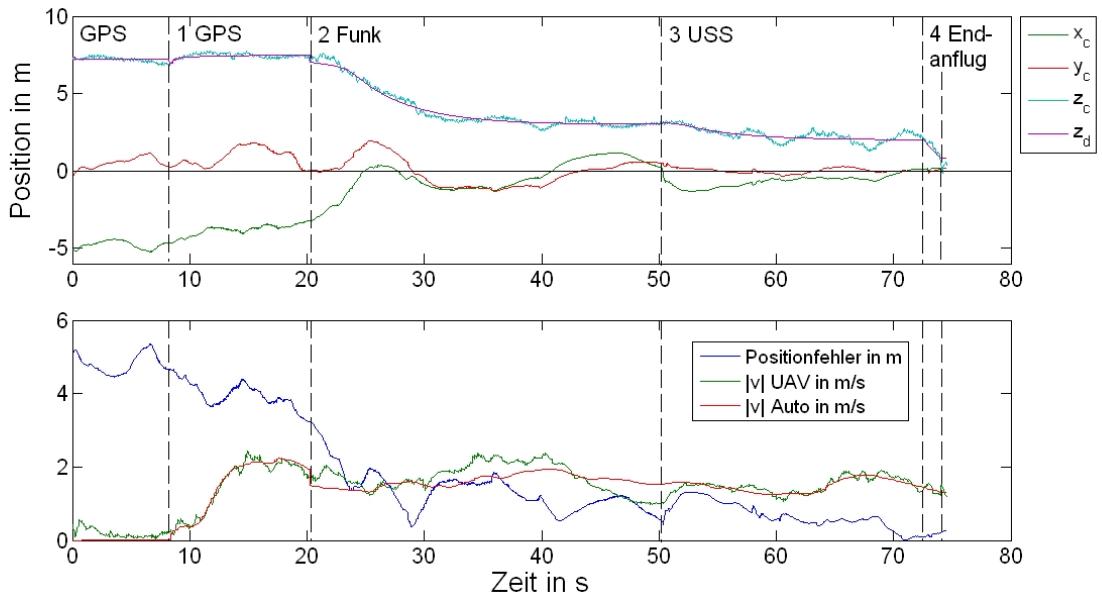


Abb. 8.16: Experiment: Landeablauf auf fahrendem Fahrzeug

Im Vergleich zur statischen Landung (siehe Abb. 8.12) ist zu erkennen, dass die Phasen 2 und 3 länger dauern. Das begründet sich darin, dass es einige Zeit dauern kann, bis sich das UAS an die Geschwindigkeit des Fahrzeuges angepasst hat, insbesondere dann, wenn die Geschwindigkeit des Fahrzeuges variiert. Vor allem der Übergang in die Endanflugphase (Phase 4) kann länger dauern, da das UAS sich für  $t_{landing}$ -Sekunden in dem relativ kleinen Radius  $r_e$  aufhalten muss. Jede Geschwindigkeitsänderung des Fahrzeuges zu diesem Zeitpunkt würde wieder zu einem Positionsfehler führen, der größer sein könnte als der Radius  $r_e$ . Eine Möglichkeit, dieses Problem zu verringern, wäre es, die Zeit  $t_{landing}$  zu verkürzen. Eine weitere Verbesserung wäre es, wenn man auf die Beschleunigungen des Fahrzeuges ohne zeitliche Verzögerung direkt reagieren könnte, indem man z.B. anstatt eines reinen GPS-Empfängers (vgl. auch 8.3) eine Avionik als Referenz verwenden würde.

In Abb. 8.17 sind zum Abschluss die Ergebnisse dreier Fahrzeuglandungen mit unterschiedlichen Geschwindigkeiten dargestellt ( $5\text{km}/\text{h}$ ,  $8\text{km}/\text{h}$  und  $12\text{km}/\text{h}$ ).

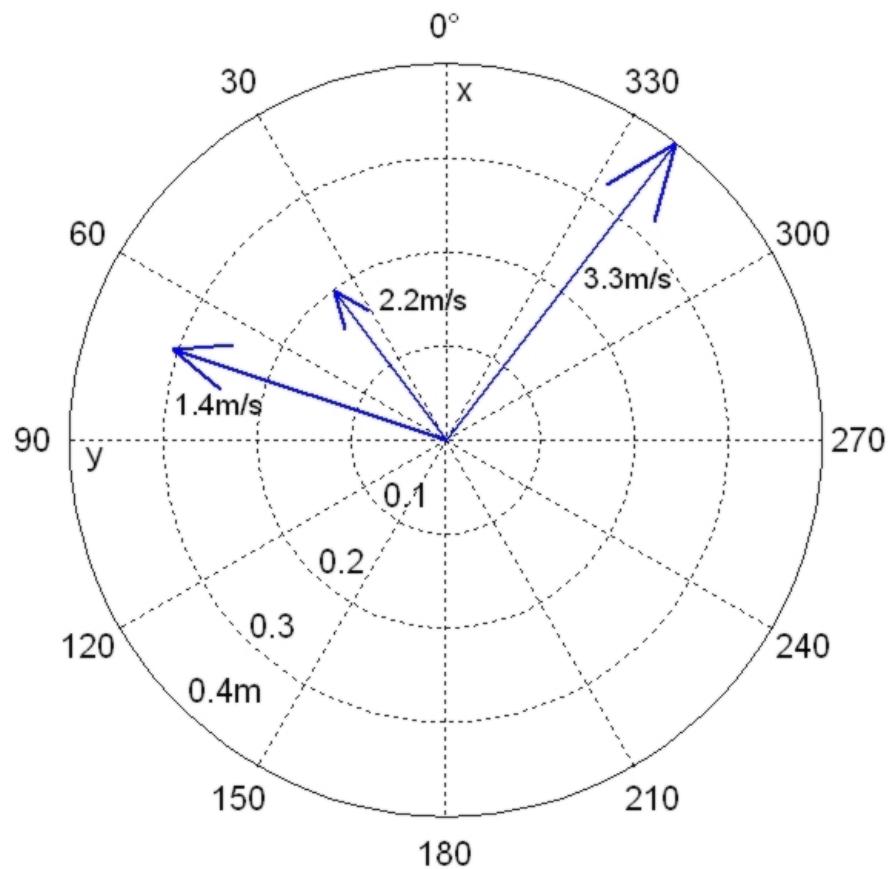


Abb. 8.17: Experiment: Landungen auf fahrendem Fahrzeug

Wie in Abb. 8.13 ist auch hier zu erkennen, dass der maximale Positionsfehler nach den Landungen immer kleiner ist als der vorgegebene (auch hier  $r_x = 0.5m$ ). Die durchschnittliche Positionsabweichung betrug  $0.3m$  bei einem maximalen Fehler von  $0.39m$ .

# Kapitel 9

## Zusammenfassung

In der letzten Vergangenheit rückten kleine unbemannte Flugsysteme immer weiter in den Fokus von Unternehmen und Forschungsgruppen. Speziell für Systeme mit Schweben- und senkrechter Start- und Landefähigkeiten wird das Spektrum an Einsatzmöglichkeiten immer größer. In dieser Arbeit wurden Methoden und Verfahren für solche VTOLs (im speziellen Quadrokopter) vorgestellt, mit Hilfe derer unterschiedliche Missionsszenarien ausgeführt werden können. Besonders der in Abschnitt 1.1 geschilderte Anwendungsfall wird durch die Ergebnisse in dieser Arbeit komplett abgedeckt.

Zunächst wurde in Kapitel 4 eine modulare Avionik vorgestellt, welche in der Lage ist, mit einer Aktualisierungsrate von bis zu einem Kiloherz Sensorwerte einzulesen, Regelalgorithmen auszuführen und im Falle eines Quadroopters neue Stellwerte für die Motoren zu berechnen. Eines der essentiellen Module einer Avionik ist eine MEMS-basierte IMU mit drei Beschleunigungs- und drei Drehratensensoren. Da diese Sensoren stark fehlerbehaftet sind, wurde ein Kalman-basiertes Filterverfahren entwickelt, das es ermöglicht, zeitlich aktuelle aber driftbehaftete Positionsdaten mit präzisen aber zeitlich versetzten Daten (z.B. GPS-Daten) zu fusionieren. Der Fokus liegt dabei darauf, dass die Daten trotz Filterung zeitlich aktuell bleiben.

