

Master-Thesis

Modellbildung und Entwurf einer Positionsregelung eines UAVs zur Unterstützung von Rettungskräften

vorgelegt von

B. Eng. Marius Beul

Master-Thesis

Modellbildung und Entwurf einer Positionsregelung eines UAVs zur Unterstützung von Rettungskräften

vorgelegt von

B. Eng. Marius Beul

Wolfstraße 22

53111 Bonn

majus1985@gmail.com

Matr.-Nr. 11084710

Fakultät für Informations-, Medien- und Elektrotechnik
Studiengang Elektrotechnik (M. Sc.)

Betreuer: Dipl.-Inform. Rainer Worst

Prüfer: Prof. Dr.-Ing. Jens Onno Krahl

31. August 2013

Danksagung

Ich möchte Herrn Dipl.-Inform. Rainer Worst für die vielfältige Unterstützung und intensive Betreuung meiner Arbeit danken. Auch Prof. Dr.-Ing. Jens Onno Krahe danke ich für die umfassende Betreuung dieser Arbeit.

Ich bedanke mich bei den Studenten und Mitarbeitern Josef Börding, Finley David Daniel, M. Sc. Thorsten Linder, Dr. Zahid Riaz und Prof. Dr.-Ing. Hartmut Surmann des Fraunhofer-Instituts für Intelligente Analyse- und Informationssysteme IAIS für die freundliche Arbeitsatmosphäre, für Rat und Tat und fürs Korrekturlesen dieser Arbeit. Weiterhin bedanke ich mich bei Dipl.-Ing. Jochen Winzer für die hilfreiche Unterstützung in der Werkstatt während des gesamten Projekts. Auch bei Herrn Prof. Dr. Sven Behnke der Rheinischen Friedrich-Wilhelms-Universität Bonn bedanke ich mich für die Nutzung der Infrastruktur und für die kreativen Anregungen.

Meiner Freundin Alexandra Plawky möchte ich für den seelischen Beistand während der Master-Thesis und für das Korrekturlesen der Arbeit danken. Mein besonderer Dank gilt meinen Eltern Hannelore Beul und Christoph Schulz-Beul, ohne deren Unterstützung mein Studium nicht möglich gewesen wäre.

Kurzfassung

In Einsätzen sind Rettungskräfte starker körperlicher und geistiger Belastung ausgesetzt. Im EU-Projekt NIFTi (Natural Human-Robot Cooperation in Dynamic Environments) wird unter anderem ein Unmanned Aerial Vehicle (UAV) zur Unterstützung von Rettungskräften entwickelt. Es dient als mobiler Sensorträger und soll im engen Zusammenspiel mit den Rettungskräften selbstständig Situationen erkennen und so die Bergung von Opfern verbessern. Um tatsächlich eine Hilfe zu sein, muss es über ein möglichst großes Maß an Autonomie verfügen.

In dieser Arbeit wird eine Positionsregelung entwickelt, die es dem UAV ermöglicht, Wegpunkte anzusteuern und dort zu verharren. Als Ansatz für die Entwicklung der Regelung dient ein grey-box-Modell der Kinematik des UAV. Diese wird zunächst qualitativ und anschließend quantitativ modelliert. Auf Basis des Modells wird ein Regelalgorithmus entwickelt und parametrisiert. Es zeigt sich, dass sich ein zweistufiger, nichtlinearer Kaskadenregler optimal für die Regelung eignet. Dieser wird auf dem UAV implementiert und zunächst in einer Simulation und anschließend in der Realität evaluiert.

Die Evaluation beweist, dass der Algorithmus in der Lage ist, eine optimale Positionsregelung des UAV durchzuführen. Seine volle Leistungsfähigkeit (Anregelzeit, Regelabweichung etc.) zeigt er in der Simulation, da hier ein gutes (genaues, schnelles) Feedbacksignal zur Verfügung steht. In der Realität kommt GPS als Feedbacksystem zum Einsatz, was die Dynamik und die Genauigkeit der Regelung einschränkt.

Abstract

Successful execution of rescue operations demand quick response from the fire-fighters which may cause physical and psychological stress on them during emergency services. In order to facilitate them to perform their task efficiently in the context of the EU-project NIFTi (Natural Human-Robot Cooperation in Dynamic Environments) an Unmanned Aerial Vehicle (UAV) is developed to support such situations.

The UAV serves as a mobile sensor platform and works in cooperation with the humans involved. In human-centric rescue operations it autonomously gains situation awareness and can improve the rescue of victims. For these tasks large-scale autonomy is necessary.

In this work a closed loop position controller for the UAV is developed. It enables the UAV to autonomously approach certain waypoints. The approach proposed in this work initially models the kinematics of the UAV and fits a custom control algorithm. A qualitative model is constructed and later experimentally identified to match the life-like behavior. It is shown that a nonlinear cascaded controller delivers optimal results. The control algorithm is implemented on the UAV and is evaluated in simulation and in real-life.

The evaluation proves that the algorithm is capable of controlling the UAV. In simulation the full performance (rise time, deviation, etc.) is achieved. The approach has also been successfully applied to actual UAV flight. Due to lack of accuracy and slower update rate of the GPS, slightly less results are obtained as compared to simulation.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Das Projekt NIFTi	1
1.2	Motivation für diese Arbeit	2
1.3	Gliederung der Arbeit	3
2	Stand der Technik	4
2.1	Komponenten des UAV	4
2.2	Positionsregelungen	8
2.2.1	Stand der Technik	8
2.2.2	Stand der Wissenschaft	10
2.2.3	Fazit	10
3	Modellbildung	11
3.1	Eindimensionales Modell (zwei Freiheitsgrade)	11
3.1.1	Herleitung des Modells	12
3.1.2	Annahmen zur Vereinfachung des Modells	15
3.1.3	Vereinfachung des Modells	16
3.1.4	Modellparameter	17
3.2	Zweidimensionales Modell (vier Freiheitsgrade)	21
3.3	Dreidimensionales Modell (sechs Freiheitsgrade)	22
4	Reglerentwurf	25
4.1	Annahmen zur Reglerentwicklung	25
4.1.1	Unterlagerte Lageregelung	25
4.1.2	Pseudokontinuierliche Regelung	25
4.2	Analyse des dreidimensionalen Modells	26
4.3	Reglerstruktur	28
4.4	Reglerparametrierung	30
4.5	Störverhalten	36
4.5.1	Störung durch GPS	36

4.5.2	Störung durch Wind	38
5	Implementierung	41
5.1	Implementierung in der Simulation	42
5.1.1	Implementierung in ROS	42
5.1.2	RVIZ	45
5.2	Implementierung in der Realität	46
5.2.1	Implementierung der Hardware	46
5.2.2	Implementierung der Software	48
6	Evaluation	49
6.1	Modell	49
6.2	Simulation	51
6.2.1	Versuchsaufbau	52
6.2.2	Durchführung	53
6.2.3	Ergebnisse	55
6.3	Realität	56
6.3.1	Versuchsaufbau	56
6.3.2	Durchführung	56
6.3.3	Ergebnisse	62
7	Zusammenfassung und Ausblick	63
7.1	Zusammenfassung	63
7.2	Ausblick	64
8	Anhang	67

Abbildungsverzeichnis

1.1	Logo NIFTi-Projekt [NIF13]	1
1.2	Vorgehensweise	3
2.1	NIFTi-UAV	5
2.2	Informationsfluss vor der Implementierung	6
2.3	PX4 Positionsregelung [PX413]	9
3.1	Ansatz für das eindimensionale Modell	11
3.2	Eindimensionales Ersatzmodell der Regelstrecke	14
3.3	Vereinfachtes eindimensionales Ersatzmodell der Regelstrecke	16
3.4	Zusammengefasstes eindimensionales Ersatzmodell der Regelstrecke	17
3.5	Ersatzmodell der Regelstrecke mit starker Reibung	17
3.6	Ersatzmodell der Regelstrecke mit schwacher Reibung	18
3.7	Versuchsaufbau zur Parameterschätzung	19
3.8	Simulinkmodell der Regelstrecke mit schwacher Reibung	20
3.9	Versuch zur Parameterschätzung	21
3.10	Ansatz für das zweidimensionale Modell	22
3.11	Zweidimensionales Ersatzmodell der Regelstrecke	22
3.12	Ansatz für das dreidimensionale Modell	23
3.13	Dreidimensionales Ersatzmodell der Regelstrecke	24
4.1	Überblick der Struktur der Regelung	27
4.2	Kaskadenregelung	29
4.3	Überschwingen bei zu starkem $K_{p,aussen}$ (Position)	31
4.4	Überschwingen bei zu starkem $K_{p,aussen}$ (Geschwindigkeit)	32
4.5	Sollgeschwindigkeit bei Positionsfehler	33
4.6	Betrag Sollgeschwindigkeit bei Positionsfehler	34
4.7	Parametrierung am Fadenpendel	35
4.8	Rauschen des GPS	37
4.9	Blockschaltbild Störung durch Wind	39

5.1	Vorgehensweise der Implementierung	41
5.2	Übersicht Knoten und Themen	43
5.3	Visualisierung in RVIZ	46
5.4	Informationsfluss nach der Implementierung	47
6.1	Vorgehensweise der Evaluation	49
6.2	Simulinkmodell zur Evaluation des Modells	50
6.3	Trajektorie des realen UAV	50
6.4	Winkel des realen UAV und des Modells	51
6.5	Versuchsaufbau	52
6.6	Simulierter Testflug mit neuer Positionsregelung ($X, \Theta_{max} = \pm 1^\circ$) . . .	53
6.7	Simulierter Testflug mit neuer Positionsregelung ($X, \Theta_{max} = \pm 3^\circ$) . . .	54
6.8	Simulierter Testflug mit neuer Positionsregelung (XY)	55
6.9	Testflug mit neuer Positionsregelung ($X, \Theta_{max} = \pm 2^\circ$)	57
6.10	Testflug mit neuer Positionsregelung ($X, \Theta_{max} = \pm 3^\circ$)	57
6.11	Testflug mit neuer Positionsregelung ($X, \Theta_{max} = \pm 5^\circ$)	58
6.12	Testflug mit alter Positionsregelung (X)	59
6.13	Testflug mit neuer Positionsregelung (XY, $\Theta_{max} = \pm 2^\circ$)	60
6.14	Testflug mit neuer Positionsregelung (XY, $\Theta_{max} = \pm 3^\circ$)	60
6.15	Testflug mit neuer Positionsregelung (XY, $\Theta_{max} = \pm 5^\circ$)	61
6.16	Testflug mit alter Positionsregelung (XY)	61

Tabellenverzeichnis

5.1	Übersicht der Themen	45
5.2	Auswahl an Parser Nachrichten	48
6.1	Eigenschaften der Regelung (Simulation)	55
6.2	Randbedingungen der realen Evaluation	56
6.3	Eigenschaften der Regelung (Realität)	62

Abkürzungsverzeichnis

GCS	Ground Control Station
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LIDAR	Light Detection and Ranging
NIFTi	Natural Human-Robot Cooperation in Dynamic Environments
ROS	Robot Operating System
RVIZ	ROS Visualization Environment
UAV	Unmanned Aerial Vehicle

1 Einleitung

1.1 Das Projekt NIFTi

Das Projekt NIFTi (Natural Human-Robot Cooperation in Dynamic Environments) wird auf dem offiziellen Internetauftritt wie folgt beschrieben: „Ereignisse wie Verkehrsunfälle in Tunneln, unbeabsichtigte Austritte von chemischen Substanzen oder Gebäudeeinstürze machen die Unfallgebiete zu unübersichtlichen Gefahrenzonen. Damit Rettungskräfte sich künftig gefahrloser ein Lagebild machen können, sollen sie Hilfe von Robotern erhalten, die als stellvertretende Werkzeuge für Menschen eingesetzt werden. Dies ist das Ziel des EU-Forschungsprojektes „Natural Human-Robot Interaction in Dynamic Environments“ (NIFTi)“ [Sur13].

Ziel ist es, Roboter nicht nur als „verlängerten Arm“ des Menschen zu benutzen, sondern sie in den Prozess als Partner des Menschen zu integrieren. Weiterführende Informationen zum sozio-technischem System Design sind in [KJK⁺12] zu finden.

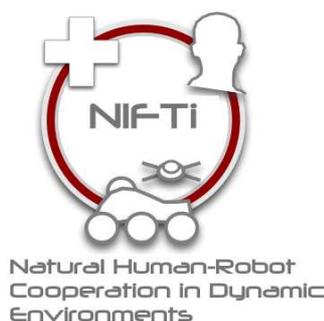


Abbildung 1.1: Logo NIFTi-Projekt [NIF13]

Mit einer Fördersumme von insgesamt 6.610.986 € unterstützt die EU die neun Projektpartner im siebten Rahmenprogramm; contract number FP7 ICT 247870, Cognitive Systems and Robotics. Unter der Leitung des deutschen Forschungszentrums für Künstliche Intelligenz DFKI kooperieren die folgenden Projektpartner:

- Deutsches Forschungszentrum für Künstliche Intelligenz DFKI (Deutschland)
- Niederländische Organisation für Angewandte Naturwissenschaftliche Forschung TNO (Niederlande)
- Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS (Deutschland)
- BlueBotics (Schweiz)
- ETH Zürich (Schweiz)
- Tschechische Technische Universität Prag (Tschechische Republik)
- Universität La Sapienza (Italien)
- Feuerwehr Dortmund (Deutschland)
- Innenministerium Italien (Italien)

Das Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS ist innerhalb des Projektes für die Entwicklung eines Unmanned Aerial Vehicle (UAV) verantwortlich, welches die Rettungskräfte im Verbund mit anderen Robotern in Gefahrensituationen selbstständig unterstützt.

1.2 Motivation für diese Arbeit

Im Feldversuch [KTL⁺12] zeigt sich, dass die Fernsteuerung des UAV zu einem „cognitive overload“ (mentale Überlastung) des Piloten führen kann. Stress und die Filterung relevanter Informationen in unübersichtlichen Situationen führt zu Überbeanspruchung. Dies schränkt den Piloten in seiner Leistungsfähigkeit ein und kann zu Fehlentscheidungen führen. Weiterhin bedarf es speziell ausgebildeter Piloten, um das UAV sicher zu steuern. Aus diesem Grund steht die Entlastung des Piloten im Fokus weiterer Arbeiten am UAV.

Durch Assistenzsysteme soll der Pilot in der Lage sein, insbesondere „Low-Level“-Aufgaben an den Roboter abzugeben. Hierdurch werden die Belastung des Piloten und somit Fehlentscheidungen unter Stress signifikant reduziert. Stattdessen wird die Überwachung und die Planung von Aufgaben mit höherem Abstraktionslevel zunehmend in den Aufgabenbereich des Piloten fallen. Weiterhin soll hierdurch die Steuerung des UAV auch ungeübten Piloten ermöglicht werden.

Vorversuche haben ergeben, dass insbesondere das genaue Anfliegen eines Punktes eine hohe Aufmerksamkeit des Piloten erfordert. Ständiges Korrigieren der Fluglage des UAV sind hierzu nötig. Auch die manuelle Korrektur der Fluglage beim Halten eines Punktes erfordern großes Geschick. Aus diesem Grund wird in dieser Arbeit ein Re-

gelgorithmus entwickelt, welcher die Positionsregelung des UAV übernimmt. Hierbei soll das UAV autonom bzw. semiautonom Wegpunkte im Raum anfliegen und halten können. Die Vorgabe dieser Wegpunkte erfolgt entweder durch den Piloten oder durch einen Algorithmus. In dieser Arbeit kommt GPS (Global Positioning System) als Sensorsystem zum Einsatz. Die Erweiterung auf ein anderes Feedbacksystem oder auf ein hybrides Positionsfeedback mehrerer Sensoren ist jedoch prinzipiell möglich.

Mit diesem Assistenzsystem ist der Pilot in der Lage, einen Teil seiner Arbeit abzugeben und so seine Belastung zu reduzieren. Das Fliegen des UAV ist auch ungeübten Piloten möglich.

1.3 Gliederung der Arbeit

Während Kapitel 1 die Motivation und Aufgabenstellung definiert, gibt Kapitel 2 einen Überblick über den Stand des Projektes (vor dieser Arbeit).

In Kapitel 3 wird ein Modell der Regelstrecke (des UAV) entwickelt, auf dessen Grundlage in Kapitel 4 ein Regelalgorithmus entworfen wird. Die Implementierung dieses Algorithmus' in das Gesamtsystem wird in Kapitel 5 dokumentiert. Anschließend wird die Funktion des Systems in Kapitel 6 evaluiert. Abschließend fasst Kapitel 7 die Ergebnisse dieser Arbeit zusammen und zeigt im Ausblick weitere Optimierungspotenziale auf. Abbildung 1.2 visualisiert die Vorgehensweise.

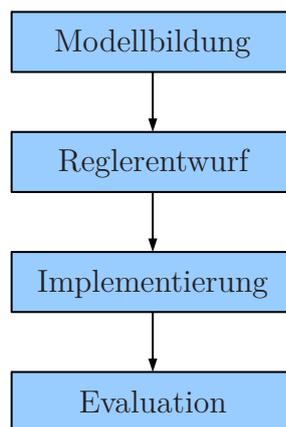


Abbildung 1.2: Vorgehensweise

2 Stand der Technik

Im Folgenden wird ein Überblick über den Projektstatus vor dem Erstellen dieser Arbeit gegeben. Der Stand der Technik der verwendeten Geräte und Methoden wird erläutert. Hierzu gibt Abschnitt 2.1 eine Einführung und einen Überblick über die Komponenten des UAV. Die Vernetzung der sich auf dem UAV befindlichen Hardware und Software wird erklärt. Auch die Verbindung/Schnittstelle zu weiteren Komponenten wird erläutert.

Abschnitt 2.2 gibt einen Überblick über vergleichbare Arbeiten aus Industrie und Wissenschaft. Es werden Ansätze für Positionsregelungen vorgestellt und in Hinblick auf die Zielsetzung dieser Arbeit bewertet. Insbesondere die Vor- und Nachteile der bisher implementierten Positionsregelung werden erläutert.

2.1 Komponenten des UAV

Zu Beginn dieser Arbeit ist die Hardware des UAV fertiggestellt. Abbildung 2.1 zeigt das flugbereite UAV. „Low-Level“-Software, die zum Fliegen notwendig ist (Motorregelung, Lageregelung, Empfängerwertung etc.), ist implementiert und getestet. Ein ferngesteuerter Flug ist somit möglich. Gleichzeitig findet eine Aufnahme und Verarbeitung der Sensordaten statt.



Abbildung 2.1: NIFTi-UAV

Abbildung 2.2 zeigt ein Blockschaltbild, welches den Informationsfluss verdeutlicht, wie er vor dieser Arbeit abläuft. Dabei zeigt der Rahmen die Systemgrenze des UAV zur Umgebung an. Blöcke symbolisieren Systeme, welche über die Pfeile uni- bzw. bidirektional kommunizieren können. Beschriftungen zeigen den Kommunikationsbus des Datenaustauschs an. Die Darstellung ist nicht vollständig, zeigt jedoch alle für diese Arbeit relevanten Informationen.

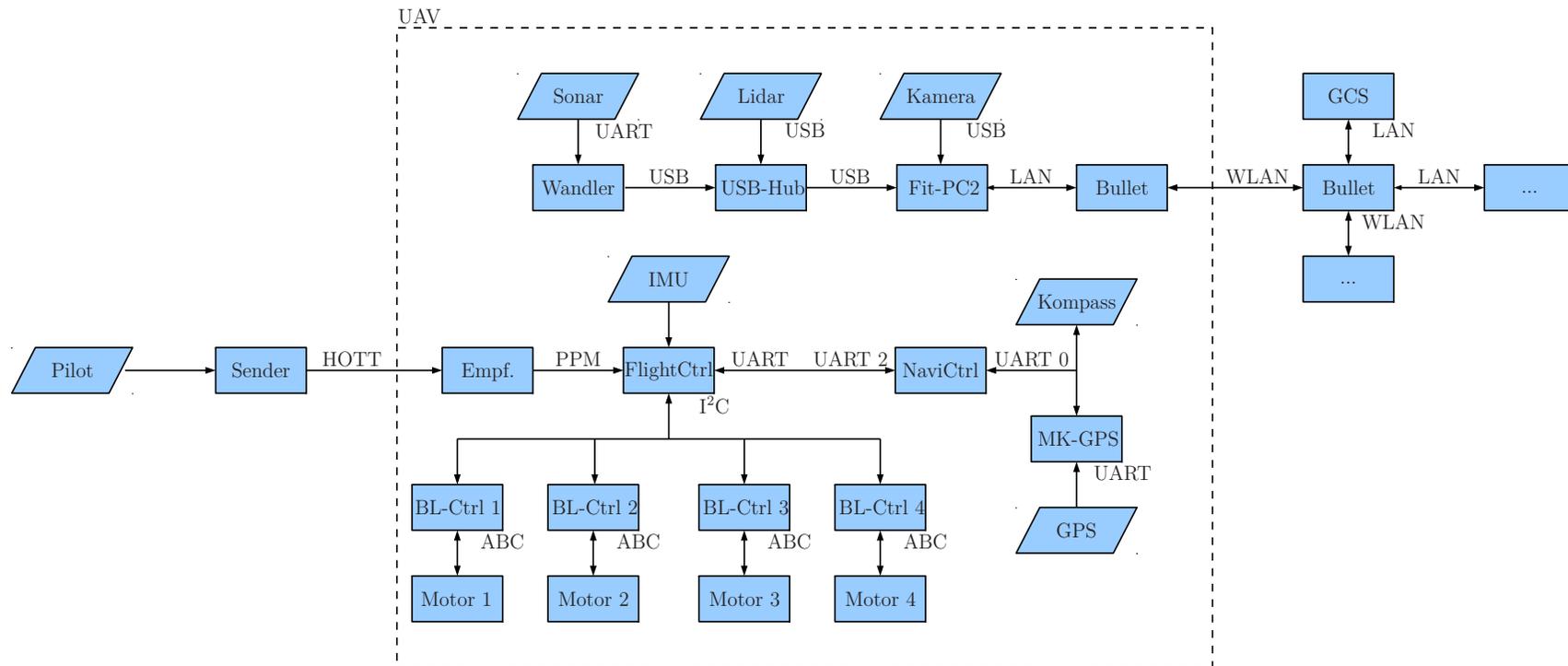


Abbildung 2.2: Informationsfluss vor der Implementierung

Auf dem UAV existieren zwei voneinander getrennte Systeme:

Zum einen gibt es ein „Low-Level“-System (unten). Mit dem Hauptrechner FlightCtrl bietet es eine (Funk-)Schnittstelle zum Piloten und kann über die Treiber BL-Ctrl die vier bürstenlosen Gleichstrommotoren ansteuern. Auf der FlightCtrl ist eine Inertial Measurement Unit (IMU) angebracht, welche die Lage des UAV im Raum misst. Mit diesem Feedback findet hier die Lageregelung des UAV statt. Die Software der FlightCtrl ist offen und modifizierbar. Weiterhin gibt es einen seriellen Kanal, über den der Koprozessor NaviCtrl auf Grundlage seiner Sensoren GPS¹ und Kompass eine übergeordnete Orientierungs- und Positionsregelung für das UAV durchführen kann. Ob diese übergeordnete Regelung stattfindet oder nicht entscheidet der Pilot mit einem Signal am Sender. Ist die Regelung aktiv, ist das UAV in der Lage, (mit einer proprietären Software definierte) Punkte im Raum anzufliegen und hierbei bestimmte Orientierungen einzunehmen. Auch ein automatisches Fliegen zum Startpunkt (z.B. bei einem Verbindungsausfall) ist möglich. Eine Bewertung dieser bereits vorhandenen Regelung findet im folgenden Abschnitt in Unterabschnitt 2.2.1 statt.

