



Fachbereich Informatik und Medien

DIPLOMARBEIT

Kameragestützte Feinpositionierung eines Quadropters

Vorgelegt von: **Gregor Landmann**
am: **7. März 2013**

zum Erlangen des akademischen Grades

Diplom-Informatiker (FH)

Erstgutachter: Prof. Dr. sc. techn. Harald Loose
Zweitgutachter: Dipl.-Inf. Ingo Boersch

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst habe. Es wurden keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt. Die wörtlich oder sinngemäß übernommenen Zitate sind als solche kenntlich gemacht.

.....
Ort, Datum

.....
Unterschrift

Danksagung

Ich möchte hiermit all denen danken, die direkt oder indirekt an der Entstehung dieser Arbeit beteiligt waren. Dies gilt insbesondere für meine Freundin Olivia Beyer, die das Korrekturlesen übernahm und mich während der ganzen Zeit unterstützt und entlastet hat. Auch möchte ich meiner Familie und meinen Freunden dafür danken, dass sie stets Gedult und Verständniss aufbrachten. Der Fachhochschule Brandenburg und dem Sportbeauftragten Robert Ballmann danke ich für das Bereitstellen der Sporthalle. Ein Dank geht auch an Herrn Prof. Loose und Herrn Boersch für das Betreuen dieser Diplomarbeit. Zu guter Letzt danke ich der gesamten MikroKopter Community, ohne die diese Arbeit niemals hätte entstehen können.

Inhaltsverzeichnis

1	Einleitung	9
1.1	Thema und Abgrenzung	10
1.2	Aufbau der Arbeit	10
2	Allgemeine Grundlagen	11
2.1	Drehflügler	11
2.1.1	Definition	11
2.1.2	Eigenschaften	11
2.1.3	Giermomentenausgleich	12
2.1.4	Steuerung	12
2.2	Quadrocopter	14
2.2.1	Realisierung der Steuerung	15
2.2.2	Stabilisierung	17
3	Analyse der Systemumgebung	19
3.1	Allgemeines	20
3.2	Flugverhalten	20
3.3	Hardware	21
3.3.1	FlightCTRL	22
3.3.2	NaviCTRL	24
3.3.3	Antrieb	25
3.3.4	Wi.232 Datenfunk	25
3.3.5	Kamera/Aufhängung	27
3.3.6	Verteilerplatine	27
3.4	MikroKopter-Tool	28
3.4.1	MikroKopter-Tool	28
3.4.2	MikroKopterTool-OSD	29
3.5	Datenschnittstelle	30
3.5.1	Protokoll	31
3.5.2	Relevante Kommandos	33
3.5.3	Kodierung	37
3.5.4	Checksumme	38
3.6	Hierarchie der Steuerkommandos	39

4	Konzeption und Entwurf	43
4.1	Anforderungsanalyse	43
4.1.1	Allgemeine Anforderungen	44
4.1.2	Missionskritische Anforderungen	45
4.2	Testsystem	47
4.2.1	Teststrecke und Bewertungstool	47
4.2.2	Testscenarien	47
4.3	Systementwurf	49
4.3.1	Reglerentwurf	49
4.3.2	Bildverarbeitung	55
4.3.3	Missionsplanung und Abarbeitung	57
4.3.4	Modifikationen der FlightCTRL Firmware	61
4.3.5	Bodenstation	62
5	Tests	65
5.1	Parameterjustierung	65
5.2	Test der Regelgrößenermittlung	67
5.3	Ergebnisse	69
6	Zusammenfassung	75
6.1	Auswertung	75
6.2	Ausblick	76
	Tabellenverzeichnis	79
	Abbildungsverzeichnis	81
	Literatur	83

1 Einleitung

Seit geraumer Zeit ist die Menschheit fasziniert vom Fliegen. Als vor rund 500 Jahren Leonardo da Vinci die ersten Entwürfe von Flugmaschinen zeichnete, war für ihn sicher nicht abzusehen, wie sich die Luftfahrttechnik weiterentwickeln würde. Heutzutage gehört das Fliegen zum Alltag. Ob in Rettungseinsätzen, zur Verkehrsüberwachung oder für das schnelle Erreichen des Urlaubsziels. In vielen Fluggeräten kommen dabei Assistenz- und Autopilotensysteme zum Einsatz, die dem Piloten das Fliegen erleichtern oder es ihm komplett abnehmen.

Doch die fortschreitende Entwicklung beschränkt sich nicht nur auf große Fluggeräte. Auch in kleineren Maßstäben, im Modellflugbereich, hat sich seit den Anfängen viel getan. Wie ihre großen Vorbilder, werden die Kleinen in Rettungs- und Erkundungsmissionen eingesetzt. Weitere Einsatzgebiete sind z. B. Luftbildfotografie, Vermessung und Überwachung. Auch ein Transportdienst im städtischen Umfeld wäre denkbar.