Als zweites wurde in Kapitel 5 ein Messverfahren für die Nahfeldlokalisierung vorgestellt, welches auf dem Einsatz eines WSN basiert. Dieses System kombiniert die Daten von Funklaufzeit- und Ultraschallmessungen. Untersuchungen haben gezeigt, dass bei der Funklaufzeitmessung mit einer Reichweite von ca.100m die Messgenauigkeit bei ca.1m liegt. Die Genauigkeit der Ultraschallmessung dagegen liegt bei 0.05m, bei einer maximalen Reichweite von ca.10m. In Kombination mit einem GPS-Sensor konnte ein drei-phasisches Messsystem entwickelt werden, mit dem die Herausforderung des Übergangs von der groben GPS-basierten bis zur hochgenauen Nahfeld-Navigation gelöst werden kann. Dieses ist eines der Grundvoraussetzungen, um vollautomatisch von irgendeiner Position aus

punktgenau landen zu können. Des Weiteren erlaubt dieses Verfahren eine Lokalisation bei Dunkelheit.

Als nächstes wurden in Kapitel 6, 7 und 8 verschiedene Regelungsverfahren und Algorithmen entwickelt. Grundlage aller Regelungen ist bei einem Quadrokopter immer eine „gut“ funktionierende Lageregelung. Ausgehend von einem existierenden global stabilen Lageregler wurde ein neuer Lageregler entworfen. Dieser Regler ist in der Lage, Asymmetrien und Störungen im System auszugleichen, ohne instabil zu werden. Solche Störungen können z.B. durch Schwerpunktverschiebungen infolge zusätzlicher Lasten auftreten.

Aufbauend auf diesem Lageregler wurden zwei unterschiedliche Positions- bzw. Wegpunktregler entwickelt, die sich durch unterschiedliche Merkmale auszeichnen. Als erstes wurde ein 2D-Wegpunktregler vorgestellt, der eingesetzt werden sollte, wenn es darum geht, ohne Geschwindigkeitsbegrenzung eine bestimmte Zielposition zu erreichen. Der Regler ist in der Lage, während des Fluges die Sollhöhe zu halten und entlang einer Sollstandlinie zu fliegen. Der zweite Regler ist der Trajektorienregler, für den gezeigt werden konnte, dass er stabil im Sinne von Lyapunov ist. Dieser Regler wird immer dann eingesetzt, wenn der Fokus auf der Positionsgenauigkeit während des Fluges zu jedem Zeitpunkt liegt. Dieser Regler ist in der Lage, jegliche Trajektorie (wie z.B. Kreise oder Splines) in  $x$ ,  $y$  und  $z$  zu fliegen. Des Weiteren ermöglicht dieser Regler die Verfolgung von fahrenden Objekten, sofern die Position und die Geschwindigkeit des Objektes zu jedem Zeitpunkt bekannt ist. In Experimenten konnte ein fahrendes Fahrzeug (bis zu  $30\text{km/h}$ ) über die Flugzeit mit einer durchschnittlichen Genauigkeit von  $1.06\text{m}$  verfolgt werden.

In Bezug auf eine Kreistrajektorie wurden unterschiedliche Soll-Geschwindigkeitsvektoren untersucht. Es hat sich gezeigt, dass ein tangentialer Geschwindigkeitsvektor die Aufgabe erfüllt, aber nicht die beste Lösung ist. Immer dann, wenn der Radius des Kreises bekannt ist, sollte ein Vektor gewählt werden, der genau auf die nächste Position des Kreises zeigt (Sekantenvektor). Durch diese Änderung kann die Genauigkeit um 60% verbessert werden. Eine weitere Möglichkeit die Güte dieser Regler zu erhöhen ist es, eine zusätzliche modifizierte Beschleunigungsdämpfung zu verwenden, die immer nur dann eingreift, wenn das System beschleunigt, aber nicht, wenn es abbremst. Diese Fallunterscheidung verbessert die Qualität um bis zu 30% gegenüber überhaupt keiner Beschleunigungsregelung und 15% gegenüber einer richtungsunabhängigen Beschleunigungsdämpfung.

Zum Schluss wurden übergeordnete Regelungsstrategien entwickelt. Dazu gehört ein Verfahren zum automatischen Starten, welches es ermöglicht, so schnell aus einer Box heraus zu starten, dass eine Kollision mit umliegenden Gegenständen oder Strukturen (z.B. Seiten der Box) praktisch ausgeschlossen werden kann. Als letztes wurde ein Algorithmus zum vollautomatischen Landen vorgestellt. Experimente haben gezeigt, dass ein Quadrokopter bei 15 aufeinander folgenden Landungen immer innerhalb eines vorgegebenen Radius von

$0.5m$  gelandet ist. Die durchschnittliche Positionsgenauigkeit lag bei  $0.24m$ , bei einem maximalen Fehler von  $0.39m$ . Des Weiteren konnte gezeigt werden, dass dieser Landealgorithmus zusammen mit dem Trajektorienregler in der Lage ist, einen Quadrokopter automatisch auf einem fahrenden Fahrzeug landen zu lassen. In Experimenten konnte der Quadrokopter durchschnittlich in einem Radius von  $0.3m$  landen, bei einer Geschwindigkeit bis zu  $12km/h$ .

Die Simulationen und Experimente haben gezeigt, dass die Verfahren und Algorithmen, die in dieser Arbeit entwickelt wurden, ein weiterer Schritt hin zur vollautonomen Operation von Quadrooptern sind. Der genannte Anwendungsfall kann mit diesen Verfahren und Algorithmen komplett durchgeführt werden. Die vorgestellten Verfahren sind dabei nicht nur auf Quadroopter beschränkt, sondern können für fast alle Drehflüglerkonfigurationen adaptiert werden.

## Ausblick

Während der Simulationen und Experimente haben sich einige Verbesserungsmöglichkeiten zu den einzelnen Themengebieten ergeben. Diese Aspekte werden im folgenden näher erläutert.

- **Bakensystem:**

Um die absolute Messgenauigkeit des Bakensystems zu erhöhen, könnten zusätzliche Sensorknoten verwendet werden. Des Weiteren könnte eine veränderte Anordnung der Knoten im Raum die Messgenauigkeit vor allem in Bezug auf die Höhenbestimmung erhöhen. Durch diese beiden Änderungen wird die Komplexität der Berechnung erheblich erhöht werden (vgl. Abschnitt 5.2.2), welche in Zukunft aber durch leistungsstärkere Prozessoren wieder kompensiert werden wird.

- **2D-Wegpunktregler:**

In Bezug auf den 2D-Wegpunktregler gibt es mehrere Aspekte, durch die die Qualität der Regelung verbessert werden kann. Zum einen sind da die ausgegrauten Phasen in Abb. 6.3 auf Seite 63 zu nennen. Während des Streckenfluges kann es vorkommen, dass trotz maximal zulässigem Stellwinkel und Schub der Positionsfehler während der Transition auf Grund von z.B. Gegenwind nicht kleiner wird. Tritt dieser Fall ein, müsste der maximal zulässige Schub für die Höhenregelung bzw. der maximal zulässige Sollwinkel temporär erhöht werden. Eine weitere Verbesserung während der Transition würde durch das Hinzufügen eines integralen Regelanteils bzgl. des „cross track“-Fehlers erreicht werden. Dieser Anteil ist notwendig, um permanente Abweichungen von der Sollstandlinie auf Grund von Seitenwinden auszugleichen.

Nähert sich der Quadrokopter am Ende einer Trajektorie dem Zielpunkt, wird, wie in Abschnitt 6.3.2 Phase 5 beschrieben, die ATSD aktiv. Es ist zu erwarten, dass eine Änderung der Funktion für  $x_b$  die Regelgüte verändert. Es müsste evaluiert werden, welches die optimale Abbremsfunktion ist. Abschließend sollte untersucht werden, inwieweit sich die 2D-Wegpunktregelung auch für nicht lineare Trajektorien anpassen lässt, um zum Beispiel größere Bogen ohne Geschwindigkeitsbegrenzung fliegen zu können.