Im Mittelpunkt des zweiten Systems auf dem UAV (oben) steht ein Fit-PC². Dieser greift über USB auf eine Kamera, ein auf den Boden gerichtetes Sonar und einen 270° Laserabstandsmesser (Lidar) zu. Außerdem ist er über ein WLAN mit einer Bodenstation und weiteren Komponenten verbunden. Als bidirektionale Übertrager zwischen LAN und WLAN dienen Bullet M2-Netzwerkbrücken. Die Teilnehmer im LAN bieten verschiedene Dienste an. Die Software dieses „High-Level“-Systems ist eine Eigenentwicklung. Sie liegt als Quellcode vor und ist modifizierbar. Zur Integration der Funktionen der Teilnehmer im Netzwerk dient ROS (Robot Operating System), welches im Folgenden beschrieben wird.

ROS ist ein Framework, welches unter anderem für die Steuerung, Regelung und Kommunikation in NIFTi eingesetzt wird. Die offizielle Webseite beschreibt ROS folgendermaßen: „ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more.“ [ROS13].

Programmteile sitzen bei ROS in sogenannten Knoten (Nodes). Knoten können untereinander über Nachrichten (Messages) kommunizieren. Jede Nachricht gehört zu einem Thema (Topic). ROS arbeitet hier nach dem Publisher-Subscriber Entwurfsmuster: Ein Knoten kann Nachrichten zu Themen veröffentlichen (publish) und Themen abonnieren (subscribe). Durch diese Architektur entsteht ein u.U. komplexes Netzwerk von Knoten, welche untereinander Daten austauschen. Die Steuerung dieses Daten-

¹u-blox LEA-6S, 5 Hz, Genauigkeit 2,5 m [u-b11]

²Single-Board PC, Intel-Atom 1,6 GHz, 1 GB RAM, 6xUSB

austauschs übernimmt ROS.

Auf Systemebene liegen Knoten, welche für die Interaktion mit dem Benutzer notwendig sind, auf der Ground Control Station (GCS). Knoten zur Sensorsteuerung, zum Beispiel der Knoten, welcher den Treiber für das Sonar bereitstellt, liegen auf dem UAV. Weitere Knoten können auf anderen Computern, Robotern etc. im Netzwerk liegen. Weiterführende Informationen zum Hardware- und Softwaredesign sind in [TWW10] zu finden.

2.2 Positionsregelungen

Es existiert bereits eine große Anzahl an Positionsregelungen für UAV. Im Folgenden wird eine (nicht vollständige) Übersicht über die bekanntesten Systeme gegeben. Die Systeme werden auf Nutzbarkeit im NIFTi-UAV geprüft und bewertet.

2.2.1 Stand der Technik

Mikrokopter

Auf dem UAV ist bereits eine Positionsregelung der Marke Mikrokopter der HiSystems GmbH implementiert. Die Software der Regelung ist proprietär. Das heißt unter anderem, dass es nicht möglich ist, eigene Wegpunkte ohne die mitgelieferte Software zu programmieren bzw. diese Punkte dynamisch im Flug zu verändern. Auch eine Anpassung der Regelung, zum Beispiel zur Sensorfusion des GPS mit anderen Sensoren, ist nicht möglich, da der Algorithmus nach dem die Regelung stattfindet proprietär ist. Da dem Nutzer die Möglichkeit gegeben wird, die Parameter „GPS-P“, „GPS-I“, „GPS-D“, „GPS Verstärkung“ etc. einzustellen, lässt auf die Verwendung eines PID-Reglers schließen. Die Parameter sind Standardparameter, welche für eine große Anzahl an sehr unterschiedlichen UAV nutzbar sein sollen. Die Leistungsfähigkeit der Regelung ist deshalb infrage zu stellen.

PX4

Im PX4-Autopilot der ETH-Zürich wird der in Abbildung 2.3 dargestellte PID-Regler eingesetzt.

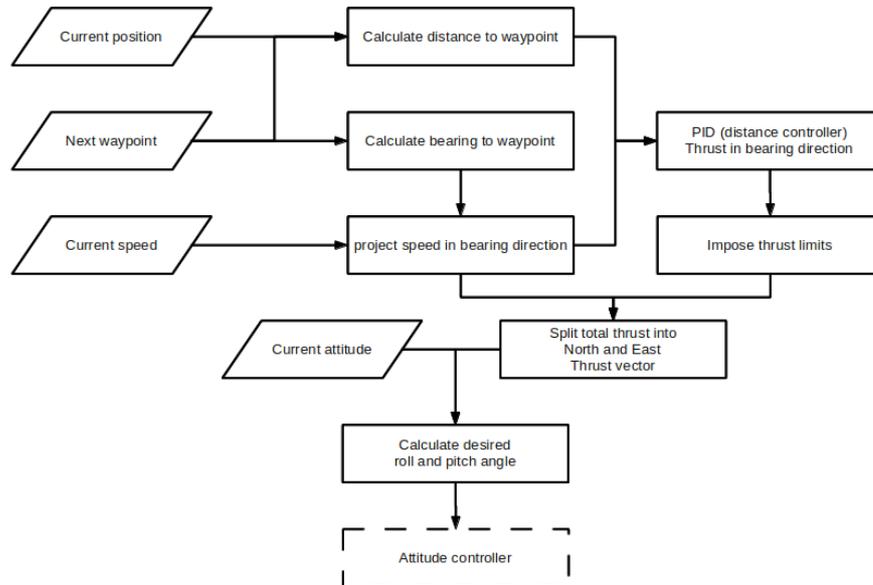


Abbildung 2.3: PX4 Positionsregelung [PX413]

Die Software der ETH-Zürich ist frei verfügbar. Die Parameter des Reglers müssen jedoch durch den Piloten iterativ erfolgen werden. Der Regler ist nicht speziell für GPS als Feedbacksystem ausgelegt.

OpenPilot CopterControl

Der Navigation Stack [Ope10] des OpenPilot CopterControl realisiert eine Positionsregelung über zwei kaskadierte PID-Regler. Die Firmware des Projekts ist offen und frei zugänglich. Alle sechs PID-Parameter müssen selber erfolgen werden. Es werden keine Standardparameter empfohlen. Auch dieser Regler ist nicht auf GPS als Feedbacksystem begrenzt.

2.2.2 Stand der Wissenschaft

In [PKKH09] wird ein PI-Regler beschrieben. Dieser wird um Korrekturterme erweitert, um das UAV auf einer direkten Verbindung vom Start- zum Zielpunkt zu führen. Je nach Nähe zum Wegpunkt wird der I-Anteil des Reglers zu- oder abgeschaltet. Die Regelung zeigt gute Ergebnisse, beruht aber auf einem vollständigen, mathematischen Modell des UAV. Durch das komplexe Zu- und Abschalten einzelner Regelungsteile werden Nichtlinearitäten und insbesondere Sprungstellen in der Regelung erzeugt. Dies kann Instabilitäten verursachen. Aus diesem Grund wird auf eine Implementierung der Regelung verzichtet.

In [BS07] wird ein komplexes Modell des UAV aus Differenzialgleichungen hergeleitet. Über Integral Backstepping werden lineare Regler für Position, Lage und Höhe des UAV abgeleitet. Das Modell wird nicht identifiziert. Der Positionsregler verwendet ein nicht näher definiertes „Vision System“ als Feedback. Regelparameter werden genannt, aber nicht hergeleitet. Auch diese Regelung wird aus oben genannten Gründen nicht implementiert.

Ein lineares Zustandsraummodell eines UAV wird in [PKJ12] entwickelt und identifiziert. In der Simulation wird ein linearer Zustandsregler implementiert und per Polvorgabe parametrisiert. Als Feedback kommt eine Kamera zum Einsatz. Die Regelung wird nur in der Simulation evaluiert. Der Regler zeigt hier gute Resultate, dies sagt jedoch über die Leistungsfähigkeit am realen UAV wenig aus.

Ein nichtlineares Modell eines UAV wird in [PPV12] erstellt. Dieses wird anschließend in einen linearen und einen nichtlinearen Teil aufgeteilt. Der lineare Teil wird per PID-, der nichtlineare Teil per Sliding-Mode geregelt. Es werden keine Parameter identifiziert und die Regelung wird nur in der Simulation evaluiert. Es wird kein Feedbacksystem beschrieben. Deshalb wird auf eine Modellbildung nach diesem Vorschlag verzichtet.

2.2.3 Fazit

Zusammenfassend kann festgestellt werden, dass jede der vorgestellten Regelungen spezifische Vor- und Nachteile hat. Die Anforderungen für das NIFTi-UAV erfüllt jedoch keine der Regelungen. Aus diesem Grund wird die Regelung von Grund auf neu entwickelt.

3 Modellbildung

In diesem Kapitel wird ein mathematisches Ersatzmodell der Kinematik des UAV hergeleitet. Die Struktur des Modells wird hierbei zuerst eindimensional betrachtet (Abschnitt 3.1). Die Erweiterung des Modells auf zwei Dimensionen folgt in Abschnitt 3.2. Dieses ist ein Sonderfall des allgemeinen, dreidimensionalen Modells, welches in Abschnitt 3.3 entwickelt wird.

3.1 Eindimensionales Modell (zwei Freiheitsgrade)

Im eindimensionalen Fall wird das UAV wie ein Massepunkt behandelt, welcher sich auf einer Achse X beschleunigt bewegen kann. Weiterhin kann es sich um einen Winkel Θ_{UAV} um die eigene Achse drehen. Die Drehachse des UAV liegt senkrecht auf der translatorischen Achse. Es hat demnach mit einem translatorischen und einem rotatorischen insgesamt zwei Freiheitsgrade. Abbildung 3.1 zeigt den Modellansatz. In Abschnitt 3.1.1 werden die Modellgleichungen hergeleitet. Das Modell wird anschließend in Abschnitt 3.1.3 vereinfacht und linearisiert. Abschließend findet im Abschnitt 3.1.4 eine Parameterschätzung der zentralen Modellparameter statt.

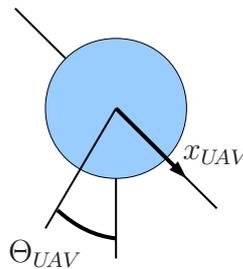


Abbildung 3.1: Ansatz für das eindimensionale Modell

3.1.1 Herleitung des Modells

Sei x_{UAV} die Position, v_{UAV} die Geschwindigkeit und a_{UAV} die Beschleunigung des UAV, so gelten die Differenzialgleichungen 3.1 und 3.2.

$$\int a_{UAV}(t) dt = v_{UAV}(t) \quad (3.1)$$

$$\int v_{UAV}(t) dt = x_{UAV}(t) \quad (3.2)$$

Weiterhin gilt die Grundgleichung der Mechanik (Gleichung 3.3), in welcher die reduzierte Strahlkraft $F_{Strahl,red}$ die Kraft ist, welche das UAV mit der Masse m_{UAV} mit a_{UAV} beschleunigt.

$$F_{Strahl,red}(t) = m_{UAV} \cdot a_{UAV}(t) \quad (3.3)$$

Da das UAV der (in diesem Fall als geschwindigkeitsproportional angenommenen) Luftreibung b_{Luft} ausgesetzt ist, reduziert sich die zur Beschleunigung verfügbare Kraft um die der Bewegung entgegengesetzte Reibkraft F_{reib} . Diesen Zusammenhang zeigen Gleichung 3.4 und 3.5.

$$F_{Strahl,red}(t) = F_{Strahl}(t) - F_{reib}(t) \quad (3.4)$$

$$F_{reib}(t) = b_{Luft} \cdot v_{UAV}(t) \quad (3.5)$$

Die Strahlkraft F_{Strahl} wird durch die Eigenschaften der Propeller, der Dichte der Luft und aus der reduzierten Strahlgeschwindigkeit $v_{Strahl,red}$ berechnet. Diese konstanten Systemeigenschaften werden im Folgenden in der Konstanten C_{Schub} zusammengefasst (Gleichung 3.6). Die Herleitung wird z.B. in [Brä04] beschrieben.

$$v_{Strahl,red}(t) = C_{Schub} \cdot F_{Strahl}(t) \quad (3.6)$$

Die reduzierte Strahlgeschwindigkeit $v_{Strahl,red}$ wird deshalb verwendet, da nur die um

die Bewegungsgeschwindigkeit des UAV reduzierte Strahlgeschwindigkeit $v_{Strahl,eff}$ (bewegtes Bezugssystem) zu einer Beschleunigung beitragen kann (Gleichung 3.7).

$$v_{Strahl,red}(t) = v_{Strahl,eff}(t) - v_{UAV}(t) \quad (3.7)$$

Betrachtet man nun das UAV im Schwebeflug, so wird deutlich, dass nur der in (genauer gesagt entgegen) der Bewegungsrichtung gerichtete Strahlanteil zu einer Beschleunigung beiträgt. Dies kann über eine Projektion der Strahlgeschwindigkeit \vec{v}_{Strahl} auf den in Achsenrichtung liegenden Einheitsvektor \vec{e}_x beschrieben werden (Gleichung 3.8). Es ergibt sich die Gleichung 3.9. Θ_{UAV} ist der Winkel, den das UAV zur Bodennormalen aufweist. Im neigungslosen Schwebeflug ist demnach $\Theta_{UAV} = 0^\circ$; die effektive Strahlgeschwindigkeit in Bewegungsrichtung $v_{Strahl,eff} = 0 \frac{m}{s}$.

$$v_{Strahl,eff}(t) = \vec{v}_{Strahl}(t) \cdot \vec{e}_x \quad (3.8)$$

$$v_{Strahl,eff}(t) = \sin(\Theta_{UAV}(t)) \cdot v_{Strahl}(t) \quad (3.9)$$

Weiterhin ist die Strahlgeschwindigkeit v_{Strahl} über die Steigung der Propeller c_{Prop} mit der Motordrehzahl n_{Motor} gekoppelt. Dies bringt Gleichung 3.10 zum Ausdruck.

$$v_{Strahl}(t) = C_{Prop} \cdot n_{Motor}(t) \quad (3.10)$$

Abbildung 3.2 zeigt den Signalflussplan der Regelstrecke im Bildbereich, wie er nach obiger Herleitung aussieht.

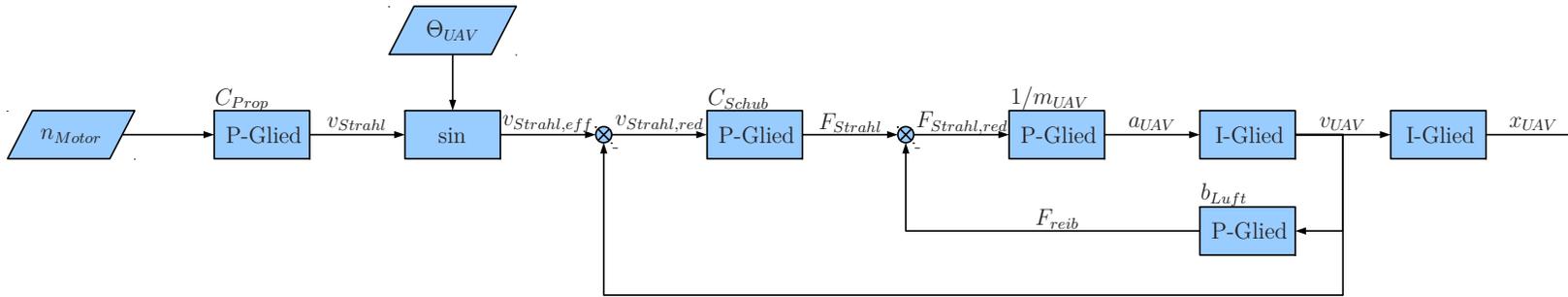


Abbildung 3.2: Eindimensionales Ersatzmodell der Regelstrecke

3.1.2 Annahmen zur Vereinfachung des Modells

Es werden nun Annahmen getroffen, die die Modellierung und spätere Reglerentwicklung vereinfachen. Diese sind nicht allgemeingültig und schränken unter Umständen die Leistungsfähigkeit des Gesamtsystems ein. Es werden die folgenden Annahmen getroffen:

Kleine Auslenkungen aus der Ruhelage

Das UAV wird durch die Positionsregelung nur um kleine Winkel aus der Ruhelage ausgelenkt. Dies hat zur Folge, dass die nichtlineare Sinusfunktion (Gleichung 3.9) nur im annähernd linearen Bereich betrieben wird. Diese wird so zu einer linearen Multiplikation vereinfacht.

Überlagerte Höhenregelung

Die Höhe des UAV wird durch eine überlagerte Höhenregelung geregelt. Diese wird als vollständig eingeschwungen angenommen. Der Schub in Richtung Erdmittelpunkt ist demnach konstant und genau so groß, dass er die Gewichtskraft des UAV kompensiert.

$$F_{Strahl,Auftr}(t) = \cos(\Theta_{UAV}(t)) \cdot F_{Strahl} \quad (3.11)$$

Mit Annahme der kleinen Auslenkungen (Abschnitt 3.1.2) ergeben sich nach Gleichung 3.11, auch durch die Verkipfung des UAV um den Winkel Θ_{UAV} , nur kleine Störgrößen für die Höhenregelung. Die Cosinus-Funktion ist in diesem Bereich nahezu konstant. Hieraus folgt die Annahme einer konstanten Motordrehzahl für das Modell in Gleichung 3.12 und 3.13.

$$n_{Motor}(t) \rightarrow n_{Motor} \quad (3.12)$$

$$v_{Strahl}(t) \rightarrow v_{Strahl} \quad (3.13)$$

Auch wenn das UAV im eindimensionalen Modell keinen Höhenfreiheitsgrad hat, so ist dennoch wichtig, dass die Motordrehzahl n_{Motor} und damit die Strahlgeschwindigkeit v_{Strahl} konstant ist.

Kleine Geschwindigkeiten

Wie in Abschnitt 3.1.2 implizit erwähnt, fliegt das UAV mit im Vergleich zur Strahlgeschwindigkeit geringen Geschwindigkeiten. Hierdurch kann die Rückkopplung, welche zur Reduktion der effektiven Strahlgeschwindigkeit führt, eliminiert werden.

3.1.3 Vereinfachung des Modells

Abbildung 3.3 zeigt das Modell nach der Vereinfachung durch obige Annahmen. Die Konstanten sind zusammengefasst und der durch die Rückkopplung der Luftreibung geschlossene Weg der Strahlkraft F_{Strahl} zur Geschwindigkeit v_{UAV} ist in ein PT1-Glied überführt worden.

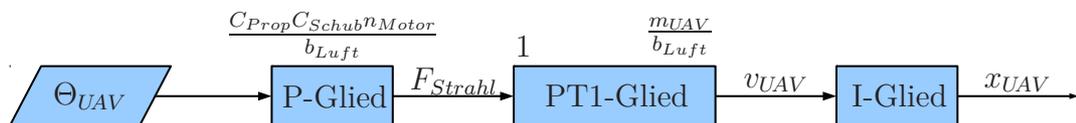


Abbildung 3.3: Vereinfachtes eindimensionales Ersatzmodell der Regelstrecke

Abbildung 3.4 zeigt die Zusammenfassung des Modells. Dieses eindimensionale Modell wird in den Modellen höherer Ordnung als bekannt vorausgesetzt.