Eine Sparte des Modellfluges, der Quadrocopter, scheint besonders für solche Aufgaben geeignet zu sein. Quadrocopter zeichnen sich durch einen einfachen mechanischen Aufbau sowie eine hohe Flugstabilität aus. Es gibt zahlreiche öffentlich zugängliche Projekte, welche sich damit befassen die Entwicklung weiter voranzutreiben. Einige dieser Projekte ermöglichen das Abfliegen von vorprogrammierten Routen. Ermöglicht wird dies durch Einsetzen einfacher GPS-Systeme. Die Grenzen solcher Systeme sind durch die folgend genannten Nachteile allerdings eng gesteckt. Die zum Einsatz kommenden GPS-Systeme haben unter optimalen Bedingungen¹ eine Genauigkeit von 1- 3 Metern[Wik12b]. In der Praxis ist eine Abweichung von ca. 10 Metern allerdings realistischer. Aufgrund der Abschattung von Satelliten ist ein GPS gestützter Einsatz in Gebäuden ebenfalls nicht denkbar. Diese Nachteile könnten durch den Einsatz

¹Verfügbarkeit von EGNOS

von bildverarbeitenden Verfahren beseitigt werden. Ein Einsatz in Gebäuden oder in Gebieten mit Hindernissen wäre somit vorstellbar.

1.1 Thema und Abgrenzung

Das Ziel der vorliegenden Arbeit ist die Erweiterung der Steuerung eines Quadrokopters vom Typ MikroKopter². Mit Hilfe einer am MikroKopter montierten Kamera soll es ermöglicht werden Objekte autonom zu finden und eine Position über diesen einzunehmen. Zuvor soll eine GPS-gestützte Grobpositionierung des MikroKopters in der Nähe des gesuchten Objektes erfolgen.

Die Aufgabenstellung gliedert sich in zwei Teile. Im ersten Teil soll zunächst eine Analyse der bestehenden Zielplattform erfolgen. Es soll herausgearbeitet werden, aus welchen Komponenten der MikroKopter besteht und wie diese zusammenarbeiten. Besondere Aufmerksamkeit soll der Hierarchie der verschiedenen Steuerquellen, wie z. B. Fernsteuerung, GPS und der Kontrolle über die Datenschnittstelle, gewidmet werden. Der zweite Teil umfasst die Konzeption und praktische Umsetzung der erweiterten Steuerung. Dies beinhaltet die GPS-gestützte Grobpositionierung des MikroKopters sowie die Objektfindung und Verfolgung mit bildverarbeitenden Verfahren.

1.2 Aufbau der Arbeit

In der dieser Arbeit werden zunächst die allgemeinen Grundlagen von Drehflüglern und Quadrokoptern erläutert. Es wird auf die grobe Funktionsweise und die Besonderheiten der Steuerung eingegangen. Anschließend wird, in Vorbereitung auf die praktische Aufgabenstellung, die Zielplattform MikroKopter analysiert. Die Hauptschwerpunkte die behandelt werden sind die Hardware, die Datenschnittstelle inklusive des verwendeten Übertragungsprotokolls sowie die Verarbeitung der Steuerkommandos verschiedener Quellen. Darauf folgend wird ein Konzept erarbeitet mit dem die zuvor beschriebene Aufgabenstellung gelöst werden soll. Abschließend wird das entwickelte System im Hinblick auf die im Konzept geforderten Eigenschaften getestet und die Ergebnisse dargelegt.

²<http://www.mikrokoetter.de>

2 Allgemeine Grundlagen

2.1 Drehflügler

In diesem Abschnitt wird dargestellt, wodurch ein Drehflügler definiert ist. Es werden die Vor- und Nachteile angerissen, welche Drehflügler gegenüber anderen Fluggeräten haben. Wichtige physikalische Grundlagen und die Steuerfunktionen werden erläutert.

2.1.1 Definition

Drehflügler stellen eine eigene Klasse von Luftfahrzeugen dar. Sie zeichnen sich dadurch aus, dass der Auftrieb hauptsächlich durch rotierende Tragflächen erzeugt wird[KM12]. Die Internationale Zivilluftfahrtorganisation ICAO definiert den Begriff Drehflügler wie folgt:

"Rotorcraft. A power-driven heavier-than-air aircraft supported in flight by the reactions of the air on one or more rotors."[Org03]

Man unterscheidet viele Arten von Drehflüglern. Die bekanntesten sind Hubschrauber, Tragschrauber und Fluggeräte mit Tilt-Rotor. Weitere Informationen zu den verschiedenen Bauarten befinden sich in [Org03] und [Wik12a].