- **Trajektorienregler:**

Um die Qualität von vorher bekannten Kurvenflügen zu erhöhen, wurde in Abschnitt 7.3.2 der Geschwindigkeitsvektor modifiziert. Diese Modifizierung sollte auch bei unbekannten Kurvenradien z.B. während des Verfolgens von Objekten vorgenommen werden. D.b., dass sich der tangentiale Geschwindigkeitsvektor während des Kurvenfluges dynamisch an die Kurve anpassen muss. Zur Berechnung von einhaltbaren Trajektorien gibt es in der Robotik den klassischen Weg, die Trajektorie über den Ruck („jerk“, Ableitung der Beschleunigung) zu berechnen. Ist z.B. der maximale Wert für den Ruck des Quadroopters bekannt, können Trajektorien berechnet werden, die exakt abgeflogen werden können. Diese Verfahren sind „sehr“ rechenaufwändig, können aber in Zukunft durch leistungsstärkere  $\mu$ -Prozessoren „online“ berechnet werden. Eine letzte Verbesserungsmöglichkeit bezieht sich auf die Höhenregelung. Sowohl beim 2D-Wegpunktregler, als auch bei der Trajektorienregelung wird der Stellwert  $y_h$  immer gleichverteilt auf alle Motoren gegeben. Befindet sich der Quadroopter aber nicht in einer horizontalen Lage, ist diese Methode nicht optimal. Es sollte untersucht werden, inwieweit der Stellwert  $y_h$  „intelligenter“ auf die Motoren verteilt werden kann.

- **Automatische Landung auf fahrendem Fahrzeug:**

Bei passenden Lichtverhältnissen können zur Unterstützung optische Sensoren bei der Landung eingesetzt werden. Sowohl absolute (z.B. mit Landmarken) als auch dynamische (z.B. optischer Fluss) Verfahren können zur Stützung des eigenen Zustandes  $S_c$  verwendet werden. Z.B. bietet das Verfahren in [HHMR10] die Möglichkeit, sich der Bewegung der Landefläche in  $x$ ,  $y$  und  $z$  dynamisch anzupassen. In Kombination mit dem vorgestellten Landeargorithmus könnte so die Genauigkeit der Landung erhöht und die Dauer der Landung verringert werden. Allgemein ist es möglich, die einzelnen vorgestellten Messverfahren zur Landung durch andere (z.B. optische) auszutauschen. Falls erforderlich könnte z.B. das Ultraschallsystem durch ein optisches landmarkenbasiertes System ausgetauscht werden.

# Literaturverzeichnis

- [AAL<sup>+</sup>09] F. ADOLF, F. ANDERT, S. LORENZ, L. GOORMANN und J. DITTRICH: *An Unmanned Helicopter for Autonomous Flights in Urban Terrain*. In: T. KRÖGER und T. M. WAHL (Herausgeber): *Advances in Robotics Research*, Seiten 275–285, Braunschweig, Germany, Juni 2009. German Workshop on Robotics (GWR).
- [ABH<sup>+</sup>09] M. ACHTELIK, A. BACHRACH, R. HE, S. PRENTICE und N. ROY: *Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments*. Orlando, FL, USA, April 2009. Unmanned Systems Technology XI, SPIE 733219-10.
- [AGMD07] P. ADIGBLI, C. GRAND, J.-B. MOURET und S. DONCIEUX: *Nonlinear Attitude and Position Control of a Micro Quadrotor using Sliding Mode and Backstepping Techniques*. Toulouse; France, September 2007. European Micro Air Vehicle Conference and Flight Competition (EMAV2007).
- [Ahr08] S. G. AHRENS: *Vision-Based Guidance and Control of a Hovering Vehicle in Unknown Environments*. Doktorarbeit, Massachusetts Institute of Technology, Mai 2008.
- [AOM02] E. ALTUG, J. P. OSTROWSKI und R. MAHONY: *Control of a Quadrotor Helicopter Using Visual Feedback*. Seiten 72–77, Washington, DC, 2002. IEEE International Conference on Robotics and Automation (ICRA).
- [AOT05] E. ALTUG, J. P. OSTROWSKI und C. J. TAYLOR: *Quadrotor Control Using Dual Camera Visual Feedback*. International Journal of Robotics Research, 24(5):329–341, Mai 2005.
- [APG10] B. AHMED, H. R. POTA und M. GARRATT: *Flight control of a rotary wing UAV using backstepping*. Int. J. Robust Nonlinear Control, 20:639–658, Mai 2010.

- [AS06] T. ABDELHAMID und M. STEPHEN: *Attitude Stabilization of a VTOL Quadrotor Aircraft*. Band 14. IEEE Transactions on Control Systems Technology, Mai 2006.
- [Asc10] ASCENDING TECHNOLOGIES GMBH: 2010.  
URL: <http://www.asctec.de/>.
- [BA05] L. BEJI und A. ABICHOU: *Trajectory Generation and Tracking of a Mini-Rotorcraft*. Band 18, Seiten 2618–2623. IEEE International Conference on Robotics and Automation (ICRA), April 2005.
- [BBT07] H. BOUADI, M. BOUCHOUCHA und M. TADJINE: *Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor*. International Journal Of Applied Mathematics and Computer Sciences (AMCS), 4(2), 2007.
- [BF07] S. BAYRAKTAR und E. FERON: *Experiments with small helicopter automated landings at unusual attitudes*. CoRR, abs/0709.1744, 2007. <http://arxiv.org/abs/0709.1744>.
- [BHC08] Y. BOUKTIR, M. HADDAD und T. CHETTIBI: *Trajectory planning for a quadrotor helicopter*. Seiten 1258–1263, Ajaccio, Corsica, France, Juni 2008. Mediterranean Conference on Control and Automation (MED).
- [BMS04] S. BOUABDALLAH, P. MURRIERI und R. SIEGWART: *Design and Control of an Indoor Micro Quadrotor*. Seiten 4393–4398, New Orleans, USA, 2004. IEEE International Conference on Robotics and Automation (ICRA).
- [BMS05] S. BOUABDALLAH, P. MURRIERI und R. SIEGWART: *Towards Autonomous Indoor Micro VTOL*, Band 18. Springer Netherlands, März 2005.
- [BNS04] S. BOUABDALLAH, A. NOTH und R. SIEGWART: *PID vs LQ Techniques Applied to an Indoor Micro Quadrotor*. Sendai, Japan, September 2004. IE-EE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Bou07] S. BOUABDALLAH: *design and control of quadrotors with application to autonomous flying*. Doktorarbeit, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, Lausanne, EPFL, 2007.
- [Bro01] R. BROCKHAUS: *Flugregelung*. Springer Verlag, Berlin, 2 Auflage, Juli 2001. ISBN: 3540418903.