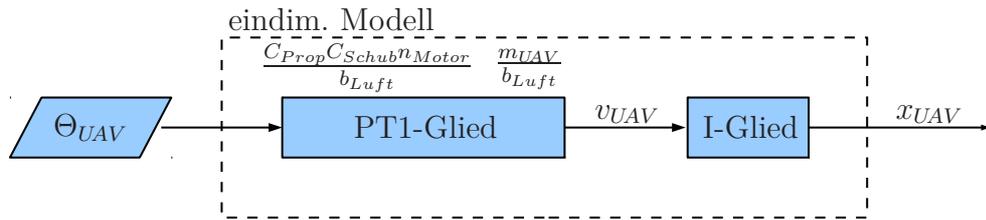


Abbildung 3.4: Zusammengefasstes eindimensionales Ersatzmodell der Regelstrecke

3.1.4 Modellparameter

Die Stärke der Rückkopplung durch den Luftwiderstand ist elementar für das Verhalten des Modells. Ist die Rückkopplung im Gegensatz zur UAV-Masse groß, geht die Zeitkonstante $\tau = \frac{m_{UAV}}{b_{Luft}} \rightarrow 0$. Das PT1-Glied mit der Verstärkung 1 kann dann als neutrales Element vernachlässigt werden. Die Regelstrecke ist ein I-Glied in Reihe mit einem P-Glied (Abbildung 3.5).

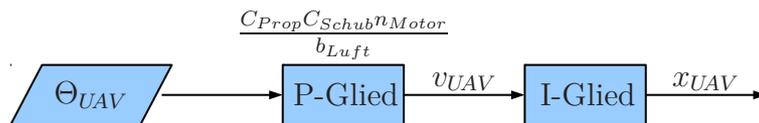


Abbildung 3.5: Ersatzmodell der Regelstrecke mit starker Reibung

Ist die Luftreibung stattdessen klein, verhält sich das PT1-Glied annähernd wie ein I-Glied. Die Strecke verhält sich doppelt integrierend (Abbildung 3.6).

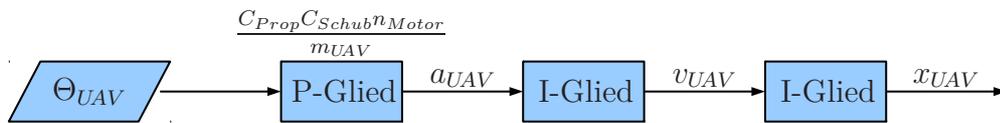


Abbildung 3.6: Ersatzmodell der Regelstrecke mit schwacher Reibung

Um herauszufinden, welches Modell das Verhalten des UAV am besten beschreibt, werden im Motion-Capturing-Labor am Institut für Informatik VI Autonome Intelligente Systeme der Rheinischen Friedrich-Wilhelms-Universität Bonn Versuche durchgeführt. Diese Versuche ermöglichen es außerdem, die Parameter des Modells zu schätzen.

Versuchsaufbau

Abbildung 3.7 zeigt den Versuchsaufbau. Das Messsystem besteht aus neun Infrarotkameras, welche einen etwa 20 m^3 großen Bereich filmen. Die Kamerabilder werden mit etwa 100 Hz abgetastet. Ein Algorithmus erkennt auf allen Kamerabildern drei am UAV befestigte Reflektoren. Aus diesen Daten extrahiert das System die Koordinaten und die Verkippung des Schwerpunkts der Reflektoren bzw. des UAV. Die Genauigkeit des Systems variiert je nach Position stark. Da diese aber immer unter 1 mm liegt, kann sie bei den folgenden Messungen vernachlässigt werden.

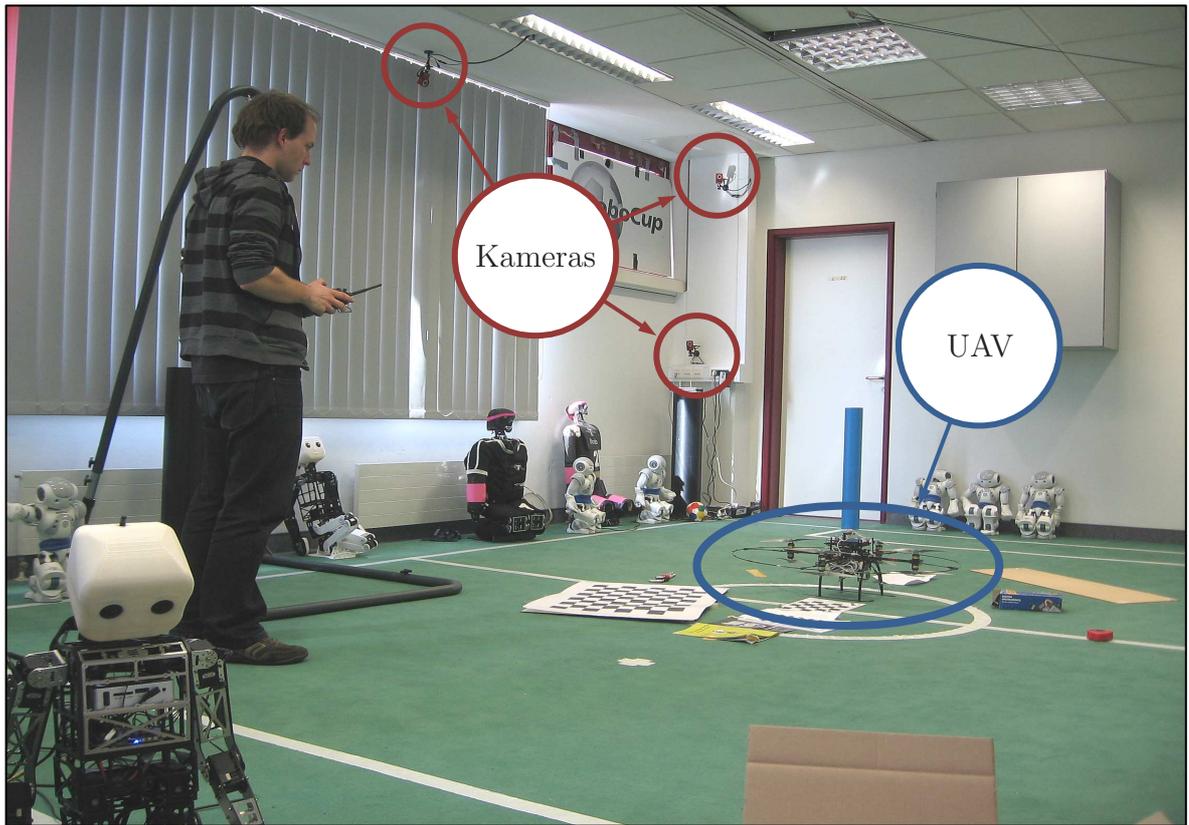


Abbildung 3.7: Versuchsaufbau zur Parameterschätzung

Durchführung

Da der begrenzte Raum nur kleine Geschwindigkeiten zulässt, ist die Aufnahme von Sprungantworten kaum möglich. Stattdessen wird das UAV per Fernsteuerung so gesteuert, dass es im Messraum eine beliebige, eindimensionale Trajektorie abfliegt. Sowohl der Winkel Θ_{UAV} , also auch die hieraus resultierende Position x_{UAV} wird aufgenommen.

Weiterhin wird in MATLAB/Simulink ein Modell der Regelstrecke ohne Reibung entworfen (Abbildung 3.8). Als Eingangsdaten für das Modell dient der aufgenommene Winkel Θ_{UAV} (SimAngle).

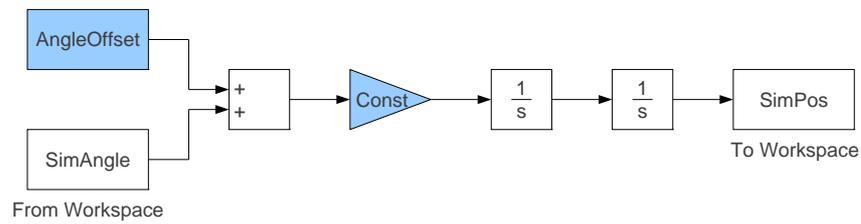


Abbildung 3.8: Simulinkmodell der Regelstrecke mit schwacher Reibung

Ergebnisse

Unter Zuhilfenahme eines Gradientenverfahrens wird eine numerische Optimierung des Modellparameters „Const“ durchgeführt. Die Trajektorie des Modells nähert sich so schrittweise an die des realen UAV an. Auch der Parameter „AngleOffset“ wird optimiert. Dieser kompensiert einen Winkeloffset im Motion-Capturing-System und ist für das UAV nicht von Bedeutung. Abbildung 3.9 zeigt die Eingangsdaten, die Position des realen UAV und des Modells.

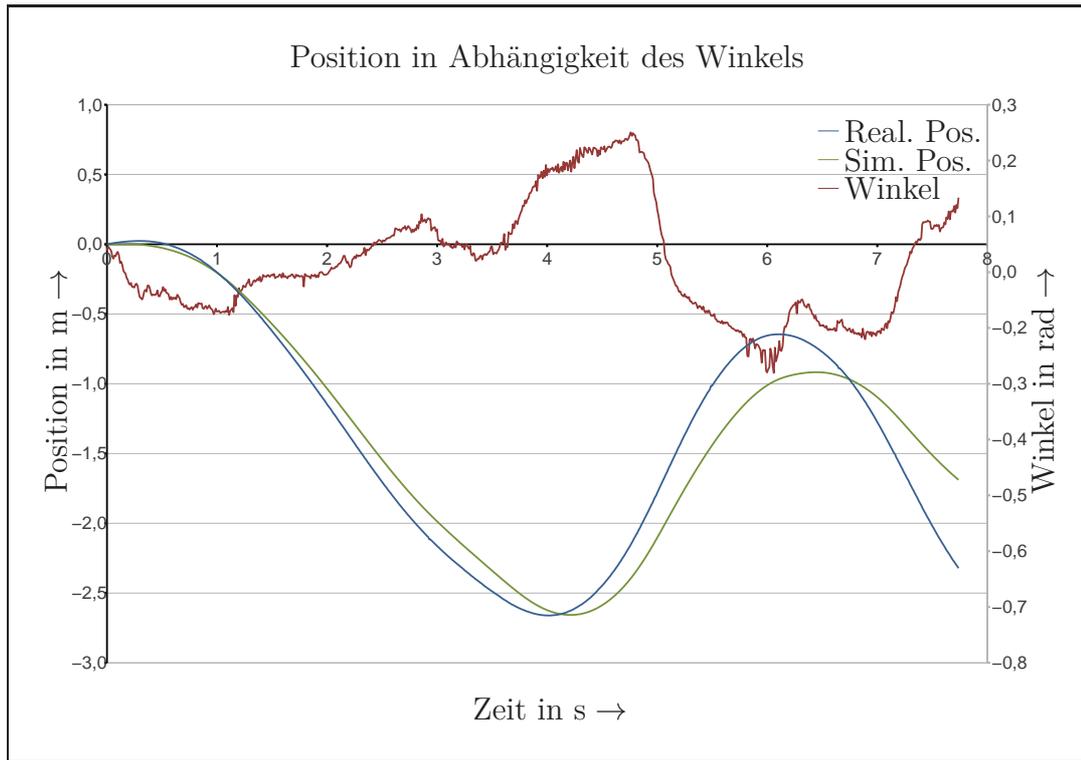


Abbildung 3.9: Versuch zur Parameterschätzung

Es zeigt sich, dass das System bei den im begrenzten Areal möglichen Geschwindigkeiten ohne Reibung modelliert werden kann. Der Parameter „Const“ ist ein Maß für die Beschleunigung in Abhängigkeit des Winkels. Er wird mit C_{beschl} bezeichnet und ergibt sich wie in Gleichung 3.14 dargestellt.

$$C_{beschl.} = \frac{C_{Schub} C_{Prop} n_{Motor}}{m_{UAV}} = 9,3 \frac{\text{m}}{\text{s}^2} \quad (3.14)$$

3.2 Zweidimensionales Modell (vier Freiheitsgrade)

Das zweidimensionale Modell verfügt über zwei translatorische und über zwei rotatorische Freiheitsgrade. Abbildung 3.10 zeigt, dass das Modell sich auf den Achsen um x_{UAV} und y_{UAV} bewegen und um die Winkel Θ_{UAV} und Φ_{UAV} drehen kann.

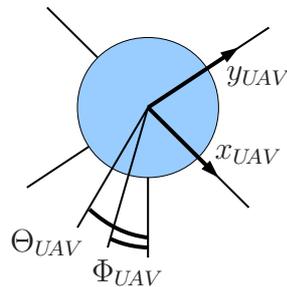


Abbildung 3.10: Ansatz für das zweidimensionale Modell

Mit Annahme 3.1.2 dürfen beide Achsen als vollständig entkoppelt angesehen werden. Deshalb kann die zweidimensionale Bewegung als Superposition von zwei eindimensionalen Bewegungen modelliert werden. Abbildung 3.11 zeigt das zweidimensionale Modell.

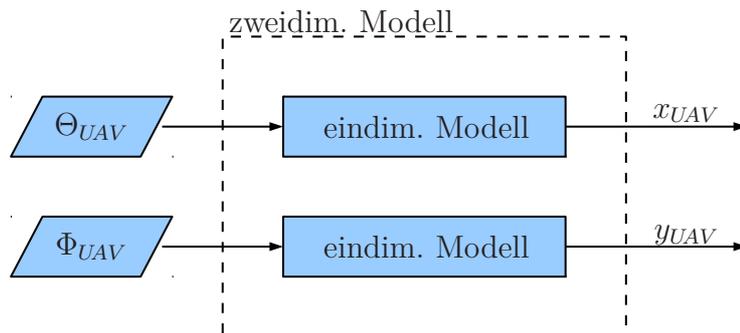


Abbildung 3.11: Zweidimensionales Ersatzmodell der Regelstrecke

3.3 Dreidimensionales Modell (sechs Freiheitsgrade)

Für eine vollständige Modellierung aller sechs Freiheitsgrade ist es notwendig, die dritte translatorische Bewegung z_{UAV} in z -Richtung und eine Rotation Ψ_{UAV} um diese zuzulassen. Abbildung 3.12 verdeutlicht dies.

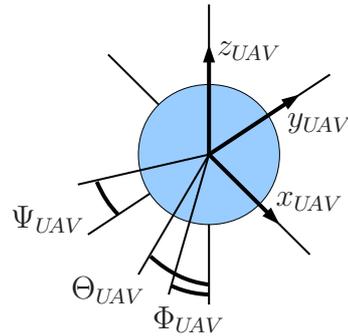


Abbildung 3.12: Ansatz für das dreidimensionale Modell

Tritt nun ein Winkel $\Psi_{UAV} \neq 0^\circ$ auf, bewirkt dies eine Verdrehung der Schubvektoren. Aus diesem Grund wird ein zweites Koordinatensystem, das Weltkoordinatensystem, eingeführt. Dieses hat analog zum Koordinatensystem des UAV die Achsen x_{Welt} , y_{Welt} und z_{Welt} . Die positive x-Achse des Weltkoordinatensystems x_{Welt} wird so festgelegt, dass sie nach Norden zeigt. Zur Überführung beider Koordinatensysteme ineinander findet eine Koordinatentransformation statt.

Der Ursprung der Koordinatensysteme wird für das Modell als identisch angesehen. Mit Annahme 3.1.2 kann außerdem $z_{Welt} = z_{UAV} = konst.$ angenähert werden. Abbildung 3.13 zeigt das vollständige Modell mit allen sechs möglichen Freiheitsgraden.

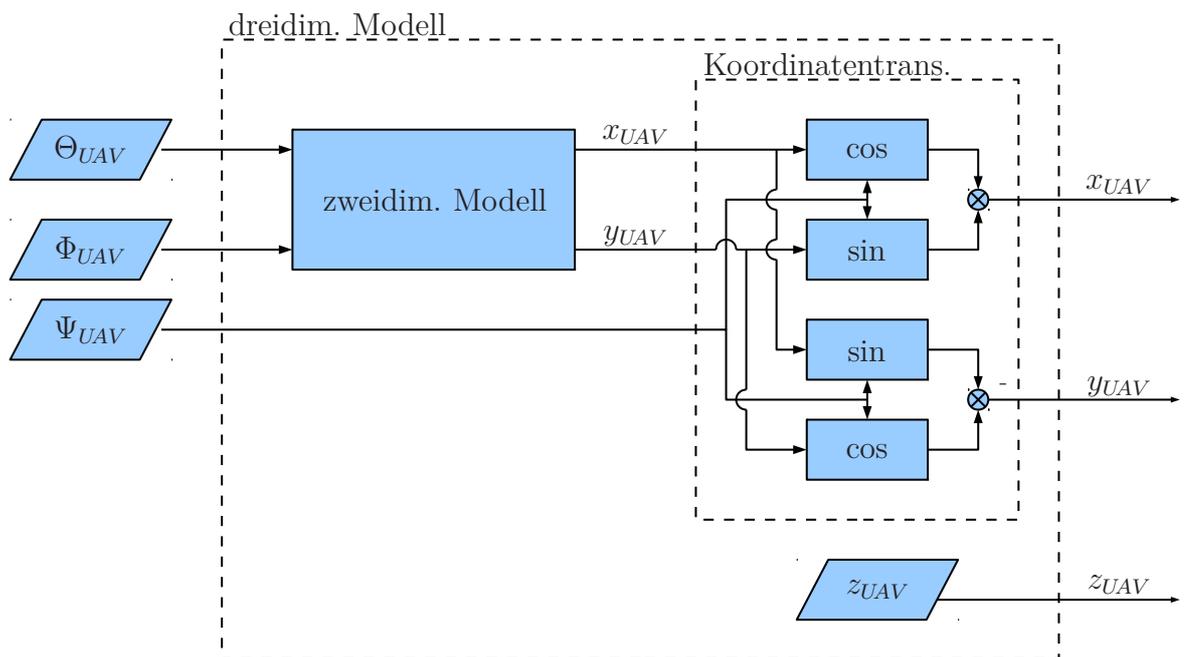


Abbildung 3.13: Dreidimensionales Ersatzmodell der Regelstrecke

Dieses dreidimensionale Modell wird im folgenden Kapitel 4 verwendet, um einen Regelalgorithmus herzuleiten. Es wird als bekannt vorausgesetzt.

4 Reglerentwurf

In Kapitel 3 wurde ein Modell der Regelstrecke erstellt. Mit diesem Modell wird im Folgenden ein Algorithmus entworfen, welcher die Position des UAV automatisch auf eine definierte Sollposition regelt.

Hierzu werden im Abschnitt 4.1 Annahmen getroffen, welche die spätere Entwicklung des Reglers vereinfachen. Anschließend findet in Abschnitt 4.2 eine Analyse des Modells statt. Hierauf aufbauend wird in Abschnitt 4.3 ein Regler entworfen, welcher abschließend in Abschnitt 4.4 parametrisiert wird.

4.1 Annahmen zur Reglerentwicklung

Neben den im Abschnitt 3.1.2 gemachten Annahmen zur Vereinfachung der Modellierung werden für die Reglerentwicklung zusätzlich die folgenden Annahmen getroffen.

4.1.1 Unterlagerte Lageregelung

Das UAV verfügt über eine unterlagerte Lageregelung für die Winkel Θ_{UAV} und Φ_{UAV} . Diese wird als so schnell und präzise angesehen, dass Winkelvorgaben unmittelbar umgesetzt werden können. Es entstehen keine für die Positionsregelung relevanten Verzögerungen.

4.1.2 Pseudokontinuierliche Regelung

Die Berechnung der Regelung erfolgt auf einem Digitalrechner. Diese erfolgt so schnell, dass durch die diskrete Datenverarbeitung keine Beeinflussung der Regelung stattfindet. Die Regelung erfolgt pseudokontinuierlich.

4.2 Analyse des dreidimensionalen Modells

In Abschnitt 3.3 wird gezeigt, dass die Kinematik des UAV im allgemeinen dreidimensionalen Fall zweidimensional modelliert werden kann, wenn sowohl die Höhe konstant ist, als auch eine Projektion der UAV-Koordinaten auf die Weltkoordinaten statt findet. Dieses zweidimensionale Modell ist wiederum eine Superposition von zwei eindimensionalen Modellen. Abbildung 4.1 zeigt die grundlegende Struktur der Regelung auf Basis von (GPS-)Weltkoordinaten.