2.1.2 Eigenschaften

Mit Ausnahme des Tragschraubers sind Drehflügler *Vertical Take-Off and Landing* (VTOL) Flugzeuge. Sie können aus dem Stand abheben und landen. Eine Landebahn, wie sie bei Flächenflugzeugen verwendet wird, ist nicht notwendig. Ein weiterer Vorteil ist die hohe Wendigkeit. Es sind Seitwärts-, Vorwärts- und Rückwärtsflug sowie Drehung um die Hochachse möglich. Im Abschnitt Steuerung auf der nächsten Seite wird näher auf die Begriffe eingegangen.

Demgegenüber haben Drehflügler Nachteile in den Flugleistungen. Die Höchstgeschwindigkeit und Gipfelhöhe sind geringer als bei Flächenflugzeugen. Hinzukommend sind sie im Schwebeflug dynamisch instabil. Wird auf spezielle Stabilisierungsverfahren verzichtet, erfordern sie einen ständigen Steuereingriff, um nicht außer Kontrolle zu geraten. Zum weiteren Studium dieses Verhaltens sei auf [Bit05a] verwiesen.

2.1.3 Giermomentausgleich

Wird ein Rotor über einen Motor angetrieben, welcher sich im Rumpf des Fluggerätes befindet, erzeugt dies ein Gegen-Drehmoment an der Rotorachse. Bei waagrecht angeordneten Rotoren spricht man auch von einem Giermoment. Werden keine Gegenmaßnahmen ergriffen, würden sich Rumpf und Rotor gegenläufig drehen. Der Ausgleich erfolgt meist durch den Einsatz eines Heckrotors oder einer Heckdüse, welche dem Giermoment entgegenwirken. Eine weitere Möglichkeit das Giermoment auszugleichen besteht darin, gegenläufige Rotoren vgl. Abbildung 2.1 zu verwenden. Die Drehmomente der Rotoren heben sich auf, wodurch ein Heckrotor überflüssig wird. Der Vorteil dieser Konstruktion ist, dass die gesamte Antriebsleistung für den Auftrieb zur Verfügung steht.

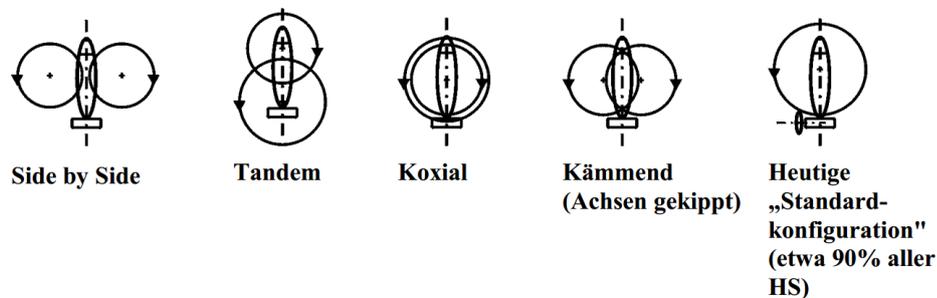


Abbildung 2.1: Übliche Rotoranordnungen[Bit05b]

2.1.4 Steuerung

Die Steuerung von Drehflüglern erfolgt in den meisten Fällen durch eine Änderung der Anstellwinkel der einzelnen Rotorblätter. Wird der Anstellwinkel eines Rotorblattes

erhöht, steigt auch dessen Auftriebskraft. Eine synchrone Winkeländerung aller Blätter wird kollektive Blattverstellung genannt. Mit ihr wird der Auftrieb des Fluggerätes bestimmt. Die Steuerung in verschiedene Richtungen wird durch eine zyklische Blattverstellung erreicht. Das bedeutet, ein Rotorblatt ändert während einer Umdrehung seinen Anstellwinkel, abhängig von seiner aktuellen Lage. Beispielsweise werden im Vorwärtsflug die Rotorblätter, welche sich gerade hinten befinden, stärker geneigt um einen höheren Auftrieb zu erzeugen. Das bewirkt eine Neigung des Fluggerätes nach vorne. Dadurch wird der Schubvektor geneigt und es entsteht Vortrieb.

Bei Quadroptern kommt eine andere Art der Steuerung zum Einsatz. Anders als Hubschrauber, verfügen sie nicht über eine zyklische Blattverstellung. Die Steuerung wird ausschließlich durch gezielte Änderung der Schubkräfte der einzelnen Rotoren ermöglicht. Dies geschieht entweder über Drehzahländerung der Rotoren oder eine kollektive Blattverstellung. Weitere Erklärungen befinden sich im Abschnitt Quadropter.