- [BS05] S. BOUABDALLAH und R. SIEGWART: *Backstepping and Sliding-mode Techniques Applied to Indoor Micro Quadrotor*. Barcelona, Spain, April 2005. IEEE International Conference on Robotics and Automation (ICRA).
- [CAM<sup>+</sup>09] A. CHERIAN, J. ANDERSH, V. MORELLAS, N. PAPANIKOLOPOULOS und B. METTLER: *Autonomous Altitude Estimation of a UAV Using a Single Onboard Camera*. Seiten 3900–3905, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [CDL04] P. CASTILLO, A. DZUL und R. LOZANO: *Real-time stabilization and tracking of a four-rotor mini rotorcraft*. IEEE Trans on Control Systems Technology, 12(4):510–516, Juli 2004.
- [CGPV08] U. COPPA, A. GUARNIERI, F. PIROTTI und A. VETTORE: *Accuracy Enhancement of Unmanned Helicopter Positioning with Low Cost System*. Band 37, Beijing, 2008. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.
- [CLD05] P. CASTILLO, R. LOZANO und A. DZUL: *Stabilization of a Mini Rotorcraft with Four Rotors*. IEEE Control Systems Magazine, Seiten 45–55, Dezember 2005.
- [CMGM09] J. COURBON, Y. MEZOUAR, N. GUENARD und P. MARTINET: *Visual Navigation of a Quadrotor Aerial Vehicle*. Seiten 5315–5320, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [CMV07] C. L. CASTILLO, W. MORENO und K. P. VALAVANIS: *Unmanned Helicopter Waypoint Trajectory Tracking Using Model Predictive Control*. Athen, Greece, Juli 2007. Mediterranean Conference on Control and Automation (MED).
- [CWC07a] I. D. COWLING, J. F. WHIDBORNE und A. K. COOKE: *Optimal Trajectory Planning And LQR Control For A Quadrotor UAV*. Glasgow, Scotland, August 2007. International Control Conference (ICC).
- [CWC07b] I. COWLING, O. Y. J. WHIDBORNE und A. COOKE: *A Prototype of an Autonomous Controller for a Quadrotor UAV*. Kos, Greece, Juli 2007. IEEE European Control Conference (ECC).
- [DMB06] L. DERAFA, T. MADANI und A. BENALLEGUE: *Dynamic Modelling and Experimental Identification of Four Rotors Helicopter Parameters*. Mumbai, India, 2006. International conference on industrial technology.

- [Dor] R. DOROBANTU: *Simulation des Verhaltens einer low-cost Strapdown IMU unter Laborbedingungen.* iMAR Gesellschaft für inertiale Mess-, Automatisierungs- und Regelsysteme mbH, Im Reihersbruch 3, D-66386 St. Ingbert / Germany.  
URL: [http://www.imar.de/downloads/papers/iapg\\_1.pdf](http://www.imar.de/downloads/papers/iapg_1.pdf).
- [DSS01] X. DENG, L. SCHENATO und S. SASTRY: *Hovering Flight Control of a Micro-mechanical Flying Insect.* Orlando, Florida, USA, Dezember 2001. Conference on Decision and Control.
- [DZ] R. DOROBANTU und B. ZEBHAUSER: *Field Evaluation of a Low-Cost Strapdown IMU by means GPS.* iMAR Gesellschaft für inertiale Mess-, Automatisierungs- und Regelsysteme mbH, Im Reihersbruch 3, D-66386 St. Ingbert / Germany.  
URL: [http://www.imar-navigation.de/downloads/papers/iapg\\_2.pdf](http://www.imar-navigation.de/downloads/papers/iapg_2.pdf).
- [EKF04] M. EICH, M. KEMPER und S. FATIKOW: *A Navigation System Concept for an Indoor Micro Air Vehicle.* Braunschweig, Germany, Juni 2004. EMAV - First European Micro Air Vehicle Conference and Flight Competition.
- [EVDF10] D. EYNARD, P. VASSEUR, C. DEMONCEAUX und V. FRÉMONT: *UAV Altitude Estimation by Mixed Stereoscopic Vision.* Seiten 646–651, Taipeh, Taiwan, Oktober 2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Fac97] *Fachkunde Industrieelektronik und Informationstechnik*, Band 6. Verlag Europa Lehrmittel, 1997.
- [FMS<sup>+</sup>08] N. FRIETSCH, O. MEISTER, C. SCHLAILE, J. SEIBOLD und G. TROMMER: *Vision Based Landing System for a VTOL-MAV.* London, UK, Oktober 2008. The Navigation Conference & Exhibition.
- [FPEN06] G. FRANKLIN, J. POWELL und A. EMAMI-NAEMI: *Feedback Control of Dynamic Systems*, Band 5. Prentice Hall New Jersey, 2006.
- [GGB09] S. GRZONKA, G. GRISETTI und W. BURGARD: *Towards a Navigation System for Autonomous Indoor Flying.* Seiten 1679–1684, Kobe, Japan, Mai 2009. IEEE International Conference on Robotics and Automation (ICRA).
- [GL10] J. GUERRERO und R. LOZANO: *Flight Formation of Multiple Mini Rotorcraft based on Nested Saturations.* Seiten 634–639, Taipeh, Taiwan, Oktober

2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [GMLD08] J. F. GUERRERO-CASTELLANOS, N. MARCHAND, S. LESECQ und J. DELAMARE: *Bounded attitude stabilization: Real-time application on four-rotor mini-helicopter*. Seoul, Korea, Juli 2008. The International Federation of Automatic Control (IFAC).
- [GSGA09] R. GOEL, S. M. SHAH, N. K. GUPTA und N. ANANTHKRISHNAN: *Modeling, Simulation and Flight Testing of an Autonomous Quadrotor*. Indian Institute of Science, Bangalore, India, Mai 2009. IISc Centenary International Conference and Exhibition on Aerospace Engineering (ICEAE 2009).
- [gum10] *Gumstix*, 2010.  
URL: <http://www.gumstix.com>.
- [GW01] M. S. GREWAL und L. WEILL: *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Inc., 2001. ISBN: 0471-35032-X.
- [HCC<sup>+</sup>04] D. M. HELMICK, Y. CHENG, D. S. CLOUSE, L. H. MATTHIES und S. I. ROUMELIOTIS: *Path following using visual odometry for a Mars Rover in high-slip environments*. In: *IEEE Aerospace Conference, Big Sky*, Seiten 6–13, 2004.
- [HG10] S. HOTA und D. GHOSE: *Optimal Geometrical Path in 3D with Curvature Constraint*. Seiten 113–118, Taipeh, Taiwan, Oktober 2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [HHMR10] B. HERISSE, T. HAMEL, R. MAHONY und F.-X. RUSSOTTO: *The landing problem of a VTOL Unmanned Aerial Vehicle on a moving platform using optical flow*. Seiten 1600–1605, Taipeh, Taiwan, Oktober 2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [HHWT07] G. M. HOFFMANN, H. HUANG, S. L. WASLANDER und C. J. TOMLIN: *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*. Hilton Head, South Carolina, August 2007. AIAA Guidance, Navigation and Control Conference and Exhibit. AIAA 2007-6461.
- [HHWT09] H. HUANG, G. M. HOFFMANN, S. L. WASLANDER und C. J. TOMLIN: *Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering*. Kobe, Japan, Mai 2009. IEEE International Conference on Robotics and Automation (ICRA).

- [HLK09] J.-H. HAN, J.-S. LEE und D.-K. KIM: *Bio-inspired Flapping UAV Design: A University Perspective*. Band 7295. Health Monitoring of Structural and Biological Systems, Proc of SPIE, 2009.
- [HM04] T. HAMEL und R. MAHONY: *Pure 2D Visual Servo control for a class of underactuated dynamic system*. Seiten 2229–2235, New Orleans, LA, 2004. IEEE International Conference on Robotics and Automation (ICRA).
- [HMC02] T. HAMEL, R. MAHONY und A. CHRIETTE: *Visual servo trajectory tracking for a four rotor VTOL aerial vehicle*. Band 3, Seiten 2781–2786, Washington, DC, 2002. IEEE International Conference on Robotics and Automation (ICRA).
- [HO99] Z. HONG und J. OSTROWSKI: *Visual servoing with dynamics: control of an unmanned blimp*. Band 1, Seiten 618 – 623, Detroit, MI, USA, Mai 1999. IEEE International Conference on Robotics and Automation (ICRA).
- [Hok] *Hokuyo*.  
URL: <http://www.hokuyo-aut.jp/index.html>.
- [Hol04] F. HOLZAPFEL: *Nichtlineare adaptive Regelung eines unbemannten Fluggerätes*. Doktorarbeit, Technische Universität München, 2004.
- [HP90] H. HIRATA und J. D. POWELL: *Sample Rate Effects on Disturbance Rejection for Digital Control Systems*. Seiten 1137–1145, San Diego, CA, USA, März 1990. American Control Conference (ACC).
- [HPR08] R. HE, S. PRENTICE und N. ROY: *Planning in Information Space for a Quadrotor Helicopter in a GPS-denied Environment*. Pasadena Conference Center, Pasadena, CA, USA, Mai 2008. IEEE International Conference on Robotics and Automation (ICRA).
- [HRHM08] B. HERISSE, F.-X. RUSSOTTO, T. HAMEL und R. MAHONY: *Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow*. Seiten 801–806, Nice, France, September 2008. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [HRW<sup>+</sup>04] G. HOFFMANN, D. RAJNARAYAN, S. WASLANDER, D. DOSTAL und J. J. C. TOMLIN: *The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)*. Band 2. The 23rd Digital Avionics Systems Conference (DASC04), Oktober 2004.