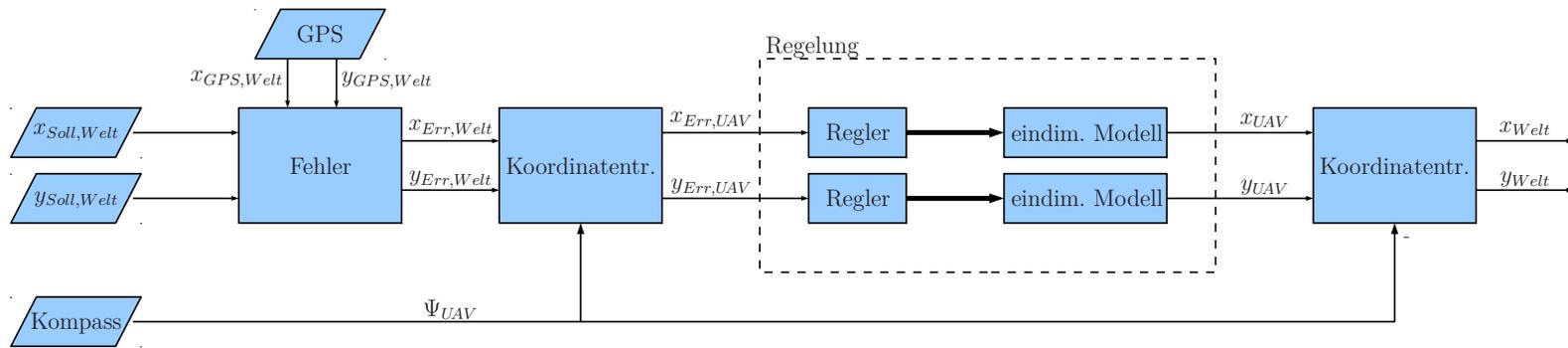


Abbildung 4.1: Überblick der Struktur der Regelung

Die grundlegende Aufgabe eines Reglers ist demnach die Regelung dieses eindimensionalen Modells. Im Folgenden gehen deshalb $x_{Soll,UAV}$ und $x_{Err,UAV}$ in x_{Soll} und x_{Err} über. Analog gehen $y_{Soll,UAV}$ und $y_{Err,UAV}$ in y_{Soll} und y_{Err} über.

Die eindimensionale Regelstrecke besteht nach Abbildung 3.4 aus einer (bekannten) Konstanten, auf welche zwei in Reihe geschaltete I-Glieder folgen.

4.3 Reglerstruktur

Die beiden Integratoren erzeugen eine Phasenverschiebung von -180° ; das System ist demnach strukturinstabil. Um dieser Instabilität entgegen zu wirken, wird eine künstliche Dämpfung in das System integriert. Diese wirkt genau wie die Luftdämpfung phasenanhebend. Des Weiteren wird der zweite Integrator über einen P-Regler gekoppelt.

Abbildung 4.2 zeigt das Ergebnis. Es handelt sich bei dieser Reglerstruktur um einen Kaskadenregler. Dieser kann optimales (unendlich schnell, kein Überschwingen, keine stationäre Regelabweichung) Einschwingverhalten zeigen.

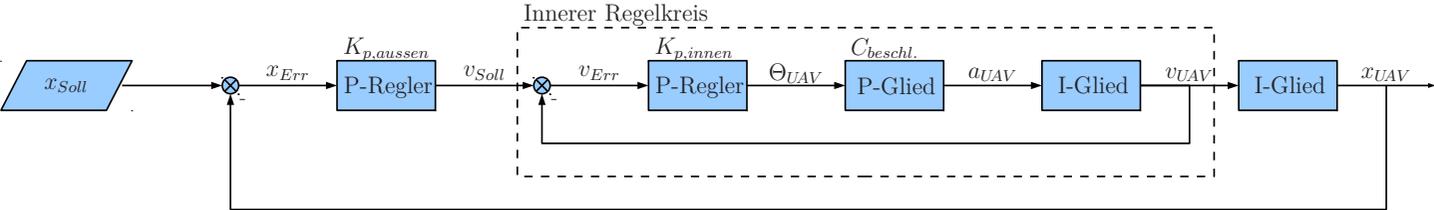


Abbildung 4.2: Kaskadenregelung

4.4 Reglerparametrierung

Die proportionalen Parameter der beiden Regler müssen bestimmt werden. Um 3.1.2 gerecht zu werden muss außerdem der Maximalwinkel des UAV begrenzt werden. Die zu parametrierenden Parameter sind demnach:

- Θ_{max}
- $K_{p,aussen}$
- $K_{p,innen}$

Da die Modellparameter bekannt sind, könnte mit einschlägigen Verfahren (z.B. symmetrisches Optimum, Polkompensation u. v. a.) eine Parametrierung des Reglers stattfinden. Weitere Auslegungsverfahren sind z.B. in [Lun11] beschrieben. Zum einen ist dieses Vorgehen entweder sehr aufwändig und/oder stark fehlerbehaftet (vgl. [Beu11]), zum anderen verhindert die starke Stellgrößenbegrenzung ein Auslegen nach Faustformeln. Aus diesen Gründen soll hier eine vollständig neuartige Parametrierungsmethodik angewandt werden:

Zuerst wird in 4.4 der Maximalwinkel Θ_{max} aus Erfahrungswerten festgelegt. Die hieraus resultierende Stellgrößenbegrenzung hat zur Folge, dass die maximal mögliche Beschleunigung begrenzt ist. Über physikalische Gleichungen wird in 4.4 ein nichtlinearer Regler hergeleitet, welcher das schnellstmögliche Einschwingen erzeugt. Abschließend wird in 4.4 der proportionale Verstärkungsfaktor $K_{p,innen}$ aufgrund von Vorversuchen parametriert.

Bestimmung von Θ_{max}

Die Stellgrößenbeschränkung Θ_{max} eignet sich, um eine Bewertung des Feedbacksystems durchzuführen. Ist das Feedbacksystem gut¹ (z.B. Motion-Capturing), werden große Auslenkungen aus der Neutralen erlaubt. Die Regelung ist aggressiv. Bei schlechtem Feedback (z.B. GPS) werden nur kleine Auslenkungen erlaubt. Die Regelung ist weniger aggressiv. Dieses adaptive Verhalten wird im Ausblick (Kapitel 7.2) weiter ausgeführt.

Beobachtet man Piloten beim Fliegen des UAV, so zeigt sich, dass der maximale Winkel Θ_{max} stark von der Erfahrung des Piloten abhängig ist. Insbesondere Anfänger trauen sich nur kleine Auslenkungen von $\Theta_{max} \sim \pm 10^\circ$ zu. Mit einem Sicherheitsfaktor von

¹Schnell im Vergleich zur UAV Zeitkonstanten, genau im Vergleich zu den UAV-Abmessungen.

$S = 2$ beträgt der maximale Winkel demnach $\Theta_{max} = \pm 5^\circ$.

Bestimmung von $K_{p,aussen}$

Ist $K_{p,aussen}$ zu groß ausgelegt, ergibt sich eine zu große Anfluggeschwindigkeit auf die Sollposition. Das UAV schafft es nicht mehr rechtzeitig abzubremsen und schwingt über die Position über. Diagramme 4.3 und 4.4 visualisieren dieses Verhalten. Obwohl der Winkel des UAV in der Begrenzung ist, kann es der Sollgeschwindigkeit nicht folgen. Beim Beschleunigen äußert sich dies in einer längeren Anregelzeit. Beim Abbremsen führt die Stellgrößenbegrenzung zu unerwünschtem Überschwingen. Die in der Beschleunigungsphase in das UAV eingespeiste, kinetische Energie kann dann nicht schnell genug abgebaut werden.

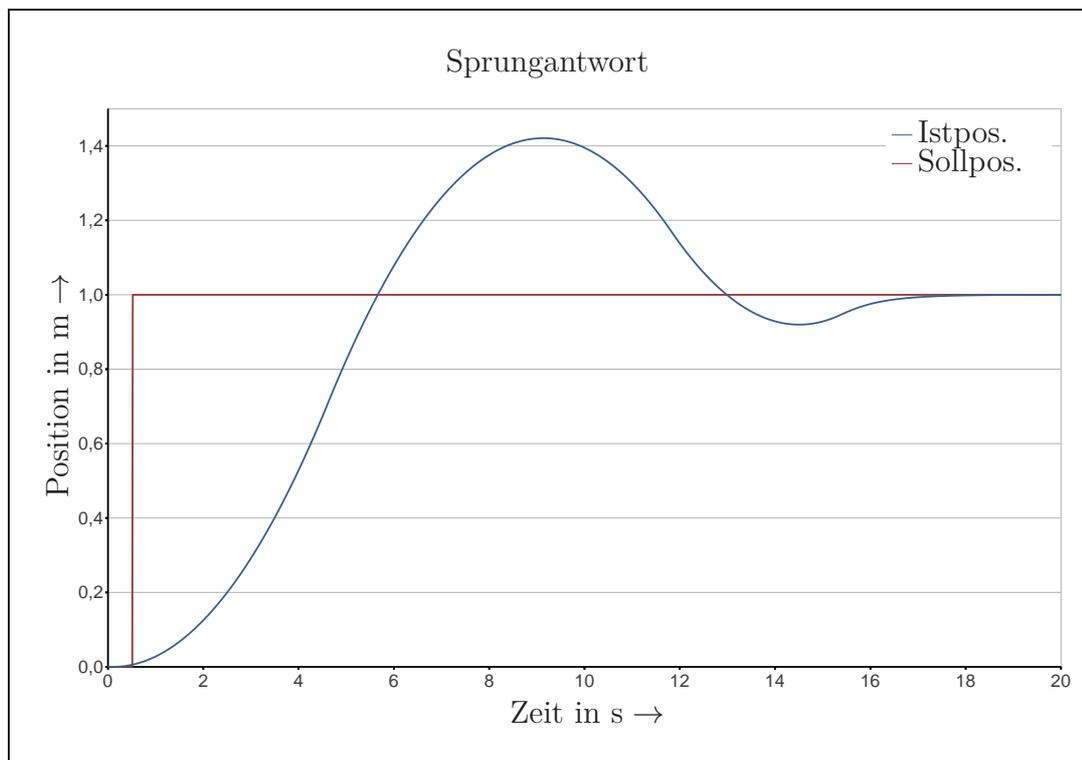


Abbildung 4.3: Überschwingen bei zu starkem $K_{p,aussen}$ (Position)

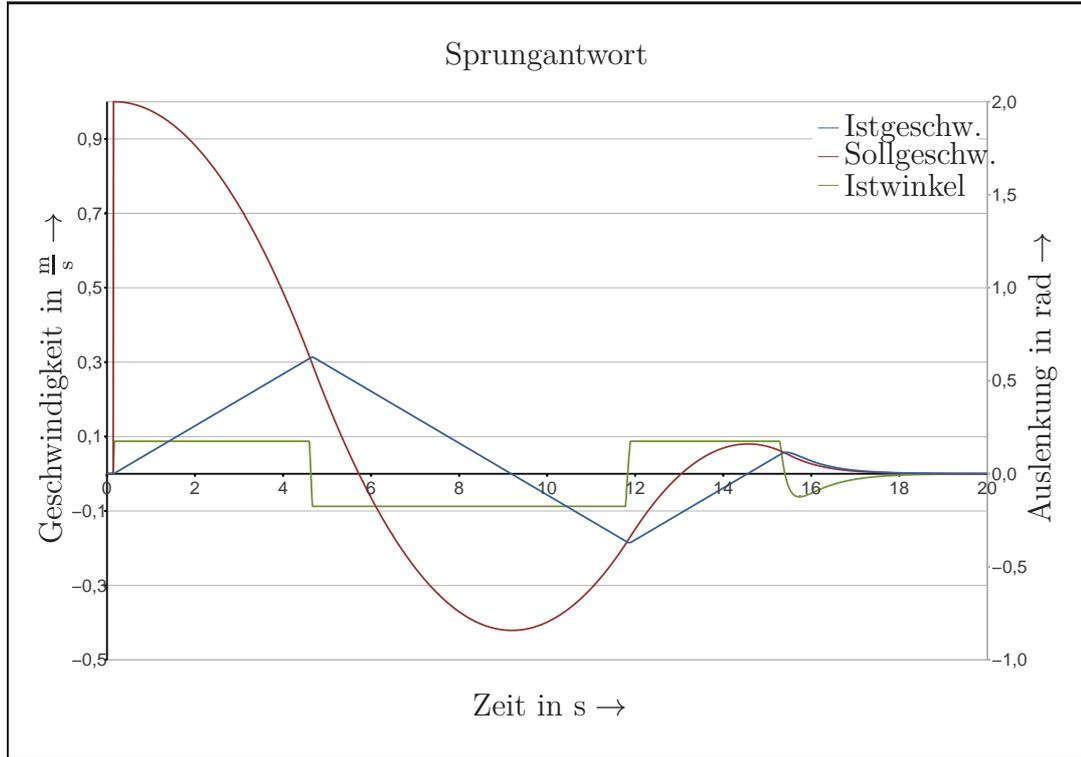


Abbildung 4.4: Überschwingen bei zu starkem $K_{p,aussen}$ (Geschwindigkeit)

Die Steigung der Sollgeschwindigkeit darf zu keinem Zeitpunkt die Steigung der Istgeschwindigkeit, welche durch Θ_{max} begrenzt ist, überschreiten. Dies ist in Abbildung 4.4 ab $t = 5\text{ s}$ zu erkennen. Um dieses Verhalten zu verhindern, wird der Wert $K_{p,aussen,krit}$ bestimmt. $K_{p,aussen,krit}$ erzeugt (bei $K_{p,innen} \rightarrow \infty$) ein optimales Regelverhalten des UAV.

Bei Sollwertsprung geht das UAV in den Zustand „volle Beschleunigung“ in Richtung des Sollwerts. Bei Überschreiten der Sollgeschwindigkeit geht es in den Zustand „volles Abbremsen“ über. Genau auf der Sollposition erreicht es die Geschwindigkeit $v_{UAV} = 0 \frac{m}{s}$ und geht in die neutrale Lage über.

$$a_{max} = \Theta_{max} \cdot C_{beschl.} \quad (4.1)$$

$$x_{max} = \iint a_{max} dt dt = \frac{1}{2} a_{max} t^2 \quad (4.2)$$

$$v_{max} = \int a_{max} dt = a_{max} \cdot t_{max} \quad (4.3)$$

$$t_{max} = \frac{v_{max}}{a_{max}} \quad (4.4)$$

$$v_{max} = K_{p,aussen,krit} \cdot x_{max} \quad (4.5)$$

$$K_{p,aussen,krit} = \sqrt{2 \frac{\Theta_{max} C_{beschl.}}{x_{max}}} \quad (4.6)$$

Der Parameter $K_{p,aussen,krit}$ muss demnach für eine optimale Regelung ein nichtlineares Verhalten aufweisen. Wird die Sollgeschwindigkeit $|v_{Soll}|$ über den Positionsfehler aufgetragen, ergibt sich das in Abbildung 4.5 dargestellte Vektorfeld. Da das Vektorfeld, wegen der Nichtlinearität nicht rotationssymmetrisch ist, weisen die außerdem im Bild dargestellten Strömungslinien keinen geraden Verlauf auf. Dies hat zur Folge, dass das UAV in einem Bogen auf das Ziel zufliegt.

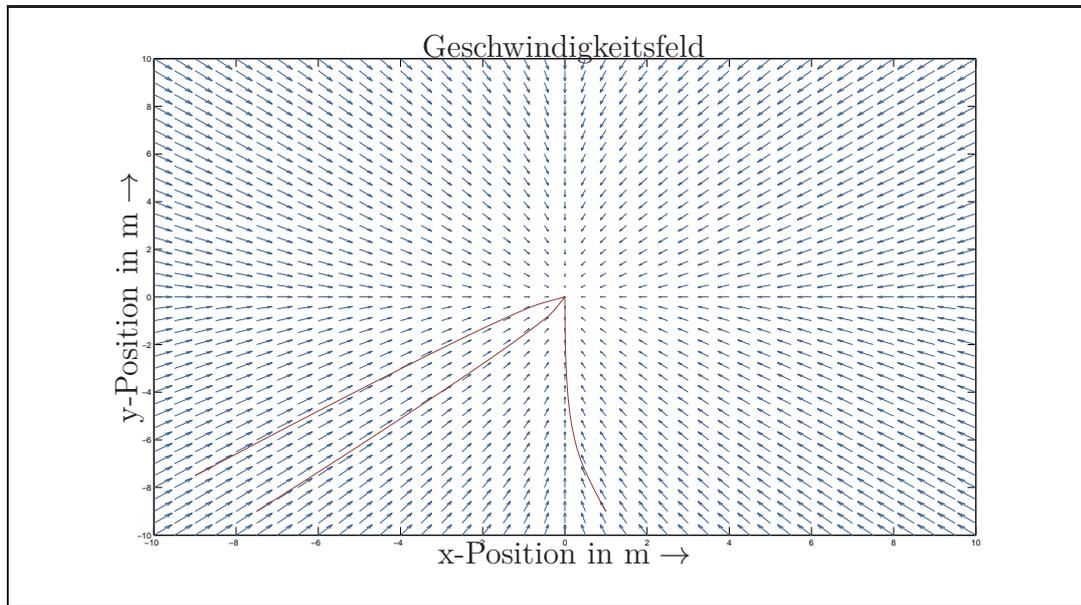


Abbildung 4.5: Sollgeschwindigkeit bei Positionsfehler

Die Nichtlinearität hat außerdem zur Folge, dass ein Drehen des UAV um die z-Achse und die damit verbundene Koordinatentransformation großen Einfluss auf die Sollgeschwindigkeit des UAV haben kann. Um diesen Effekt besser erkennen zu können, ist

der Betrag der Geschwindigkeit in Abbildung 4.6 abgebildet.

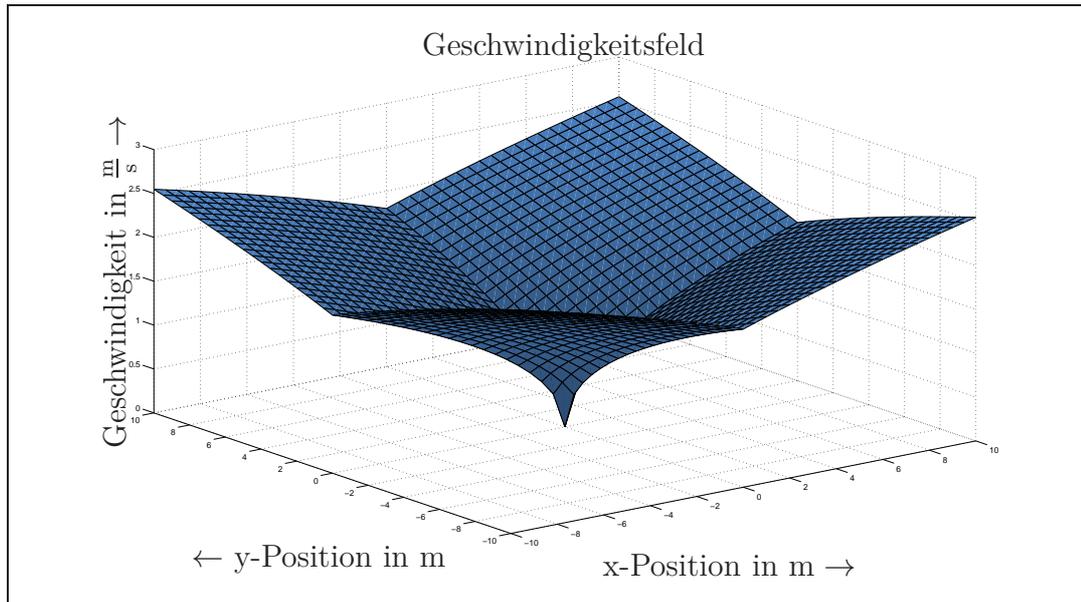


Abbildung 4.6: Betrag Sollgeschwindigkeit bei Positionsfehler

Da diese prädiktive Regelung modellbasiert arbeitet, ist sie gegen Parameterschwankungen nur begrenzt robust. Diese haben allerdings aufgrund der Wurzel quantitativ nur geringen Einfluss.

Bestimmung von $K_{p,innen}$

In Vorversuchen konnte folgendes beobachtet werden: Beim Anfliegen eines Punktes steuern Piloten das UAV so, als würde es an einem Seil hängen und wie ein Fadenpendel gedämpft auf die Sollposition einschwingen. Die Höhe der Verankerung ist ein Maß für die Erfahrung des Piloten und für die Aggressivität der Regelung. Je niedriger die virtuelle Verankerung (vgl. h'_{UAV}), desto aggressiver der Regler. Abbildung 4.7 verdeutlicht diesen Zusammenhang.

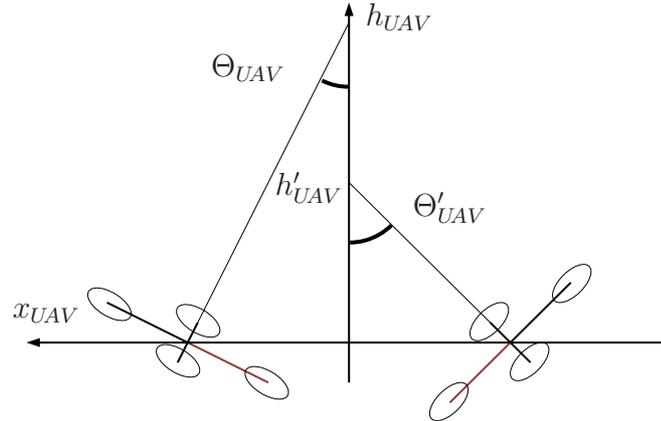


Abbildung 4.7: Parametrierung am Fadenpendel

Als guter² Wert der Verankerungshöhe für menschliche Piloten beim Flug unter freiem Himmel erweist sich $h_{UAV} = 10$ m. Mit einem Sicherheitsfaktor von $S = 2$ beträgt die Verankerungshöhe demnach $h_{UAV} = 20$ m. Nach linearer Interpolation der Tangensfunktion zeigen Gleichungen 4.7 bis 4.11, dass das UAV hier bei einem Abstand von $x_{Begr,UAV} = 1,75$ m in die Winkelbegrenzung von $\Theta_{UAV} = \Theta_{max} = 5^\circ$ geht.

$$\tan(\Theta_{UAV}) = \frac{x_{UAV}}{h} \quad (4.7)$$

$$x_{UAV} = h \cdot \tan(\Theta_{UAV}) \quad (4.8)$$

$$x_{Begr,UAV} = h \cdot \tan(\Theta_{max}) \quad (4.9)$$

$$x_{Begr,UAV} = 20 \text{ m} \cdot \tan(5^\circ) \quad (4.10)$$

$$x_{Begr,UAV} = 1,75 \text{ m} \quad (4.11)$$

Wird nun von einem stillstehenden UAV ausgegangen ($v_{UAV} = 0 \frac{\text{m}}{\text{s}} \rightarrow v_{Err,UAV} = v_{Soll,UAV}$), zeigen Gleichungen 4.12 bis 4.15 die Herleitung von $K_{p,innen}$ mit $\Theta_{Soll,UAV}$ in Grad.