- [HSDM09] S. HAN, A. D. STRAW, M. H. DICKINSON und R. M. MURRAY: *A Real-Time Helicopter Testbed for Insect-Inspired Visual Flight Control*. IEEE International Conference on Robotics and Automation (ICRA), 2009.
- [HTB<sup>+</sup>10] A. S. HUANG, S. TELLEX, A. BACHRACH, T. KOLLAR, D. ROY und N. ROY: *Natural Language Command of an Autonomous Micro-Air Vehicle*. Seiten 2663–2669, Taipeh, Taiwan, Oktober 2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [HWT08] G. M. HOFFMANN, S. L. WASLANDERY und C. J. TOMLIN: *Quadrotor Helicopter Trajectory Tracking Control*. Seite 7410, Honolulu, Hawaii, August 2008. AIAA Guidance, Navigation and Control Conference and Exhibit.
- [HWL03] B. HOFMANN-WELLENHOF, M. WIESER und K. LEGAT: *Navigation: Principles of Positioning and Guidance*. Springer Verlag, Wien, 1 Auflage, Oktober 2003. ISBN: 3211008284.
- [iBPV05] S. B. I BADIA, P. PYK und P. F. VERSCHURE: *A Biologically Based Flight Control System for a Blimp-based UAV*. Barcelona, Spain, April 2005. IEEE International Conference on Robotics and Automation (ICRA).
- [Int07] INTEL CORPORATION: *OpenCV Computer Vision Library*, 2007.  
URL: <http://www.intel.com/technology/computing/opencv/>.
- [JTB<sup>+</sup>09] R. J.D.MORE, S. THURROWGOOD, D. BLAND, D. SOCCOL und M. V.SRINIVASAN: *A Stereo Vision System for UAV Guidance*. Seiten 3386–3391, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Kem02] [Kem07] *Regelungstechnik Script WS 2002*. Carl von Ossietzky Universität Oldenburg, 2002.
- [Kem07] M. KEMPER: *Development of an Indoor Attitude Control and Indoor Navigation System for 4-Rotor-Micro-Helicopter*. Doktorarbeit, Fakultät für Kommunikationswissenschaften Università della Svizzera italiana, Uhlhornsweg 84, Februar 2007.
- [KFL08] F. KENDOUL, I. FANTONI und R. LOZANO: *Asymptotic stability of hierarchical inner-outer loop-based flight controllers*. Seiten 1741–1746, Seoul, Korea, Juli 2008. 17th IFAC World Congress.

- [KFN09] F. KENDOUL, I. FANTONI und K. NONAMI: *Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles*. Robotics and Autonomous Systems, 57(6-7):591–602, Juni 2009.
- [KL80] V. C. KLEMA und A. J. LAUB: *The singular value decomposition: Its computation and some applications*. Band AC-25. IEEE Transactions on Automatic Control, April 1980.
- [KMF03] M. KEMPER, M. MERKEL und S. FATIKOW: *A Rotorcraft Micro Air Vehicle for Indoor Applications*. Seiten 1215–1220, Coimbra, Portugal, Juni 2003. International Conference on Advanced Robotics (ICAR).
- [KN09] F. KENDOUL und K. NONAMI: *A Visual Navigation System for Autonomous Flight of Micro Air Vehicles*. Seiten 3888–3893, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Kno00] T. KNOHL: *Anwendung künstlicher neuronaler Netze zur nichtlinearen adaptiven Regelung*. Doktorarbeit, Fakultät für Elektrotechnik und Informations-technik an der Ruhr-Universität Bochum, 2000.
- [Kor02] T. KORDES: *Modelling of the Nonlinear Dynamic Behaviour of a Micro-Aerial-Vehicle (MAV) in an Environment of a Turbulent Atmosphere*. Toronto, Canada, September 2002. Congress of International Council of the Aeronautical Sciences.
- [Kor05] T. KORDES: *Auslegung eines Autopilotensystems für autonome Mikroflugzeuge*. Doktorarbeit, Technischen Universität Carolo-Wilhelmina zu Braunschweig, Juni 2005.
- [KY10] F. KENDOUL und Z. YU: *Guidance and Nonlinear Control System for Autonomous Flight of Minirotorcraft Unmanned Aerial Vehicles*. Journal of Field Robotics, 27(3):311–334, 2010.
- [Lea99] B. W. LEACH: *Low Cost Strapdown Inertial / GPS Integrated Navigation for Flight Test Requirements*. St. Petersburg, Russia, Mai 1999. RTO SCI International Conference on Integrated Navigation Systems.
- [LLF<sup>+</sup>06] K. LILLYWHITE, D. LEE, S. FOWERS, A. DENNIS, B. NELSON und J. ARCHIBALD: *An Embedded Vision System for an Unmanned Four-rotor Helicopter*. Boston, MA, USA, Oktober 2006. 24th Conference on Intelligent robots and computer vision.

- [LSP08] S. LANGE, N. SÜNDERHAUF und P. PROTZEL: *Autonomous Landing for a Multirotor UAV Using Vision*. Seiten 482–491, Venice, Italy, November 2008. International Conference on Simulation, Modelling and Programming for Autonomous Robots (SIMPAR).
- [LZF09] S. LEVEN, J.-C. ZUFFEREY und D. FLOREANO: *A Minimalist Control Strategy for Small UAVs*, Seiten 2873–2878, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [MB04] A. MOKHTARI und A. BENALLEGUE: *Dynamic Feedback Controller of Euler Angles and Wind parameters estimation for a Quadrotor Unmanned Aerial Vehicle*. Seiten 2359–2366, New Orleans, LA, 2004. IEEE International Conference on Robotics and Automation (ICRA).
- [MBD06] A. MOKHTARI, A. BENALLEGUE und B. DAACHI: *Robust Feedback Linearization and GH Controller for a Quadrocopter Unmanned Aerial Vehicle*. Journal of Electrical Engineering, 57(1):20–27, 2006.
- [MBM01] V. MISTLER, A. BENALLEGUE und N. K. M’SIRDI: *Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback*. Seiten 586–593. IEEE International Workshop on Robot and Human Interactive Communication, 2001.
- [MC99] D. MURPHYA und J. CYCON: *Applications for mini VTOL UAV for law enforcement*. Band 3577, Seiten 35–43. SPIE, Januar 1999.
- [McK04] P. MCKERROW: *Modeling the Draganflyer four-rotor helicopter*. Seiten 3596–3601, New Orleans, LA, 2004. IEEE International Conference on Robotics and Automation (ICRA).
- [MCMO09] C. MARTINEZ, P. CAMPOY, I. F. MONDRAGÓN und M. A. OLIVARES-MENDEZ: *Trinocular Ground System to Control UAVs*. Seiten 3361–3367, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Mic10] MICRODONES GMBH: 2010.  
URL: <http://www.microdones.de>.
- [Mik10] *Mikrokopter*, 2010.  
URL: <http://www.mikrokopter.de>.

- [MK06] M. MAHN und M. KEMPER: *A Behaviour-Based Navigation System for an Autonomous Indoor Blimp*. Seiten 837–842, Heidelberg, Germany, September 2006. IFAC Symposium on Mechatronic Systems (IFAC).
- [MMLF10] L. MEJIAS, S. McNAMARA, J. LAI und J. FORD: *Vision-Based Detection and Tracking of Aerial Targets for UAV Collision Avoidance*. Seiten 87–92, Taipeh, Taiwan, Oktober 2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [MMW<sup>+</sup>07] O. MEISTER, R. MÖNIKES, J. WENDEL, N. FRIETSCH, C. SCHLAILE und G. TROMMER: *Development of a GPS/INS/MAG navigation system and waypoint navigator for a VTOL UAV*. Band 6561, Seite 65611D, Orlando, FL, USA, April 2007. SPIE Unmanned Systems Technology IX.
- [MNC09] D. G. MACHARET, A. A. NETO und M. F. M. CAMPOS: *On the Generation of Feasible Paths for Aerial Robots in Environments with Obstacles*. Seiten 3380–3385, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Mor97] S. J. MORRIS: *Design and flight test results for micro sized fixed wing and VTOL aircraft*. Georgia Institute of Technology, Atlanta, Georgia,, Februar 1997. International Conference on Emerging Technologies for Micro Air Vehicles.
- [Nan] *Nanotron Technologies*.  
URL: <http://www.nanotron.com>.
- [NKS<sup>+</sup>10] K. NONAMI, F. KENDOUL, S. SUZUKI, W. WANG und D. NAKAZAWA: *Autonomous Flying Robots Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Springer, 2010.
- [PH10a] T. PULS und A. HEIN: *3D-Trajectory Control for Quadrocopter*. Seiten 640–645, Taipeh, Taiwan, Oktober 2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [PH10b] T. PULS und A. HEIN: *Outdoor Position Estimation and Autonomous Landing Algorithm for Quadrocopter using a Wireless Sensor Network*. 2010.
- [PKKH09] T. PULS, M. KEMPER, R. KÜKE und A. HEIN: *GPS-based Position Control and Waypoint Navigation System for Quadrocopters*. Seiten 3374–3379, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

- [PMC] P. POUNDS, R. MAHONY und P. CORKE: *Modelling and Control of a Quad-Rotor Robot*. Auckland, New Zealand, November. Australasian Conference ob Robotics and Automation.
- [PMHR02] P. POUNDS, R. MAHONY, P. HYNES und J. ROBERTS: *Design of a Four-Rotor Aerial Robot*. Auckland, New Zealand, November 2002. Australasian Conference ob Robotics and Automation.
- [Pro] PROFESSOR J.GORDON LEISHMAN: *The Bréguet-Richet Quad-Rotor Helicopter of 1907*.
- [PSTS97] M. PELLETIER, A. SAHMOTO, C. TEDER und G. SALNTONGE: *Autonomous Navigation and Control Functions of the CL-327 VTOL UAV*. AGARD MSP Symposium on System Design Considerations for Unmanned Tactical Aircraft (UTA), Oktober 1997.
- [Pul10] T. PULS: *go!CART - Das UAV Programm der Metropolregion Bremen-Oldenburg*. In: T. LUHMANN und C. MÜLLER (Herausgeber): *Photogrammetrie-Laserscanning Optische 3D-Messtechnik, Beiträge der Oldenburger 3D-Tage 2010*, Oldenburg, Germany, Februar 2010. Institut für Angewandte Photogrammetrie und Geoinformatik (IAPG), Verlag Herbert Wichmann. ISBN: .
- [PWE<sup>+</sup>09] T. PULS, H. WINKELMANN, S. EILERS, M. BRUCKE und A. HEIN: *Interaction of Altitude Control and Waypoint Navigation of a 4 Rotor Helicopter*. In: T. KRÖGER und T. M.WAHL (Herausgeber): *Advances in Robotics Research*, Seiten 287–298, Braunschweig, Germany, Juni 2009. German Workshop on Robotics (GWR).
- [RGCF10] E. RONDON, L.-R. GARCIA-CARRILLO und I. FANTONI: *Vision-Based Altitude, Position and Speed Regulation of a Quadrotor Rotorcraft*. Seiten 628–633, Taipeh, Taiwan, Oktober 2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Rhe10] RHEINMETALL DEFENSE ELECTRONICS: 2010. Projekbeschreibung.
- [RK07] W. RÜTHER-KINDEL: *Flugmechanik - Grundlagen und stationäre Flugzustände*. Vorlesungsscript WS 2006/2007, 2007.  
URL: <http://www.tfh-wildau.de/1105/3.Semester/Flugmechanik/>.
- [RPHB07] A. ROTTMANN, C. PLAGEMANN, P. HILGERS und W. BURGARD: *Autonomous blimp control using model-free reinforcement learning in a continuous*

- state and action space.* Seiten 1895–1900, San Diego, California, USA, Oktober 2007. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Sch08] H.-W. SCHULZ: *Towards Vision-Based Autonomous Landing for Small Unmanned Aerial Vehicles: Image Processing Hardware Development*. Journal of Aerospace Computing, Information And Communication, 5, 2008.
- [SFS02] S. SARIPALLI, J. F. MONTGOMERY und G. S. SUKHATME: *Vision-based Autonomous Landing of an Unmanned Aerial Vehicle*. Seiten 2799–2804, Washington, DC, USA, 2002. IEEE International Conference on Robotics and Automation (ICRA).
- [SK08] SICILIANO und KHATIB (Herausgeber): *Handbook of Robotics*. Springer, Juli 2008.
- [SKPN09] A. SYARIL, F. KENDOUL, D. PERBRIANTI und K. NONAMI: *Visual Servoing of an Autonomous Micro Air Vehicle for Ground Object Tracking*. Seiten 5321–5326, St.Louis, Mo, USA, Oktober 2009. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [SPKR06] Y.-S. SUH, S.-K. PARK, H.-J. KANG und Y.-S. RO: *Attitude Estimation Adaptively Compensating External Acceleration*. JSME International Journal, 49:172–179, 2006.
- [Sul07a] F. . SULLIVAN: *Study Analysing the Current Activities in the Field of UAV - First Element: Status*. Technischer Bericht, European Commission, Enterprise and Industry Directorate-General, 2007.
- [Sul07b] F. . SULLIVAN: *Study Analysing the Current Activities in the Field of UAV - Second Element: Way forward*. Technischer Bericht, European Commission, Enterprise and Industry Directorate-General, 2007.
- [SWT04] C. SCHLAILE, J. WENDEL und G. F. TROMMER: *Stabilizing a Four-Rotor Helicopter using Computer Vision*. Braunschweig, Germany, Juli 2004. Proceedings of the 1st European Micro Air Vehicle Conference and Flight Competition.
- [tas10] taskit, 2010.  
URL: <http://www.taskit.de/>.
- [TBF05] S. THRUN, W. BURGARD und D. FOX (Herausgeber): *Probabilistic Robotics*. The Mit Press, September 2005.

- [TVH06] G. P. TOURNIER, M. VALENTI und J. P. HOW: *Estimation and Control of a Quadrotor Vehicle Using Monocular Vision and Moiré Patterns*. Keystone, Colorado, August 2006. AIAA Guidance, Navigation and Control Conference and Exhibit. AIAA 2006-6711.
- [ubl07] UBLOX AG: *Essentials of Satellite Navigation*, Doc ID: GPS-X-02007-C Auflage, April 2007.  
URL: [http://www.u-blox.com/images/downloads/Product\\\_Docs/GPS\\\_Compendium\%28GPS-X-02007\%29.pdf](http://www.u-blox.com/images/downloads/Product\_Docs/GPS\_Compendium\%28GPS-X-02007\%29.pdf).
- [ubl09] UBLOX AG: *GPS und GNSS: Grundlagen der Ortung und Navigation mit Satelliten*, Januar 2009.  
URL: [http://www.zogg-jm.ch/Dateien/Update\\\_Zogg\\\_Deutsche\\\_Version\\\_Jan\\\_09\\\_Version\\\_Z4x.pdf](http://www.zogg-jm.ch/Dateien/Update\_Zogg\_Deutsche\_Version\_Jan\_09\_Version\_Z4x.pdf).
- [Unb07] H. UNBEHAUEN: *Regelungstechnik II: Zustandsregelungen, digitale und nicht-lineare Regelsysteme*, Band 9. Vieweg+Teubner, März 2007.
- [Vie] *Backstepping - Ein Rekursives Entwurfsverfahren für den Nichtlinearen Regelungsentwurf*. Technische Universität Wien, Teil 1: Grundlagen und Einführung.
- [VN09] H. VOOS und B. NOURGHASSEMI: *Nonlinear Control of Stabilized Flight and Landing for Quadrotor UAVs*. Zielona Gora, Poland, November 2009. European Institute for Applied Research (IAR).
- [Voo09a] H. VOOS: *Nonlinear Control of a Quadrotor Micro-UAV using Feedback-Linearization*. Malaga, Spain, 2009. IEEE Int Conf on Mechatronics.
- [Voo09b] H. VOOS: *Nonlinear Landing Control for Quadrotor UAVs*. Karlsruhe, Germany, Dezember 2009. Autonome Mobile Systeme (AMS 2009).
- [VSB<sup>+</sup>03] P. VÖRSMANN, H.-W. SCHULZ, M. BUSCHMANN, T. KORDES und S. WINKLER: *Autonomous Navigation for Micro Airplanes, The Project „CAROLO“ - an Overview*. Berlin, Germany, Oktober 2003. 11th IAIN World Congress, Smart Navigation - Systems and Services.
- [VSB<sup>+</sup>04] P. VÖRSMANN, H.-W. SCHULZ, M. BUSCHMANN, T. KORDES und S. WINKLER: *Fully Autonomous Navigation of the Micro Airplane „Carolo“*. Cairo, Egypt, 2004. International Navigation Conference, Melaha.