²Mäßig aggressiver Flug eines erfahrenen Piloten.

$$v_{Soll,UAV} = K_{p,aussen} \cdot x_{Err,UAV} \quad (4.12)$$

$$\Theta_{Soll,UAV} = K_{p,innen} \cdot v_{Soll,UAV} \quad (4.13)$$

$$K_{p,innen} = \frac{\Theta_{max}}{K_{p,aussen} \cdot x_{Begr,UAV}} \quad (4.14)$$

$$K_{p,innen} = 0,39 \frac{\text{s}}{\text{m}} \quad (4.15)$$

4.5 Störverhalten

Auf der NIFTi Joint Exercise 2013 in Prato, Italien wurden Versuche durchgeführt um die Hauptstörquellen des Systems zu identifizieren. Abschnitte 4.5.1 und 4.5.2 beschreiben die Versuche. Außerdem wird eine Strategie zum Umgang mit den Störgrößen beschrieben.

4.5.1 Störung durch GPS

GPS weist systembedingte Ungenauigkeiten auf. Zum einen rufen Umwelteinflüsse (Wolken, Regen, Sonnenstürme etc.) eine absolute Ungenauigkeit hervor. Da es sich hierbei um relativ zur UAV-Zeitkonstanten langsame Vorgänge handelt, wirkt diese Ungenauigkeit nur auf die Position des UAV. Sie kann durch das System nicht gemessen und deshalb auch nicht kompensiert werden. Aus diesem Grund ist die UAV-Position im Bezug zur geodätischen Absolutposition immer offsetbehaftet.

Zum anderen weist GPS Rauschen auf. Dieses Rauschen zeigt sich in sprunghaften Messwerten einer konstanten Position. Zur Verdeutlichung zeigt Abbildung 4.8 eine Messreihe der NIFTi Joint Exercise 2013.

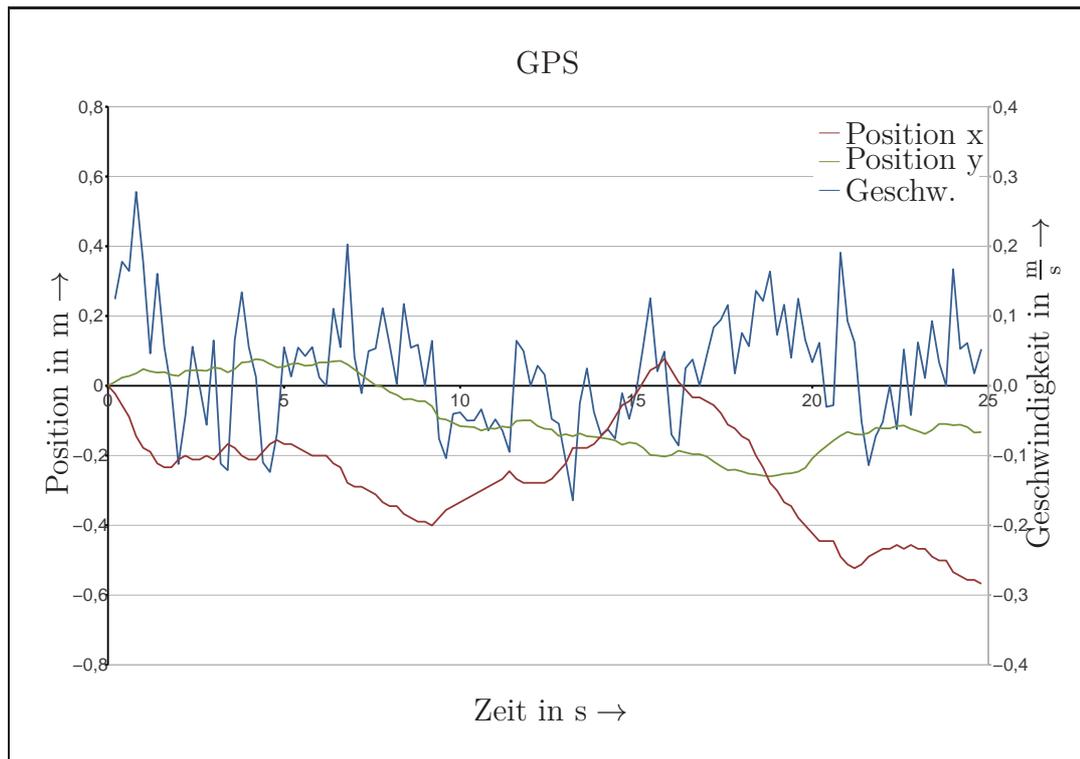


Abbildung 4.8: Rauschen des GPS

Die Frequenz des Rauschens folgt aus der Frequenz des GPS von 5 Hz (vgl. Abschnitt 2.1). Die Massenträgheit und die Stellgrößenbegrenzung des UAV wirkt in diesen Frequenzen stark dämpfend auf das System. Das Rauschen ist demnach in der Position des UAV nur mit sehr kleinen Amplituden vorhanden. Es kann aus diesem Grund für die Position des UAV ignoriert werden.

Anders verhält es sich bei der Ermittlung der Geschwindigkeit des UAV. Diese ist ein innerer Systemzustand und kann daher nicht direkt gemessen werden. Durch Differenzierung des kumulierten Positionssignals ist es dennoch möglich, auf die Geschwindigkeit zu schließen. Da das Positionssignal jedoch starkem Rauschen unterliegt, kann das tatsächliche Geschwindigkeitssignal nur durch starkes Tiefpassfiltern extrahiert werden. Da Tiefpassfilter jedoch immer eine Phasenverschiebung aufweisen, ist diese Art der Geschwindigkeitsermittlung für Echtzeitsysteme nicht möglich. Stattdessen wird die Geschwindigkeit durch einen im GPS integrierten Beobachter geschätzt. Dieser ist bereits parametrisiert und nicht auf die Benutzung auf einem UAV optimiert. Auf diesen Zusammenhang wird im Ausblick weiter eingegangen. Nur durch den Beobachter ist es möglich, auf den inneren Systemzustand Geschwindigkeit zu schließen, ohne die Regelung durch das starke Rauschen negativ zu beeinflussen.

4.5.2 Störung durch Wind

Auf der NIFTi Joint Exercise 2013 wurde unter anderem eine Rettungsübung durchgeführt. Hierbei simulieren Feuerwehrmänner einen radioaktiven Unfall. Da die Radioaktivität am Unfallort für Menschen zu stark ist, sind die Rettungskräfte zwingend auf die Nutzung des UAV angewiesen. Bei der manuellen Steuerung des UAV zeigt sich, dass Wind eine wesentliche Störgröße darstellt. Während die Ungenauigkeit des GPS störend sowohl auf die Position als auch auf die Geschwindigkeit des UAV wirkt, wirkt Wind als externe Kraft indirekt auf die Beschleunigung. Abbildung 4.9 zeigt, wie Wind das UAV beeinflusst.

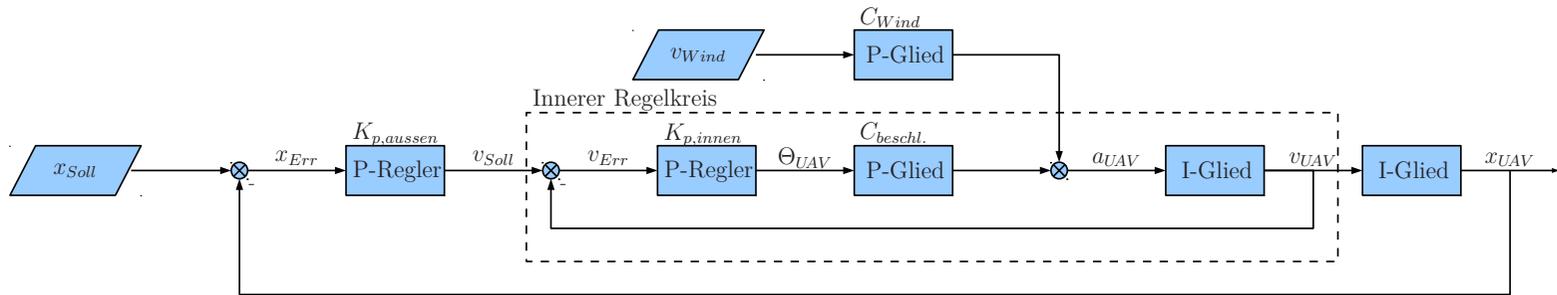


Abbildung 4.9: Blockschaltbild Störung durch Wind

Die Windgeschwindigkeit v_{Wind} erzeugt durch den Strömungswiderstandskoeffizient C_w des UAV eine Kraft F_{Wind} . Diese wiederum wirkt nach $a_{UAV} = \frac{F_{Wind}}{m_{UAV}}$ beschleunigend auf das UAV. Die Konstanten sind in $C_{Wind} = \frac{C_w}{m_{UAV}}$ zusammengefasst.

Wind ist demnach eine integrale Störgröße und betrifft den inneren Regler. Wird dieser, wie in 4.4 zunächst angenommen, unendlich stark ausgelegt, so kann Wind als Störgröße vollständig kompensiert werden. Hierzu muss davon ausgegangen werden, dass die Stellgrößenbeschränkung Θ_{max} ausreicht, um die Kraft des Windes vollständig zu kompensieren.

Da der Regler jedoch nach 4.4 ausgelegt ist, wird je nach Stärke des Windes, Strömungswiderstandskoeffizient, Masse des UAV etc. eine bleibende Regelabweichung generiert. Da diese systembedingt innerhalb der Grenzen $x_{Begr,UAV} = \pm 0,87$ m liegt, kann sie jedoch vernachlässigt werden.

5 Implementierung

Der Regler nach Abbildung 4.2 wird in das Gesamtsystem implementiert. Hierbei wird in Abschnitt 5.1 zuerst eine Simulation aufgesetzt, in welcher der Regler implementiert wird. Anschließend erfolgt die Implementierung auf der realen Hardware (Abschnitt 5.2). Bei beiden Implementierungen kommt RVIZ (Abschnitt 5.1.2) zur Visualisierung der Daten zum Einsatz. Abbildung 5.1 zeigt das schrittweise Vorgehen bei der Implementierung.

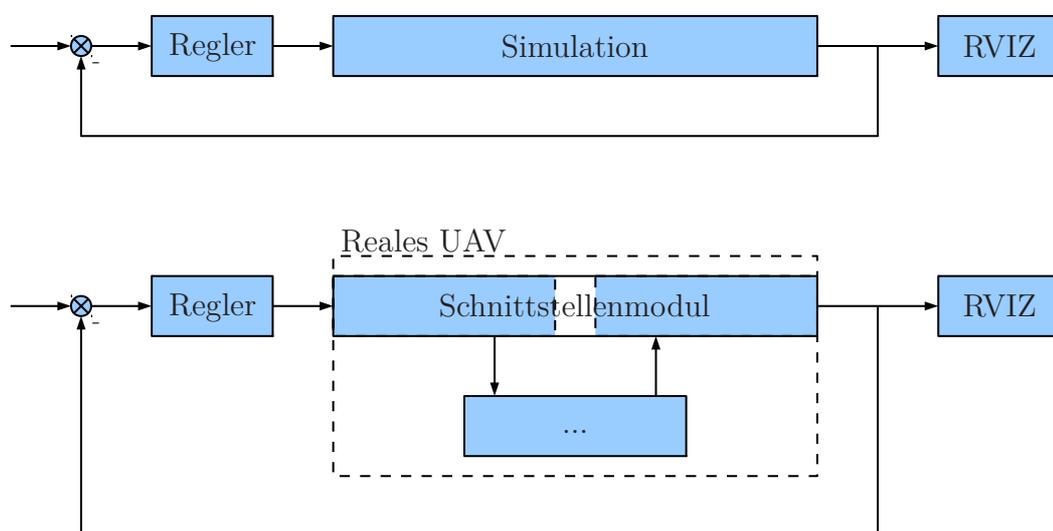


Abbildung 5.1: Vorgehensweise der Implementierung

5.1 Implementierung in der Simulation

Um ein späteres Testen der Regelung zu vereinfachen, wird zunächst eine Simulation des UAV auf einem Computer anstelle des realen UAV eingesetzt (Abschnitt 5.1.1). Diese Simulation empfängt virtuelle Eingangsdaten (Θ_{UAV} und Φ_{UAV}) und erzeugt virtuelle Ausgangsdaten (x_{UAV} , y_{UAV} und Ψ_{UAV}). Außerdem findet eine Visualisierung der Daten statt (Abschnitt 5.3). Es handelt sich bei der Simulation um ein Modell der Regelstrecke aus Kapitel 3.

5.1.1 Implementierung in ROS

Als erstes werden die Teilaufgaben der Regelung und der Simulation auf Knoten aufgeteilt. Anschließend werden Themen definiert, über welche die Knoten kommunizieren. Abbildung 5.2 zeigt die Übersicht der erstellten Knoten (blau) und die dazu gehörenden Themen (Pfeile). Zur besseren Übersicht sind die Knoten, in denen eine Koordinatentransformation stattfindet, rot markiert.

Die Funktion der einzelnen Knoten und die Definition der Nachrichteninhalte erfolgt in C++. Beispielhaft wird der Quellcode des Knotens „error_extractor_outer“ in Listing 5.1 abgebildet. Der Quellcode aller Knoten befindet sich im Anhang.

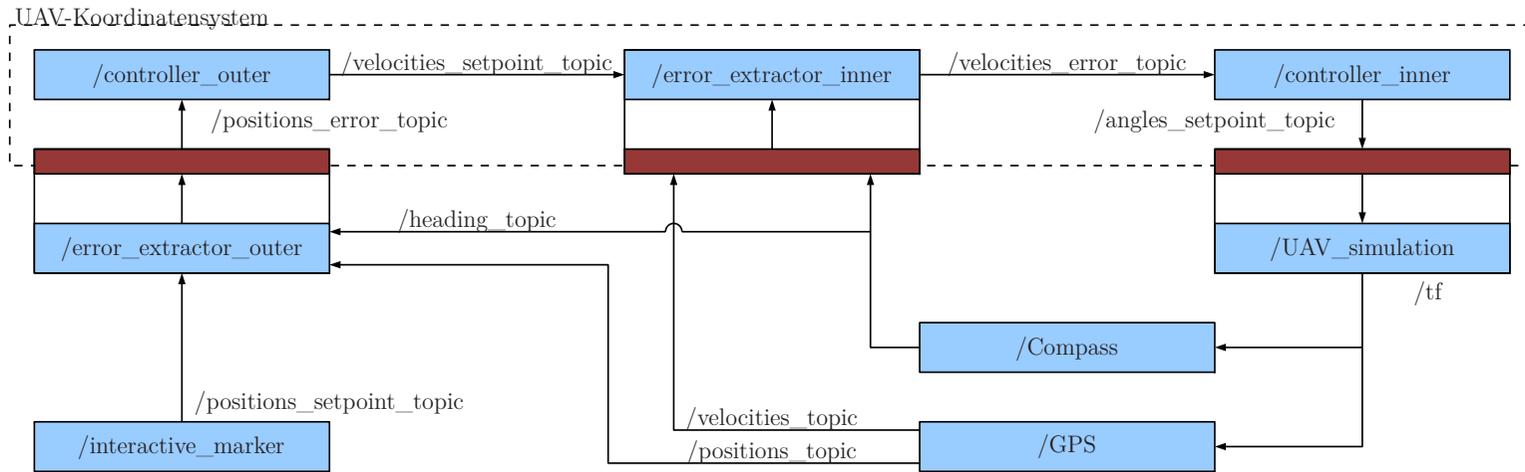


Abbildung 5.2: Übersicht Knoten und Themen

Listing 5.1: error_extractor_outer.cpp

```
#include <ros/ros.h>
#include <Simulation/positions.h>
#include <Simulation/heading.h>

Simulation::positions actualpositions;
Simulation::positions actualpositions_setpoint;
Simulation::positions actualpositions_error;
Simulation::positions actualpositions_error_transformed;
Simulation::heading actualheading;

//Parameters
float ts = 0.01;      //s

void callback_positions_topic(const Simulation::positions::ConstPtr& msg)
{
    actualpositions.x = msg->x;
    actualpositions.y = msg->y;
}

void callback_positions_setpoint_topic(const Simulation::positions::ConstPtr& msg)
{
    actualpositions_setpoint.x = msg->x;
    actualpositions_setpoint.y = msg->y;
}

void callback_heading_topic(const Simulation::heading::ConstPtr& msg)
{
    actualheading.Psi = msg->Psi;
}

int main(int argc, char** argv){
    ros::init(argc, argv, "error_extractor_outer");
    ros::NodeHandle node;
    ros::Rate rate(1/ts);
    ros::Publisher positionspublisher = node.advertise<Simulation::positions>("positions_error_topic", 1000);
    ros::Subscriber positionssubscriber = node.subscribe("positions_topic", 1000, callback_positions_topic);
    ros::Subscriber positionssubscriber_setpoint = node.subscribe("positions_setpoint_topic",
    1000, callback_positions_setpoint_topic);
    ros::Subscriber headingsubscriber = node.subscribe("heading_topic", 1000, callback_heading_topic);

    while (node.ok()){
        //calculate error
        actualpositions_error.x = actualpositions_setpoint.x - actualpositions.x;
        actualpositions_error.y = actualpositions_setpoint.y - actualpositions.y;

        //coordinate transformation
        actualpositions_error_transformed.x = actualpositions_error.x * cos(actualheading.Psi)
        + actualpositions_error.y * sin(actualheading.Psi);
        actualpositions_error_transformed.y = -actualpositions_error.x * sin(actualheading.Psi)
        + actualpositions_error.y * cos(actualheading.Psi);

        //publish
        positionspublisher.publish(actualpositions_error_transformed);

        //wait
        ros::spinOnce();
        rate.sleep();
    }
    return 0;
};
```

Die Themen bestehen aus Nachrichten, welche wiederum aus Variablen bestehen. Tabelle 5.1 zeigt den Inhalt und die Bedeutung der Variablen.

Thema	Nachricht	Variable	Bedeutung
/positions_setpoint_topic	positions	x y	Sollpos. x-Achse in Weltkoord. Sollpos. y-Achse in Weltkoord.
/positions_topic	positions	x y	Istpos. x-Achse in Weltkoord. (GPS) Istpos. y-Achse in Weltkoord. (GPS)
/positions_error_topic	positions	x y	Pos.-Fehler x-Achse in Weltkoord. Pos.-Fehler y-Achse in Weltkoord.
/velocities_setpoint_topic	velocities	x y	Sollgeschw. x-Achse in Weltkoord. Sollgeschw. y-Achse in Weltkoord.
/velocities_topic	velocities	x y	Istgeschw. x-Achse in Weltkoord. (GPS) Istgeschw. y-Achse in Weltkoord. (GPS)
/velocities_error_topic	velocities	x y	Geschw.-Fehler x-Achse in Weltkoord. Geschw.-Fehler y-Achse in Weltkoord.
/angles_setpoint_topic	angles	Theta Phi	Sollwinkel Theta in UAV-Koord. Sollwinkel Phi in UAV-Koord.
/heading_topic	heading	Psi	Istwinkel Psi in Weltkoord. (Kompass)
/tf	Transform	Vector3 Quaternion	3D-Koordinatenverschiebung (x,y,z) 3D-Koordinatenrotation (x,y,z,w)

Tabelle 5.1: Übersicht der Themen

5.1.2 RVIZ

Um die Daten der UAV-Simulation anschaulich darzustellen, findet eine Visualisierung der UAV-Simulation statt. Hierzu wird in RVIZ (3D-Visualisierumgebung) ein Modell des UAV erstellt. RVIZ abonniert alle für die Animation des UAV relevanten Themen. Abbildung 5.3 zeigt ein Bildschirmfoto der Visualisierung.