- [WBV06] S. WINKLER, M. BUSCHMANN und P. VÖRSMANN: *An Integrated Navigation and Flight Control System for Small Unmanned Aircraft*. Portoroz, Slovenija, Dezember 2006. Proceedings of the 10th International Conference on Traffic Science ICTS 2006.
- [Wen07] D. J. WENDEL: *Integrierte Navigationssysteme*. Oldenbourg Wissenschaftsverlag GmbH, März 2007. ISBN: 3486581600.
- [WGLL07] K. WONG, J. GUERRERO, D. LARA und R. LOZANO: *Attitude Stabilization in Hover Flight of a Mini Tail-Sitter UAV with Variable Pitch Propeller*. San Diego, USA, Oktober 2007. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [WHJT05] S. L. WASLANDER, G. M. HOFFMANN, J. S. JANG und C. J. TOMLIN: *Multi-Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2005.
- [WMZ10a] K. E. WENZEL, A. MASSELLI und A. ZELL: *A Quadrocopter Hovering Above a Person Wearing a Modified Cap*. Braunschweig, Deutschland, Juli 2010. International Micro Air Vehicle Conference and Flight Competition.
- [WMZ10b] K. E. WENZEL, A. MASSELLI und A. ZELL: *Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle*. Journal of Intelligent and Robotic Systems, September 2010.
- [WRZ10] K. E. WENZEL, P. ROSSET und A. ZELL: *Low-Cost Visual Tracking of a Landing Place and Hovering Flight Control with a Microcontroller*. Journal of Intelligent and Robotic Systems, 57:297–311, Januar 2010. Kluwer Academic Publishers, Hingham, MA, USA.
- [YGP10] X. YANG, M. GARRATT und H. POTA: *Monotonous Trend Estimation of Deck Displacement for Automatic Landing of Rotorcraft UAVs*. Journal of Intelligent and Robotic Systems, September 2010.
- [YLW05] C. C. YANG, L. C. LAI und C. J. WU: *Time optimal control of a hovering quadrotor helicopter*. Seiten 295–300. IEEE International Conference on Systems and Signals (ICSS), 2005.
- [ZWS10] S. ZINGG, D. SCARAMUZZA, S. WEISS und R. SIEGWART: *MAV Navigation through Indoor Corridors Using Optical Flow*. Seiten 3361 – 3368, Anchorage,

AK, Mai 2010. IEEE International Conference on Robotics and Automation (ICRA).



# Abbildungsverzeichnis

2.1	Klassifizierung der Flugsysteme nach Flughöhe (Altitude) und maximalen Abfluggewicht (MTOW, Maximum Take Off Weight) [Rhe10] . . . . .	7
2.2	Quadrokopterkonfigurationen, links: die MD-200 von Microdrones [Mic10] und rechts: der Folcon 8 von Ascending Technologies [Asc10] . . . . .	10
2.3	Modell: System mit Kräften, Momenten und Koordinatensystemen [Kem07]	11
2.4	Reglerstruktur eines Quadroopters: Lageregler (siehe Abschnitt 2.4), Lokalisation (2.5), Positions-/Höhenregler (2.6.1 und 2.6.2), Kollisionsvermeidung (2.7.1), Pfadplanung (2.7.2), Automatisches Landen (2.7.3) . . . . .	12
2.5	Landmarken: symmetrische Landmarke (links) - asymmetrische Landmarke (rechts) . . . . .	28
4.1	Guard M . . . . .	35
4.2	Guard S . . . . .	36
4.3	Foto der Avionik mit Modulen: Der GPS-Sensor und das Funkmodul befinden sich auf dem FMB und sind im Bild nicht zu sehen. . . . .	37
4.4	Avionik . . . . .	38
4.5	Signalverzögerung bei GPS-Signalen bei linearer Bewegung . . . . .	40
4.6	Laufzeitverzögerung: Signal nach Filterung . . . . .	41
4.7	Laufzeitverzögerung: Signal nach Filterung (Ausschnitt) . . . . .	42
5.1	Kamerasysteme: Kamera (Nadir) trackt Zielmarke am Boden (links) - Kamera (Zenit) trackt UAS am Himmel (rechts) . . . . .	44
5.2	Schema: Bakensystem mit vier passiven Sensorknoten am Boden und einem aktiven unter dem UAS. . . . .	45
5.3	Guard S mit Lokalisationssystem (links) - Funkknoten mit Ultraschallempfängern (rechts) . . . . .	46

5.4	Messaufbau: Quadrokopter im Zentrum des Vierecks . . . . .	47
5.5	Messdaten der vier Sensorknoten: Funklaufzeit (links) - Ultraschall (rechts)	47
5.6	Messaufbau: Quadrokopter an Rückposition (links) - Quadrokopter an Eckposition (rechts) . . . . .	48
5.7	Abstandsdaten vom Knoten 1: UAS an Rückposition und an Eckposition .	48
5.8	Lokalisation mit Funk und Ultraschall: UAS im Zentrum . . . . .	51
5.9	Lokalisation mit Funk und Ultraschall: UAS in Eckposition . . . . .	51
5.10	Höhenbestimmung mit Funk und Ultraschall . . . . .	51
6.1	Übersicht über die entwickelten Regler: Lageregler (siehe Abschnitt 6.1), 2D-Wegpunktregler (6.3), Trajektorienregler (6.4), Objekt verfolgen (6.6), Automatischer Start (6.7), Automatische Landung (6.8) . . . . .	53
6.2	Höhenregler . . . . .	58
6.3	2D-Wegpunktregler . . . . .	63
6.4	Sollstandlinienregelung . . . . .	65
6.5	Verschiedene Trajektorien - links: linear und rechts: Kreis . . . . .	79
6.6	Verschiedene Geschwindigkeitsverläufe / Rampen . . . . .	80
6.7	Schematische Darstellung der Landung inkl. der verwendeten Messverfahren und der daraus resultierenden Phasen . . . . .	84
6.8	Zustandsdiagramm des Landeargorithmus . . . . .	85
6.9	Messfehler bei sich bewegender Messvorrichtung . . . . .	88
7.1	Simulation: Lageregelung inklusive I-Anteil . . . . .	89
7.2	Simulation: Höhenregler . . . . .	90
7.3	Simulation: Positionshaltung mit Wind . . . . .	91
7.4	Simulation: Positionshaltung mit Wind und unterschiedlichen Beschleunigungsdämpfungen . . . . .	91
7.5	Simulation: Streckenflug . . . . .	93
7.6	Simulation: Sollstandlinienregelung . . . . .	93
7.7	Simulation: simultane Höhen- und Positionsänderung . . . . .	94
7.8	Simulation: Abflug eines Vierecks . . . . .	95