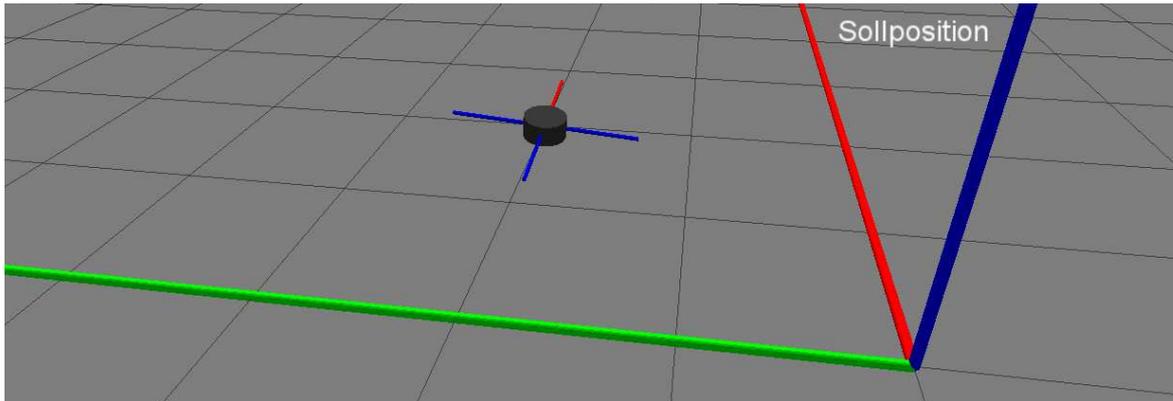


Abbildung 5.3: Visualisierung in RVIZ

5.2 Implementierung in der Realität

Die Implementierung auf dem realen UAV erfolgt analog zu der Implementierung in der Simulation. Statt einem Knoten, welcher Eingangsdaten erhält und simulierte Ausgangsdaten erzeugt, kommt hier jedoch das reale UAV zum Einsatz. Hierzu ist zum einen eine Modifikation der Hardware von Nöten (Abschnitt 5.2.1). Zum anderen wird ein Schnittstellenmodul benötigt, welches die Daten des UAV verarbeitet (Abschnitt 5.2.2).

5.2.1 Implementierung der Hardware

Wie in Abbildung 5.4 dargestellt, wird ein Bypass in den seriellen Bus gelegt (rot). So ist der auf dem Fit-PC2 laufende Algorithmus in der Lage, sowohl mit NaviCtrl als auch mit FlightCtrl bidirektional zu kommunizieren. Interagiert der Computer mit FlightCtrl, leitet die NaviCtrl alle an UART 2 ankommenden Daten an UART 1 weiter und umgekehrt. Vergleichend hierzu wird auf den Informationsfluss vor dieser Arbeit (Abbildung 2.2) verwiesen.

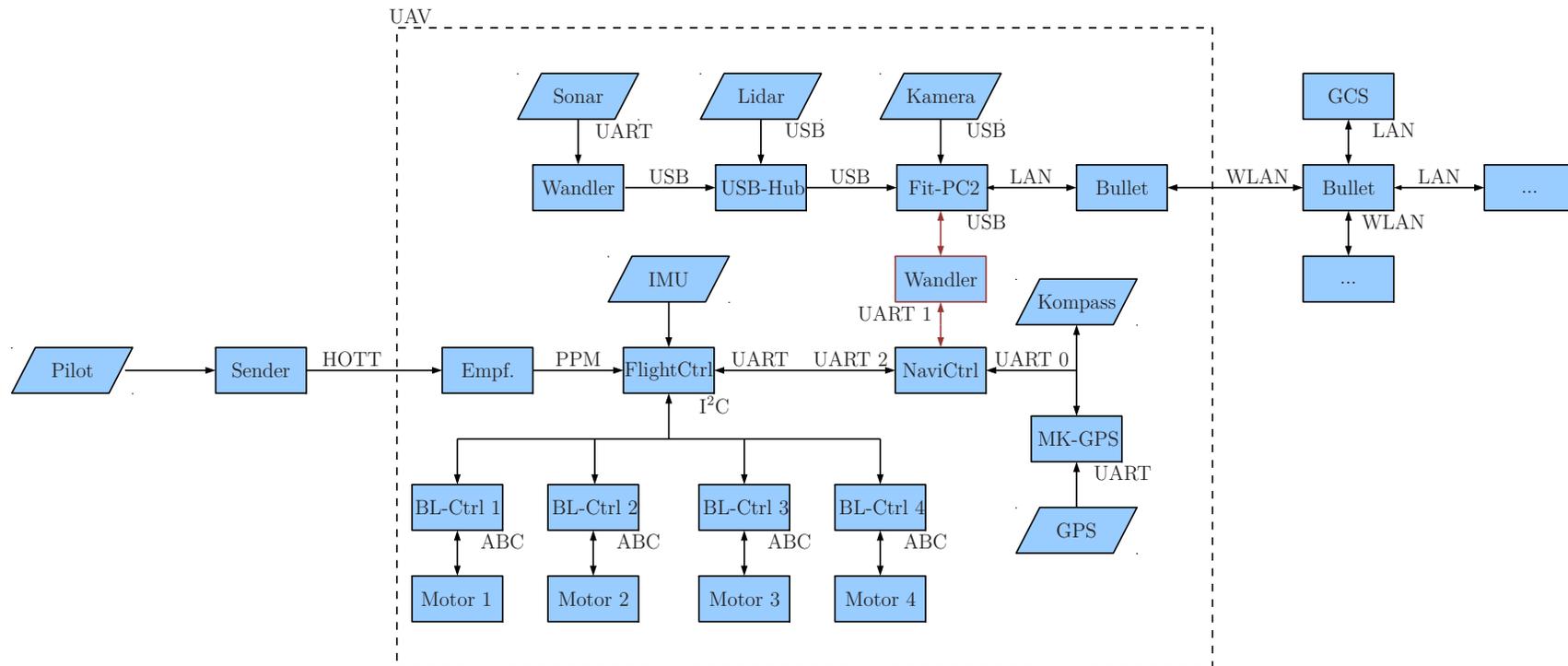


Abbildung 5.4: Informationsfluss nach der Implementierung

5.2.2 Implementierung der Software

Um eine Kommunikation mit dem bestehenden System aufbauen zu können, wird ein Schnittstellenmodul geschrieben. Dieses dient als bidirektionaler Übertrager zwischen der seriellen Schnittstelle der FlightCtrl und ROS. Das Modul empfängt Nachrichten, welche über die serielle Schnittstelle gesendet werden, interpretiert diese und sendet sie nach Anpassung (Einheitenumrechnungen, Normalisierungen etc.) als ROS Thema. Gleichzeitig abonniert das Modul bestimmte ROS Themen und schickt diese angepasst über die serielle Schnittstelle an die FlightCtrl. Hierdurch ist es in ROS möglich, zum Beispiel Sollwinkel vorzugeben und die aktuelle Geschwindigkeit zurückzulesen. Welche Nachrichten das Modul unterstützt hängt von der zentralen Komponente, dem Parser, ab. Eine Auswahl, für diese Arbeit relevanter Nachrichten, ist in Tabelle 5.2 dargestellt.

FlightCtrl → ROS	ROS → FlightCtrl
Θ_{UAV}	$\Theta_{Soll,UAV}$
Φ_{UAV}	$\Phi_{Soll,UAV}$
Ψ_{UAV}	
GPS Latitude	
GPS Longitude	
GPS Geschwindigkeit über Grund	
GPS Bewegungsrichtung über Grund	

Tabelle 5.2: Auswahl an Parser Nachrichten

Die Möglichkeiten, welche der Parser im Detail bietet und welche Themen er abonniert, sind im Anhang zu sehen. Das Schnittstellenmodul wurde in dieser Arbeit nicht von Grund auf neu geschrieben, sondern orientiert sich an vorhergehenden Arbeiten.

6 Evaluation

Der Reglerentwurf basiert auf einem Modell, welches durch physikalische Grundgleichungen hergeleitet und durch experimentell ermittelte Parameter vervollständigt wurde. Aus diesem Grund wird zuerst im Abschnitt 6.1 evaluiert, inwieweit das Modell das reale UAV abbildet. Anschließend wird die Positionsregelung zum einen in einer Simulation (Abschnitt 6.2) und zum anderen am realen UAV (Abschnitt 6.3) evaluiert. Hierbei findet in der Simulation ein „Proof of Concept“ statt. Am realen Modell wird insbesondere die Parametrisierung evaluiert. Die Regelung wird hier mit dem bestehenden System verglichen. Abbildung 6.1 visualisiert das schrittweise Vorgehen bei der Evaluation.

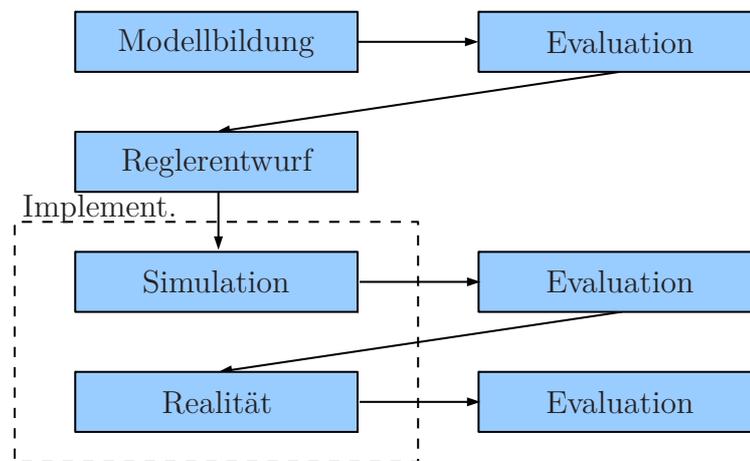


Abbildung 6.1: Vorgehensweise der Evaluation

6.1 Modell

Um zu zeigen, dass das Modell sowohl in Struktur als auch in Parametern mit der Realität in akzeptablem Maße übereinstimmt, werden weitere Versuche im Motion-Capturing-Labor durchgeführt. Das UAV wird im Schwebeflug aus der Neutralen in x-Richtung so ausgelenkt, dass es die in Abbildung 6.3 abgebildete Trajektorie beschreibt.

Parallel hierzu wird das parametrisierte Modell aus Abbildung 6.2 berechnet¹.

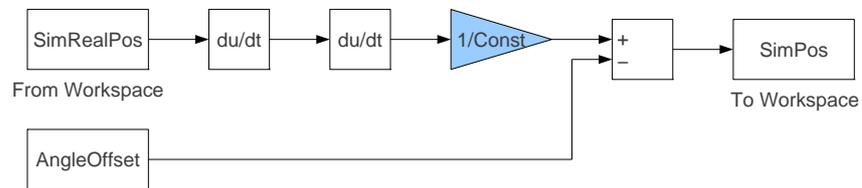


Abbildung 6.2: Simulinkmodell zur Evaluation des Modells

Abbildung 6.4 zeigt das Ergebnis des Versuchs.

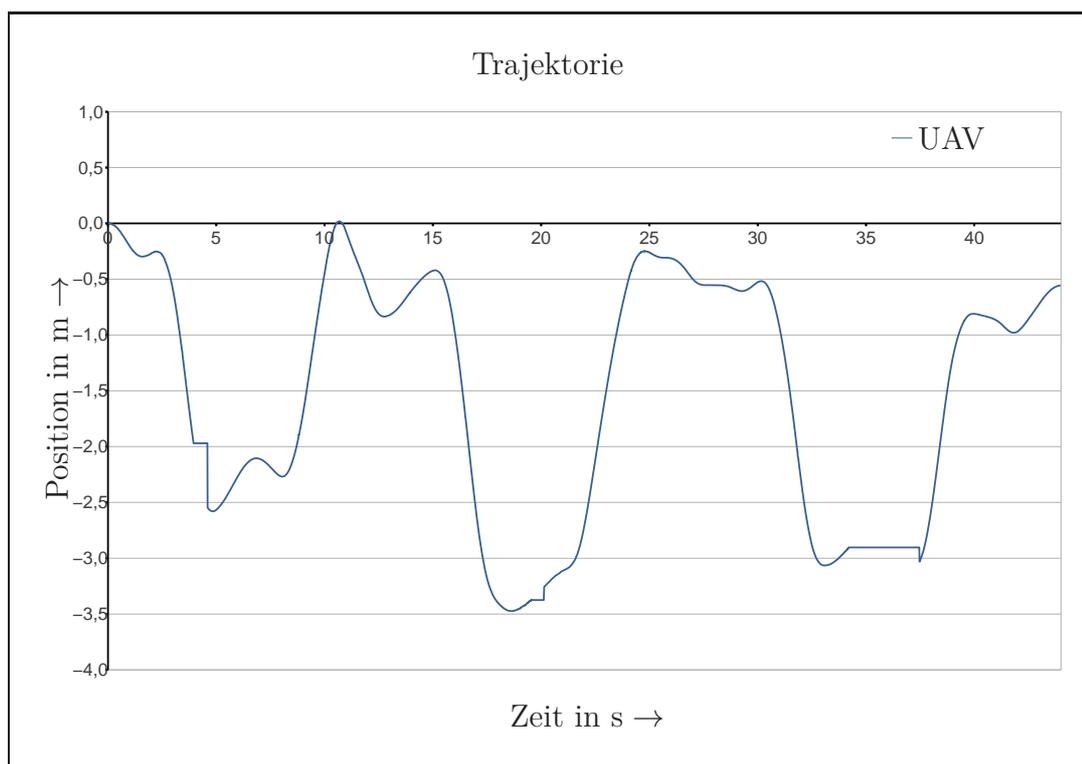


Abbildung 6.3: Trajektorie des realen UAV

¹Abtastrate: 6,67 Hz

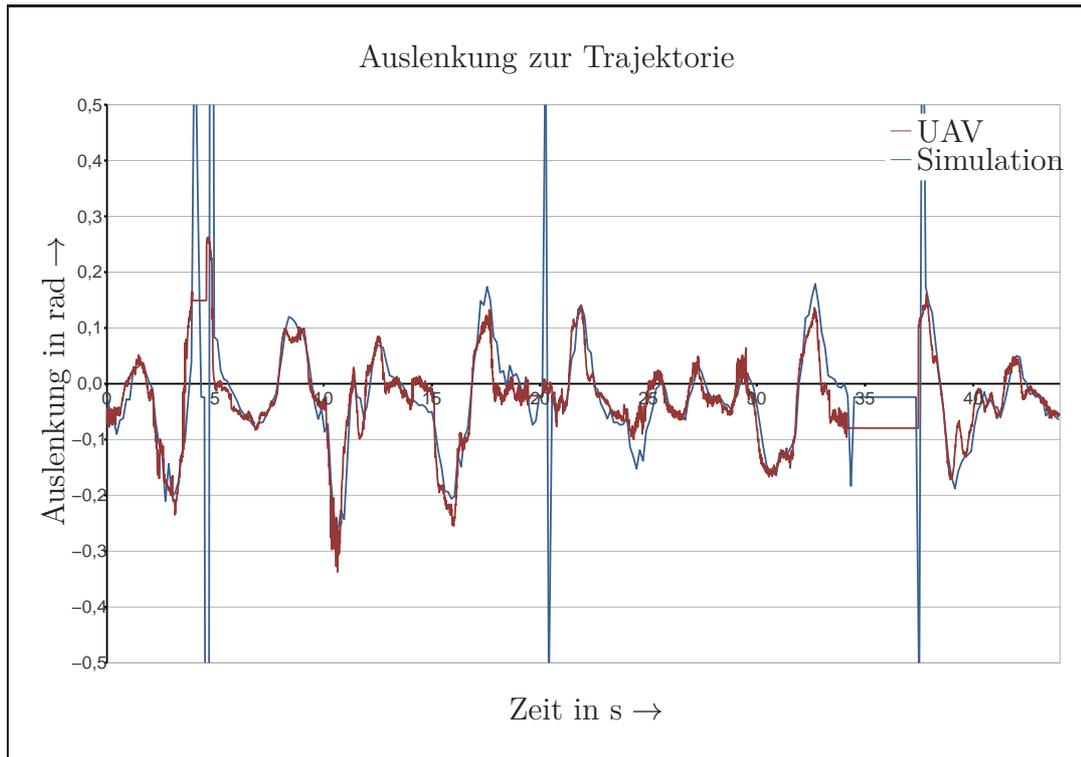


Abbildung 6.4: Winkel des realen UAV und des Modells

Es zeigt sich, dass das Modell sehr stark mit dem realen UAV korreliert. Große Spitzenwerte treten dann auf, wenn das UAV aus dem Messbereich aus- und wieder eintritt. Da die Eintrittsstelle nicht genau die Austrittsstelle ist, ist das Positionssignal an dieser Stelle nicht differenzierbar. Die Spitzenwerte begründen sich demnach durch das Messsystem und haben keine Bedeutung für die Qualität des Modells.

6.2 Simulation

Wie im Kapitel 5.1 erwähnt wird eine Simulation des UAV in ROS implementiert. Das UAV wird hier, wie im Kapitel 3.3 Modellbildung hergeleitet, modelliert. Die Schnittstellen zum Modell entsprechen denen des realen UAV (Eingang: Winkel Θ_{UAV}, Ψ_{UAV} , Ausgang: Positionen x_{Welt}, y_{Welt}).

6.2.1 Versuchsaufbau

Um das System zu evaluieren wird der in Abbildung 6.5 dargestellte Testfall konstruiert.

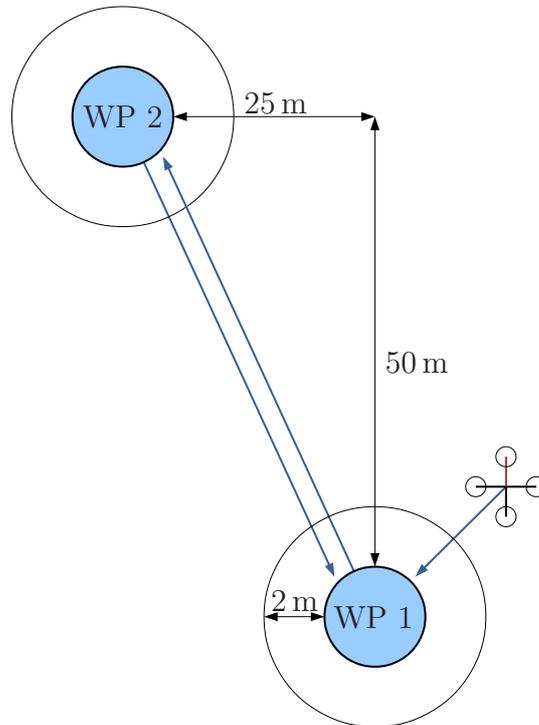


Abbildung 6.5: Versuchsaufbau

Das UAV fliegt von einem Startpunkt aus zuerst einen Wegpunkt (WP 1) an. Befindet es sich in einem Radius von 2 m um den Punkt, wird dieser als erreicht deklariert. Dann fliegt es zu einem 56 m entfernten WP 2 und wieder zurück zu WP 1. Durch Hin- und Rückflug werden nur in eine Richtung wirkende Störgrößen sichtbar (z.B. Wind). Das UAV behält während des gesamten Versuchs eine Ausrichtung von $\Psi_{UAV} = 0^\circ \pm 10^\circ$. Es ist demnach eine Bewegung sowohl in x-Richtung als auch in y-Richtung im UAV Koordinatensystem nötig, um zum Ziel zu kommen.

6.2.2 Durchführung

Diagramm 6.6 zeigt die eindimensionale Sprungantwort der Regelung bei $\Theta_{max} = \pm 1^\circ$. Diagramm 6.7 zeigt analog hierzu dasselbe Verhalten bei $\Theta_{max} = \pm 3^\circ$. Alle Größen werden in Weltkoordinaten dargestellt.

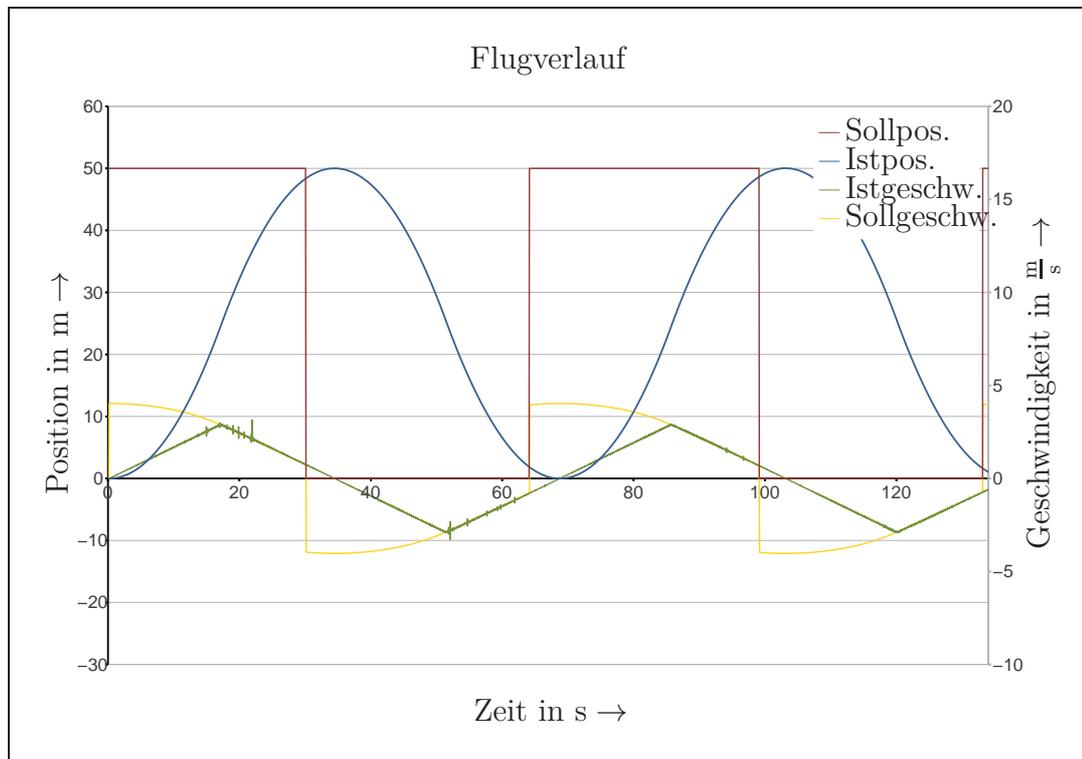


Abbildung 6.6: Simulierter Testflug mit neuer Positionsregelung (X , $\Theta_{max} = \pm 1^\circ$)

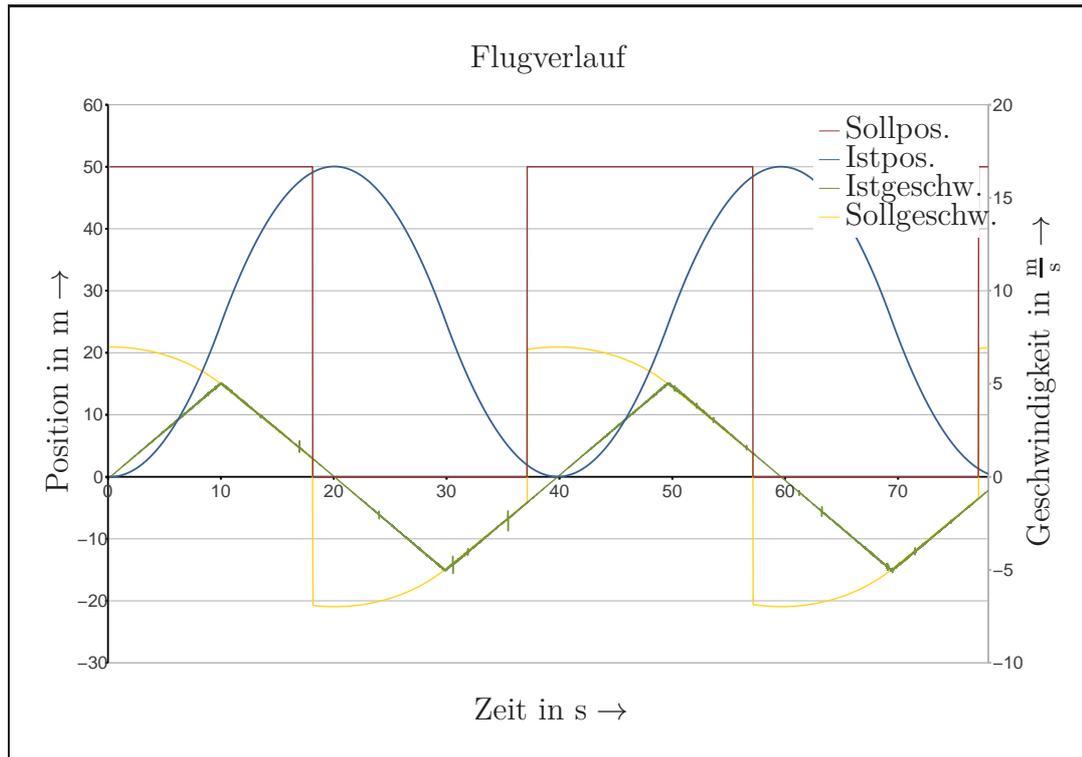


Abbildung 6.7: Simulierter Testflug mit neuer Positionsregelung (X , $\Theta_{max} = \pm 3^\circ$)

Diagramm 6.8 zeigt den Flugverlauf in einem XY-Diagramm. Dieser ist unabhängig von Θ_{max} und deshalb bei allen Versuchen identisch.

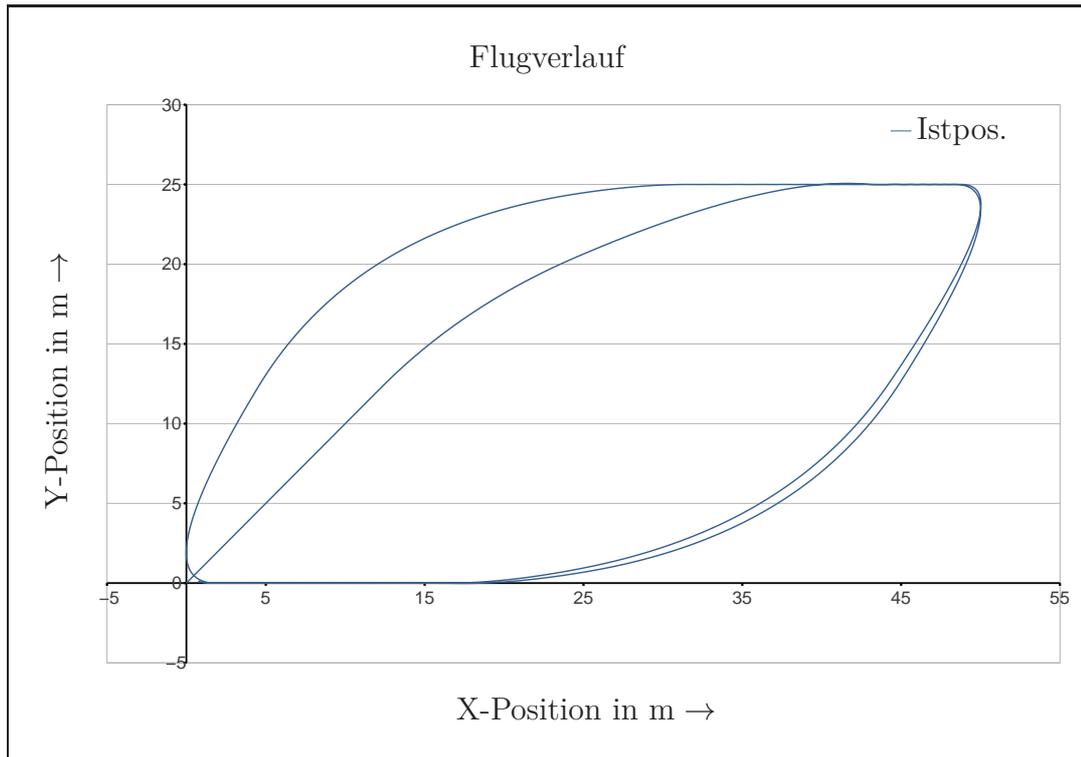


Abbildung 6.8: Simulierter Testflug mit neuer Positionsregelung (XY)

6.2.3 Ergebnisse

Die Simulation zeigt die in Tabelle 6.1 für die Bewertung des Reglers charakteristischen Eigenschaften.

Θ_{max}	Anregelzeit	Stat. Fehler	Überschwingen
1°	30,1 s	0,0 m	0,0 m
3°	18,1 s	0,0 m	0,0 m

Tabelle 6.1: Eigenschaften der Regelung (Simulation)

Dies entspricht dem in Kapitel 4 als optimal definierten Verhalten. Insbesondere zeigt die Simulation, dass es bei der Implementierung keine fundamentalen Fehler (Vorzeichenfehler, Vertauschung der Achsen, Skalierungsfehler etc.) gibt.

Die Trajektorie des Flugs ist oval. Dies ist dann der Fall, wenn das UAV sowohl eine Bewegung in x-Richtung (x_{UAV}) als auch y-Richtung (y_{UAV}) durchführen muss, um zum Ziel zu kommen. Das UAV fliegt dann nicht auf dem direkten, jedoch auf dem

regelungstechnisch schnellsten Weg zum Ziel. Auf diese Eigenschaft wird in Kapitel 7.2 weitergehend eingegangen.

6.3 Realität

Nach der Evaluation in der Simulation wird der Algorithmus auch auf dem realen System evaluiert.

6.3.1 Versuchsaufbau

Der Versuchsaufbau ist derselbe wie in Abschnitt 6.2.1 beschrieben. Alle Versuche werden mit den in Tabelle 6.2 beschriebenen Randbedingungen durchgeführt. Als Testobjekt kommt nicht das UAV aus den Versuchen im Motion-Capturing-Labor (Abbildung 3.7) zum Einsatz, sondern das NIFTi-UAV welches in Abbildung 2.1 zu sehen ist. Dies soll eine Evaluation im realen NIFTi-Einsatz ermöglichen.

Umweltbedingung	Wert
Windgeschwindigkeit	$3,0 \frac{\text{m}}{\text{s}}$
Temperatur	$28,0 \text{ }^\circ\text{C}$
Luftfeuchte	$52,6 \%$
Luftdruck	$1028,5 \text{ hPa}$

Tabelle 6.2: Randbedingungen der realen Evaluation

6.3.2 Durchführung

Diagramme 6.9, 6.10 und 6.11 zeigen den Verlauf der Position und die Geschwindigkeiten bei Testläufen der x-Richtung (Weltkoordinatensystem). Der Maximalwinkel ist auf $\Theta_{max} = \pm 2,0^\circ$ bzw. $\Theta_{max} = \pm 3,0^\circ$ und $\Theta_{max} = \pm 5,0^\circ$ festgelegt.

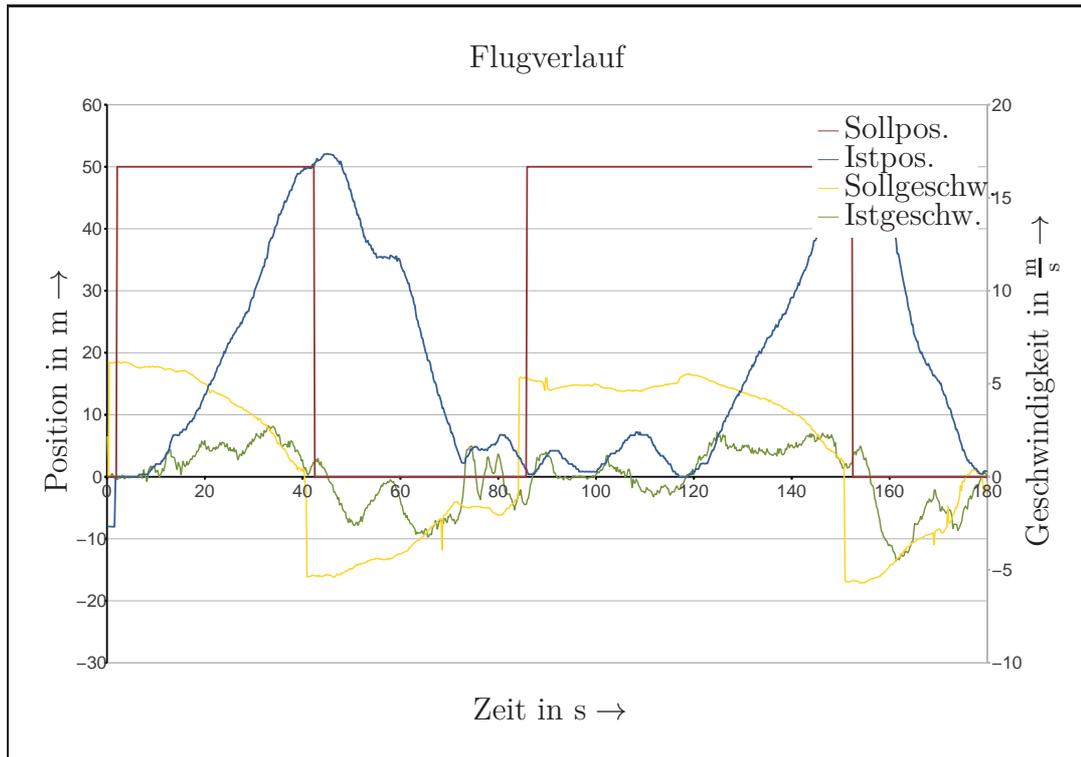


Abbildung 6.9: Testflug mit neuer Positionsregelung ($X, \Theta_{max} = \pm 2^\circ$)

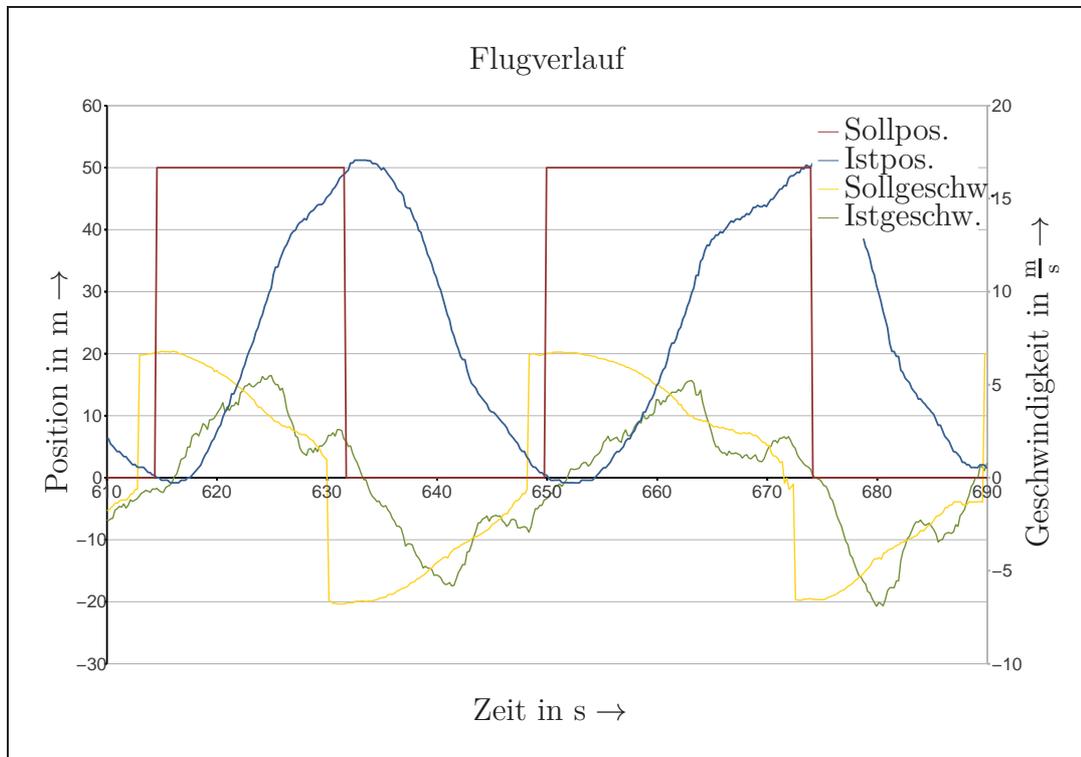


Abbildung 6.10: Testflug mit neuer Positionsregelung ($X, \Theta_{max} = \pm 3^\circ$)

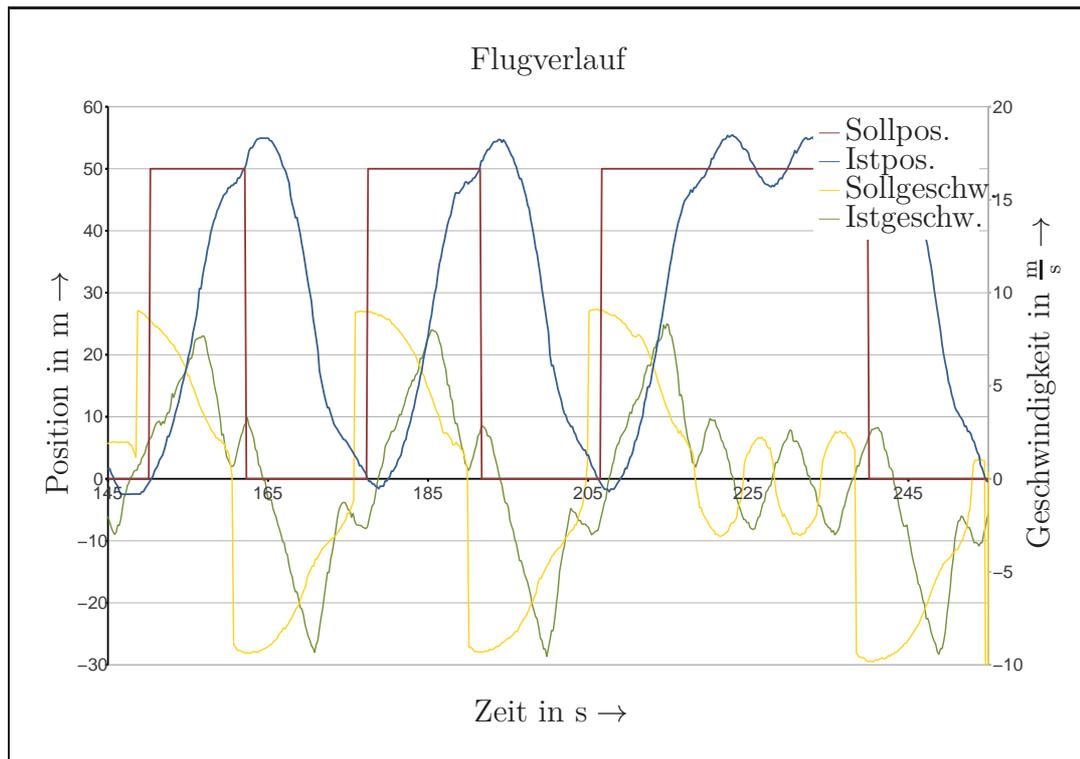


Abbildung 6.11: Testflug mit neuer Positionsregelung (X , $\Theta_{max} = \pm 5^\circ$)

Derselbe Versuch wird mit der bestehenden Positionsregelung (siehe Abschnitt 2.2) durchgeführt. Der Flugverlauf ist in Abbildung 6.12 zu sehen.

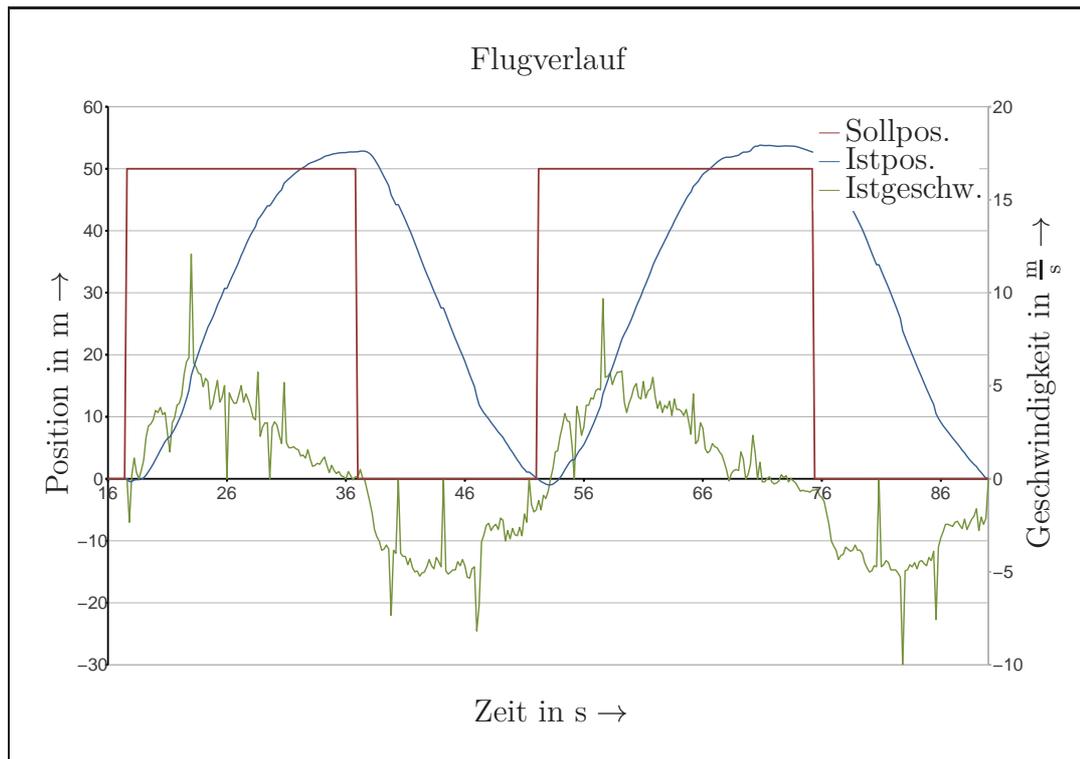


Abbildung 6.12: Testflug mit alter Positionsregelung (X)

Die Trajektorie, die das UAV hierbei fliegt, ist in Abbildungen 6.13, 6.14 und 6.15 zu sehen. Die Trajektorie der alten Positionsregelung zeigt Abbildung 6.16.