7.9	Simulation: 100m-Flug mit 6s-cos(x)-Rampe, $v_{cr} = 5 \frac{m}{s}$	96
7.10	Simulation: Positionsfehler während des 100m-Fluges aus Abb. 7.9 mit den verschiedenen Rampenfunktionen aus Abb. 6.6.	97
7.11	Simulation: korrespondierende berechnete Winkel $\theta_d$ von Abb. 6.6	98
7.12	Simulation: Positionsabweichung während des 100m-Fluges mit unterschiedlichen Rampenzeiten $t_{alter}$	98
7.13	Simulation: Geozentrischer Kreis mit einem Radius von 25m, 18s-cos(x)-Rampe, $v_{cr} = 5 \frac{m}{s}$	99
7.14	Simulation: Positionsabweichung während des Kreisfluges	99
7.15	Tangentialvektor $v_{cr}^T$ und Sekantenvektor $v_{cr}^T$	100
7.16	Simulation: Abweichungen während des Kreisfluges mit Tangential- und Sekantenvektor	100
7.17	Simulation: Höhenänderung mit $x^2$ -Funktion und verschiedenen Rampenzeiten, $v_{cr} = 3 \frac{m}{s}$	101
7.18	Simulation: Fahrzeugverfolgung	102
7.19	Simulation: automatischer Start	103
7.20	Simulation: automatische Landung	104
7.21	Simulation: automatische Landung auf Fahrzeug, 3D-Trajektorie des UASs: 1) GPS-Modus 2) Funkmessung Modus 3) USS-Modus 4) Endanflug Modus	104
7.22	Simulation: automatische Landung auf Fahrzeug	105
8.1	Experiment: Höhenregler	108
8.2	Experiment: Streckenflug	108
8.3	Experiment: simultane Höhen- und Positionsänderung	109
8.4	Experiment: Abflug eines Vierecks	110
8.5	Experiment: lineare 3D Trajektorie, 6s- $x^2$ -Rampe, $v_{cr} = 3 \frac{m}{s}$	111
8.6	Experiment: Geozentrischer Kreis mit 30m-Radius und 6s- $x^2$ -Rampe, $v_{cr} = 3 \frac{m}{s}$	112
8.7	Experiment: Positionsabweichung während des geozentrischer Kreises	112
8.8	Experiment: Verfolgen eines Fahrzeuges	113
8.9	Versuchsaufbau	114

8.10 Experiment: Trajektorie während Positions haltung . . . . .	115
8.11 Experiment: Abstand zur Mitte während Positions haltung . . . . .	115
8.12 Experiment: Landeanflug . . . . .	116
8.13 Experiment: Landungen . . . . .	117
8.14 Experiment: Landeanflug mit Endanflug-Abbruch . . . . .	118
8.15 Fahrzeug mit Landevorrichtung . . . . .	118
8.16 Experiment: Landeablauf auf fahrendem Fahrzeug . . . . .	119
8.17 Experiment: Landungen auf fahrendem Fahrzeug . . . . .	120

# Tabellenverzeichnis

2.1	Übersicht Lokalisationsverfahren . . . . .	20
6.1	Ziegler und Nichols Einstellregeln [Kem02] . . . . .	81
6.2	Durch Experimente optimierte Reglerparameter: . . . . .	82
7.1	Durchschnittliche und maximale Positionsabweichung über die Flugzeit nach Störeinfluss: 1) keine Beschleunigungsdämpfung vorhanden 2) mit Beschleunigungsdämpfung 3) nur Beschleunigungen werden gedämpft, das Abbremsen nicht . . . . .	92
7.2	Durchschnittliche Positionsabweichung während des 100m-Fluges mit unterschiedlichen Funktionen . . . . .	97
8.1	Über die Flugzeit durchschnittliche Positions- und Geschwindigkeitsabweichungen von der linearen 3D-Trajektorie . . . . .	112



# Variablenverzeichnis

$m$	Masse des UAS : $kg$
$l$	Abstand Rotor zum Mittelpunkt des UAS : $m$
$I$	Trägheitstensor : $kg \cdot m^2$
$u_i$	Eingaben
$C_{Di}$	Proportionalitätsfaktoren der Widerstände : Translation $kg/m$ , Rotation $kg \cdot m$
$F_i$	Schub : $N$
$M_i$	Drehmoment : $Nm$
$r_{pol}$	Erdradius am Pol : $m$
$r_{\ddot{a}q}$	Erdradius am Äquator : $m$
$K$	Kalman-Verstärkung
$\delta$	halber Basisabstand der Sensorknoten : $m$
$\delta_i$	Abstand UAS zum Sensorknoten $i$ : $m$
$WP$	Wegpunkt
$lat$	Latitude
$lon$	Longitude
$d$	Distanz zum Ziel : $m$
$\omega$	Peilung zum Ziel : $rad$
$\omega_f$	durch Sollstandlinie korrigierter Peilwinkel : $rad$
$\omega_{psi}$	korrigierter Peilwinkel $\omega_f$ inkl. Gierwinkelkorrektur des UAS : $rad$
$POT$	point on target
$d_0$	Distanz zum Ziel, ab der keine Sollstandlinienregelung mehr stattfindet. : $m$
$m_b$	Steigung der Sollstandlinienfunktion
$b_b$	Offset der Sollstandlinienfunktion
$CTE$	„cross-track“-Fehler : $m$
$ATE$	„along-track“-Fehler : $m$
$CTSD$	„cross-track“-Geschwindigkeitsdämpfung
$ATSD$	„along-track“-Geschwindigkeitsdämpfung
$AD$	Beschleunigungsdämpfung
$S$	Zustandsvektor des UAS

$p$	Position : $m$
$x$	x-Position : $m$
$y$	y-Position : $m$
$z$	z-Position : $m$
$v$	Geschwindigkeit : $m/s$
$a$	Beschleunigung : $m/s^2$
$\phi$	Rollwinkel : $rad$
$\theta$	Nickwinkel : $rad$
$\psi$	Gierwinkel : $rad$
$\Omega$	Drehgeschwindigkeit : $rad/s$
$e$	Regeldifferenz
$k$	Regelparameter
$y$	Stellwert
$E$	Arbeitspunkt
$V$	Lyapunov-Funktion
$\xi$	Funktionsvariablen der Lyapunov-Funktion
$d$	Index, Sollwert (desired)
$c$	Index, Istwert (current)
$f$	Index für köperfestes Koordinatensystem
$g$	Index für geodätisches Koordinatensystem
$h$	Index für Regeldifferenz, Höhe
$p$	Index für Regelparameter, Proportionalanteil
$i$	Index für Regelparameter, Integralanteil
$ih$	Index für Regelparameter, Integralanteil höherer Wert
$il$	Index für Regelparameter, Integralanteil niedrigerer Wert
$d$	Index für Regelparameter, Geschwindigkeit
$a$	Index für Regelparameter, Beschleunigung
$sr$	Index für Regelparameter, Sinkrate
$cross$	Index für Regelparameter, „cross-track“-Proportionalanteil

# Abkürzungsverzeichnis

<b>DDD</b> .....	Dull Dirty Dangerous
<b>FCS</b> .....	Flight Control System
<b>GNSS</b> .....	Global Navigation Satellite System
<b>GPS</b> .....	Global Positioning System
<b>HALE</b> .....	High Altitude Long Endurance
<b>IMU</b> .....	Inertial Measurement Unit
<b>INS</b> .....	Inertial Navigation Systems
<b>MALE</b> .....	Medium Altitude Long Endurance
<b>MCL</b> .....	Monte-Carlo-Localization
<b>MUAS</b> .....	Mini Unmanned Aerial System
<b>MUAV</b> .....	Mini Unmanned Aerial Vehicle
<b>SBAS</b> .....	Satellite-Based Augmentation System
<b>SLAM</b> .....	Simultaneous Localization and Mapping
<b>UAS</b> .....	Unmanned Aerial System
<b>UAV</b> .....	Unmanned Aerial Vehicles
<b>VTOL</b> .....	Vertical Takeoff or Landing
<b>WSN</b> .....	Wireless Sensor Network