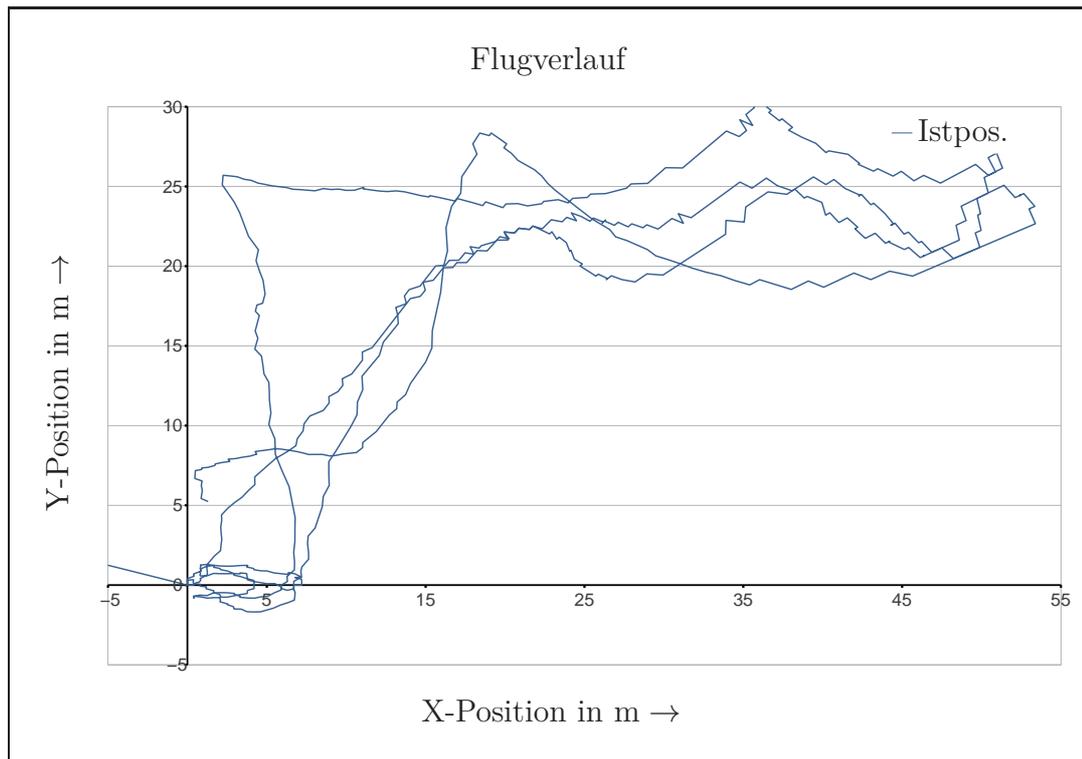


Abbildung 6.13: Testflug mit neuer Positionsregelung ($XY, \Theta_{max} = \pm 2^\circ$)

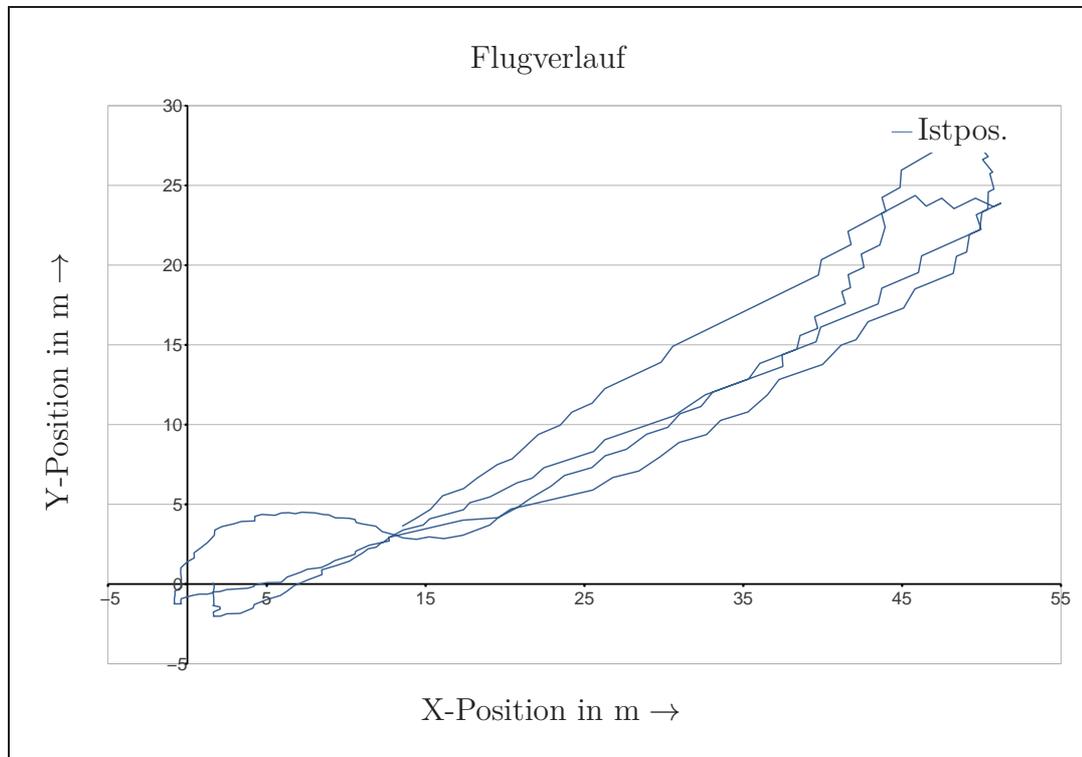


Abbildung 6.14: Testflug mit neuer Positionsregelung ($XY, \Theta_{max} = \pm 3^\circ$)

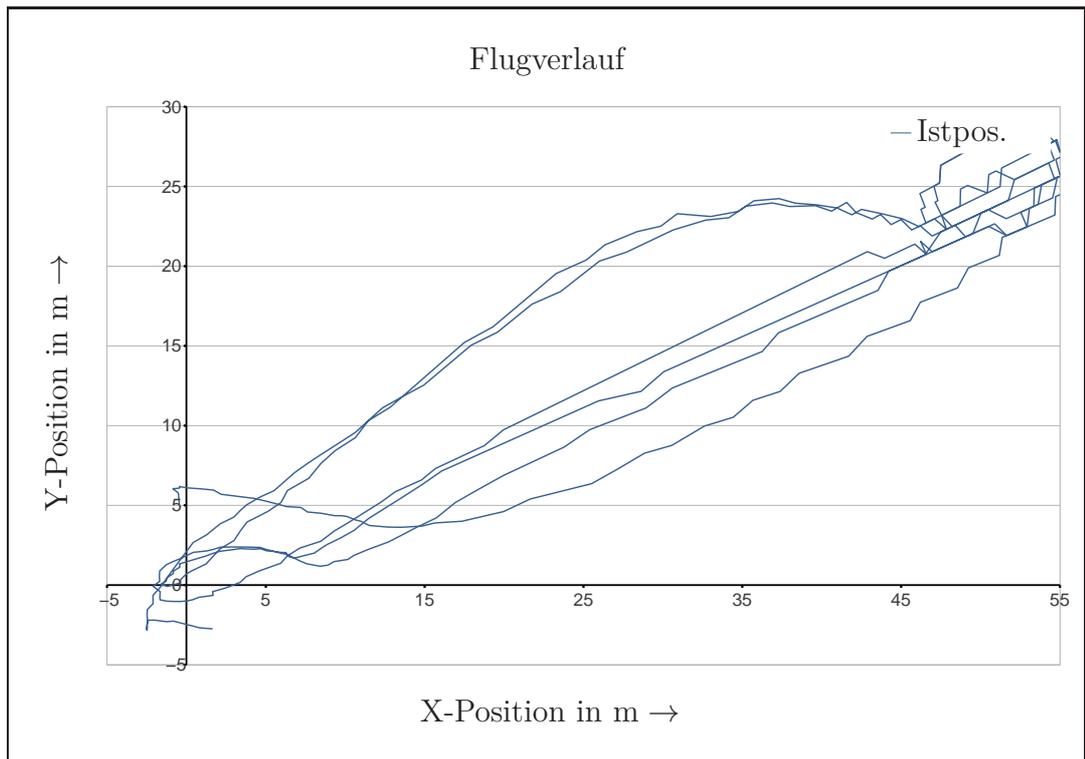


Abbildung 6.15: Testflug mit neuer Positionsregelung ($XY, \Theta_{max} = \pm 5^\circ$)

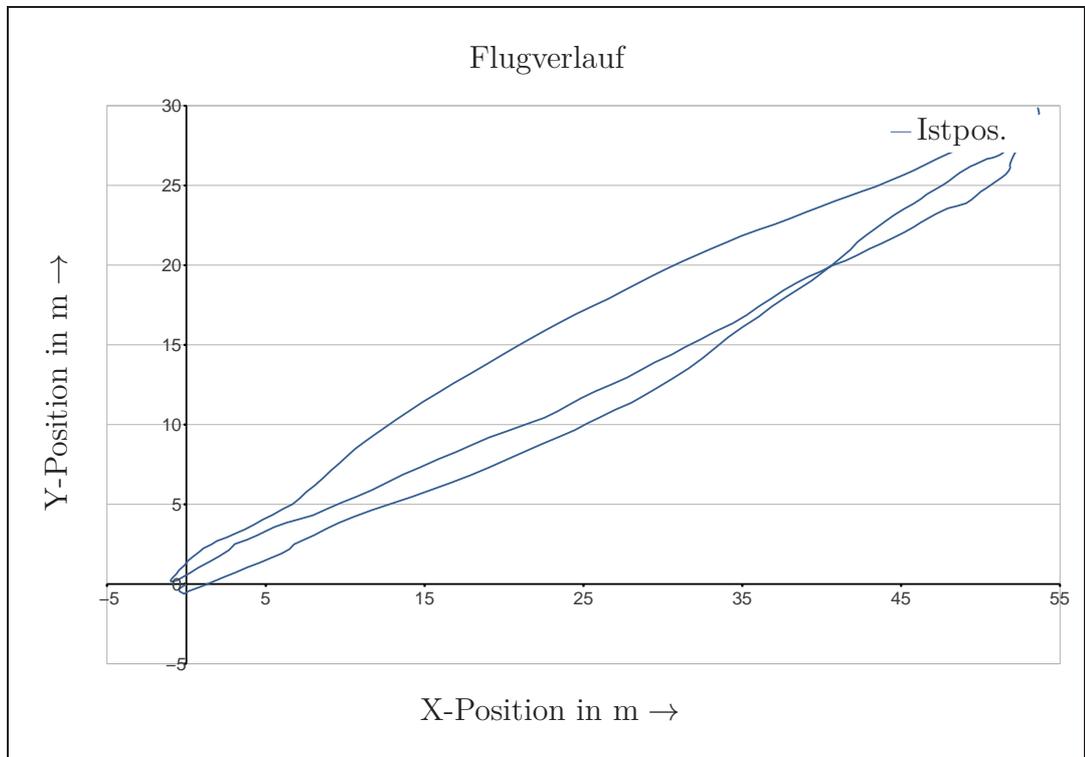


Abbildung 6.16: Testflug mit alter Positionsregelung (XY)

6.3.3 Ergebnisse

Kennwerte der Regelung im realen Versuch sind in Tabelle 6.3 zu sehen.

Regelung	Θ_{max}	Anregelzeit	Überschwingen
Neue Regelung	2°	45 s	1,3 m
Neue Regelung	3°	18 s	0,7 m
Neue Regelung	5°	14 s	3,3 m
Alte Regelung	-	18 s	2,2 m

Tabelle 6.3: Eigenschaften der Regelung (Realität)

Während die Regelung sowohl in Bezug auf Geschwindigkeit als auch auf Genauigkeit in der Simulation überragende Ergebnisse erzielt (vgl. Abschnitt 6.2.3), ist die Leistungsfähigkeit der Regelung im realen Versuch eingeschränkt. Dies liegt zum einen am schlechteren Feedbacksignal und der hiermit verbundenen Phasenverschiebung von Ist- und Sollsignal (vgl. Abschnitt 4.5.1). Zum anderen treten in der Realität immer Störgrößen auf. Dies betrifft sowohl Positions- als auch Geschwindigkeitsinformation. Als Hauptstörgröße ist dies vor allem Wind. Aber auch parasitäre Einflüsse, z.B. der Höhenregelung, beeinflussen das Ergebnis negativ. Ansätze zur Verbesserung des Feedbacksignals werden im Ausblick aufgezeigt.

Wie zu erwarten, ist die Regelung mit größerem Θ_{max} schneller, neigt aber auch zu Überschwingen. Es fällt jedoch auf, dass die Regelung mit $\Theta_{max} = \pm 3^\circ$ weniger Überschwingen zeigt, als die Regelung mit $\Theta_{max} = \pm 2^\circ$. Wie die unsymmetrische Sprungantwort zeigt, liegt dies höchstwahrscheinlich am Wind, welcher das UAV in eine Richtung drückt. Auch die zugehörige Trajektorie bestätigt dies.

Es zeigt sich außerdem, dass die Beschleunigungskonstante C_{beschl} nicht exakt mit der in Kapitel 3.1.4 identifizierten übereinstimmt. Der Geschwindigkeitsverlauf beweist, dass die Konstante bei dem hier verwendeten UAV größer sein muss als die angenommene. Im Vergleich zur bereits auf dem UAV vorhandenen Positionsregelung zeigt insbesondere der Versuch mit $\Theta_{max} = \pm 3^\circ$ ein besseres Einschwingverhalten. Zwar ist die Anregelzeit identisch, der hier entwickelte Algorithmus schwingt jedoch deutlich weniger über.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Gegenstand dieser Arbeit war die Entwicklung einer Positionsregelung eines UAV. Zunächst wurde die Arbeit in den Kontext des Projekts NIFTi eingeordnet und die Motivation für die Arbeit vorgestellt. Es handelte sich bei dem beschriebenen UAV um einen flugfähigen Quadrocopter mit bereits implementierter Lageregelung, GPS Hardware und Onboard-Computer.

Um die Positionsregelung zu entwickeln, wurde zunächst ein qualitatives grey-box-Modell des UAV auf Basis von physikalischen Gleichungen aufgebaut. Hierbei wurde schrittweise vom eindimensionalen Modell mit zwei Freiheitsgraden bis zum dreidimensionalen Modell mit sechs Freiheitsgraden vorgegangen. Es wurden Annahmen getroffen, auf deren Basis das Modell vereinfacht wurde. Anschließend wurden die Modellparameter identifiziert. Das qualitative Modell wurde hiermit in ein quantitatives überführt.

Das Modell wurde analysiert und eine für das Modell optimale Reglerstruktur gewählt. Der Regler wurde aufgrund physikalischer Überlegungen parametrisiert. Das Führungs- und Störverhalten der Regelung, sowie das Verhalten bei Parameterunsicherheiten wurden untersucht.

Der Regler wurde in ROS, einem Betriebssystem für Roboter, auf dem UAV implementiert. Die Schnittstelle zu bereits vorhandener Software wurde hergestellt.

Es folgte zunächst eine Evaluation des Modells. Anschließend wurde der Regler sowohl in der Simulation als auch in der Realität evaluiert. Hierauf folgt diese Zusammenfassung und ein Ausblick über noch zu bearbeitende Aufgaben. Es werden Verbesserungsvorschläge gegeben.

7.2 Ausblick

Im Kapitel 4.4 wird die maximale Auslenkung des UAV auf $\Theta_{max} = \pm 5^\circ$ festgelegt. Weitere Parameter werden von diesem Wert abgeleitet. Aus diesem Grund eignet sich der Parameter gut, um eine Bewertung des Feedbacksystems durchzuführen. Ist beispielsweise der Empfang vieler GPS-Satelliten möglich, kann der Wert erhöht werden. Des Weiteren kann der Wert weiter gesteigert werden, wenn ein zusätzlicher Sensor (Lidar, Sonar etc.) als Positionsfeedback zur Verfügung steht.

Mit der Steigerung von Θ_{max} steigt die Leistungsfähigkeit der Regelung, da eine geringere Stellgrößenbeschränkung größere Beschleunigungen zulässt. Führungs- und Störverhalten werden verbessert. In Kapitel 3.1.2 werden jedoch Vereinfachungen am Modell vorgenommen, welche nicht allgemeingültig sind. Insbesondere wird angenommen, dass das UAV nur kleine Auslenkungen aus der Ruhelage durchführt und die Höhenregelung keinen Einfluss auf die Positionsregelung hat. Bei großen Auslenkungen muss der Einfluss der Positionsregelung auf die Höhenregelung (und umgekehrt) in die Berechnung der Stellgröße einfließen. Dies kann zum Beispiel über eine nichtlineare Vorsteuerung erfolgen. Auch die Berechnung der Beschleunigung kann dynamisch linearisiert und vorgesteuert werden.

In Kapitel 4.5.1 wird gezeigt, dass zur Geschwindigkeitsschätzung ein im GPS integrierter Beobachter verwendet wird. Dieser ist jedoch nicht auf den hier vorliegenden Fall optimal parametrisiert. Besser ist es, die Geschwindigkeit über das ohnehin zur Verfügung stehende Modell des UAV zu rekonstruieren. Durch Fusion der IMU-Daten mit den GPS-Daten kann auch die Position genauer geschätzt werden. So kann eine vollständige Zustandsschätzung des Modells erfolgen.

In Kapitel 6.2.3 zeigt sich, dass das UAV nicht auf einer direkten Linie vom Anfangs- zum Endpunkt fliegt. Stellt dies für die akute Bewegung ein Problem dar, müssen mehrere Zwischenpunkte auf der direkten Verbindung eingefügt werden. Das UAV wird dann nur noch kleine Abweichungen von der Bahn zeigen. Auch ein Drehen des UAV, sodass für die Bewegung nur noch eine einachsige Bewegung von Nöten ist, ist möglich.

Literaturverzeichnis

- [Beu11] BEUL, Marius: *Untersuchung und Modellbildung einer Piezoaktorik zur Fehlerkompensation von Industrierobotern*, Hochschule Bonn-Rhein-Sieg, Diplomarbeit, 2011
- [Brä04] BRÄUNLING, Willi J.: *Flugzeugtriebwerke: Grundlagen, Aero- Thermodynamik, Kreisprozesse, Thermische Turbomaschinen, Komponenten- und Emissionen*. VDI-Buch, 2004
- [BS07] BOUABDALLAH, Samir ; SIEGWART, Roland: Full Control of a Quadrotor. In: *International Conference on Intelligent Robots and Systems*, 2007
- [KJK⁺12] KRUIJFF, G.J.M. ; JANÍČEK, M. ; KESHAVDAS, S. ; LAROCHELLE, B. ; ZENDER, H. ; SMETS, N.J.J.M. ; MIOCH, T. ; NEERINCX, M.A. ; DIGGELEN, J. v. ; COLAS, F. ; LIU, M. ; POMERLEAU, F. ; HLAVÁČ, V. ; SVOBODA, T. ; PETRIČEK, T. ; ZIMMERMAN, K. ; PIRRI, F. ; GIANNI, M. ; PAPADAKIS, P. ; SINHA, A. ; BALMER, P. ; TOMATIS, N. ; WORST, R. ; LINDER, T. ; SURMANN, H. ; TRETYAKOV, V. ; CORRAO, S. ; PRATZLER-WANCZURA, S. ; SULK, M.: Experience in System Design for Human-Robot Teaming in Urban Search & Rescue / DFKI GmbH, TNO, ETH Zürich, CTU Prague, ROMA, BLUE, Fraunhofer, VVFF und FDDO. 2012. – Forschungsbericht
- [KTL⁺12] KRUIJFF, G.J.M. ; TRETYAKOV, V. ; LINDER, T. ; PIRRI, F. ; GIANNI, M. ; PAPADAKIS, P. ; PIZZOLI, M. ; SINHA, A. ; PIANESE, E. ; CORRAO, S. ; PRIORI, F. ; FEBRINI, S. ; ANGELETTI, S.: Rescue Robots at Earthquake-Hit Mirandola, Italy: a Field Report / DFKI GmbH, Fraunhofer IAIS, University of Rome La Sapienza und Vigili del Fuoco. 2012. – Forschungsbericht
- [Lun11] LUNZE, Jan: *Regelungstechnik 1*. Springer, 2011

- [NIF13] *NIFTi Extranet*. <http://www.nifti.eu>, 2013. – [Online; Zugriff 26-August-2013]
- [Ope10] *OpenPilot CopterControl Navigation Stack*. <http://git.openpilot.org/browse/OpenPilot/flight/Modules/Guidance/guidance.c?hb=true>, 2010. – [Online; Zugriff 10-Mai-2013]
- [PKJ12] PFEIFER, Erick ; KASSAB JR., Fuad: Dynamic Feedback Controller of an Unmanned Aerial Vehicle. In: *Brazilian Robotics Symposium and Latin American Robotics Symposium*, 2012
- [PKKH09] PULS, T. ; KEMPER, M. ; KÜKE, R. ; HEIN, A.: GPS-based Position Control and Waypoint Navigation System for Quadcopters. In: *International Conference on Intelligent Robots and Systems*, 2009
- [PPV12] PATEL, Ankit R. ; PATEL, Mahesh A. ; VYAS, Dhaval R.: Modeling and Analysis of Quadrotor using Sliding Mode Control. In: *44th IEEE Southeastern Symposium on System Theory*, 2012
- [PX413] *PX4-position control dokumentation diagramm*. <https://github.com/PX4/Firmware/tree/master/Documentation>, 2013. – [Online; Zugriff 15-März-2013]
- [ROS13] *Documentation - ROS Wiki*. <http://www.ros.org/wiki/>, 2013. – [Online; Zugriff 15-März-2013]
- [Sur13] SURMANN, Hartmut: *NIFTi - Mobile Roboter sollen Rettungskräfte im Einsatz unterstützen*. <http://www.iais.fraunhofer.de/nifti.html>, 2013. – [Online; Zugriff 15-März-2013]
- [TWW10] TRETYAKOV, Viatcheslav ; WINZER, Jochen ; WORST, Rainer: Platform manufacturing and sensor integration (UAV) / Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS. 2010. – Forschungsbericht
- [u-b11] *u-blox LEA-6 GPS Modules Data Sheet*. 2011

8 Anhang

Alle Anlagen befinden sich im digitalen Format, nach Kapiteln sortiert, auf einer DVD. Die Rechte an der Software liegen beim Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS. Eine Einsicht erfolgt nur in Rücksprache mit den verantwortlichen Projektleitern.

Ansprechpartner:

Dipl.-Inform. Rainer Worst

Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS

Schloss Birlinghoven

53757 Sankt Augustin

Tel.: +49 2241/14 - 2281

Fax.: +49 2241/14 - 42281

E-Mail: rainer.worst@iais.fraunhofer.de

Eidesstattliche Erklärung

Hiermit versichere ich, B. Eng. Marius Beul, dass die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel von mir verfasst wurde. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Bonn, 31. August 2013

B. Eng. Marius Beul