Die Orientierung im Raum wird durch ein objektbezogenes, rechtshändiges Koordinatensystem beschrieben. Der Koordinatenursprung liegt im Mittelpunkt des Fahrzeuges. Für die Ausrichtung der Achsen existieren zwei verschiedenen Konventionen. Das *East-North-Up* (ENU) und *North-East-Down* (NED) System. Bei Fluggeräten wird das NED-System verwendet. Es zeichnet sich dadurch aus, dass die y-Achse nach rechts und die z-Achse nach unten zeigt. Siehe Abbildung 2.2 auf der nächsten Seite. Die Steuerfunktionen um die Achsen des Koordinatensystems haben folgende Bezeichnungen[Wik12c].

Nicken (engl. pitch): Drehung des Fluggerätes um die Querachse(y-Achse). Bei einem Drehflügler entspricht das Nicken dem sogenannten Nase hoch bzw. runter. Es wird damit der Vorwärts- und Rückwärtsflug eingeleitet.

Rollen (engl. roll): Drehung um die in Flugrichtung verlaufende x-Achse. Ausgehend vom Schwebestand wird ein Fliegen nach rechts oder links eingeleitet.

Gieren (engl. yaw): Drehung um die senkrecht durch das Fluggerät verlaufende z-Achse.

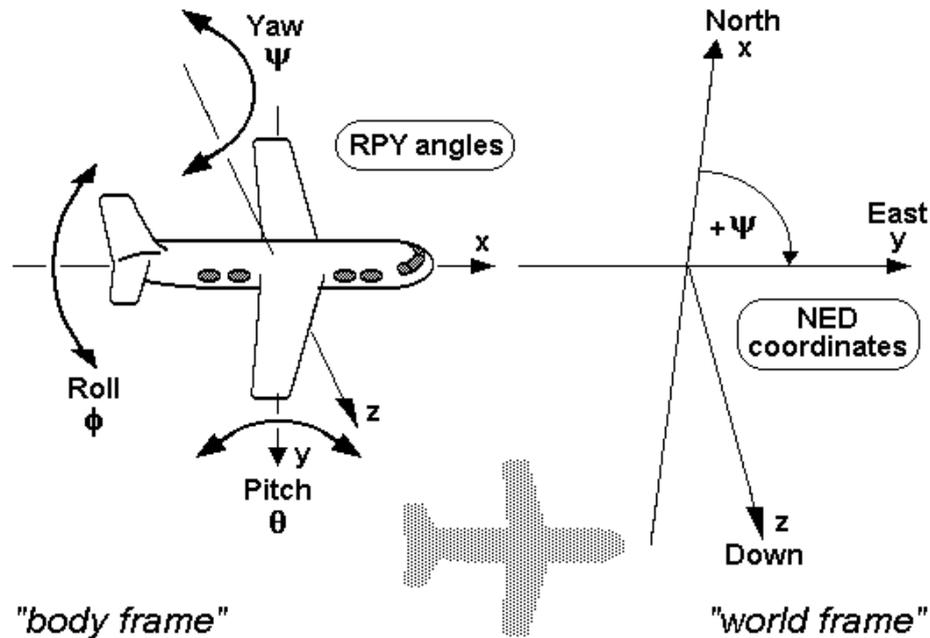


Abbildung 2.2: Roll-Pitch-Yaw Winkel von Luftfahrzeugen[Wik12c]

2.2 Quadrocopter

Die Entstehungsgeschichte der Quadrocopter reicht bis in die Anfangszeiten von Drehflüglern zurück. Bereits im Jahr 1924 existierte mit dem „Oehmchen No. 2“ eine Flugmaschine, welche über vier Hauptrotoren verfügte[Bit05b]. Mit der Erfindung der Taumelscheibe, welche die zyklische Blattverstellung ermöglichte, geriet die Entwicklung von Quadrocoptern in den Hintergrund. Erst mit den Fortschritten in der Elektronik und Sensorik schritt die Entwicklung weiter voran. Die Entwicklungen beschränken sich hauptsächlich auf den Modellflugsektor. Projekte mit manntragenden Koptern wie der „e-volo“ sind die Ausnahme[vol13].

Quadrocopter gehören zu der Klasse der Hubschrauber und sind ebenfalls VTOL-Flugzeuge[Wik13]. Beim klassischen Quadrocopter wird der Auftrieb durch vier nach oben gerichtete, in einem Quadrat angeordnete, Rotoren erzeugt. Unter anderem existieren auch Varianten mit sechs(Hexakopter), acht(Oktokopter) oder zwölf(Dodekakopter) Rotoren. Der Vorteil bei der Verwendung von mehr als vier Rotoren liegt in der höheren Ausfallsicherheit und der größeren maximalen Nutzlast. In Abbildung 2.3 werden

verschiedene Bauarten und Rotoranordnungen gezeigt. Aufgrund der unterschiedlichen Anzahl der verwendeten Rotoren und dessen Anordnung wird häufig die Allgemeine Bezeichnung Multikopter verwendet.

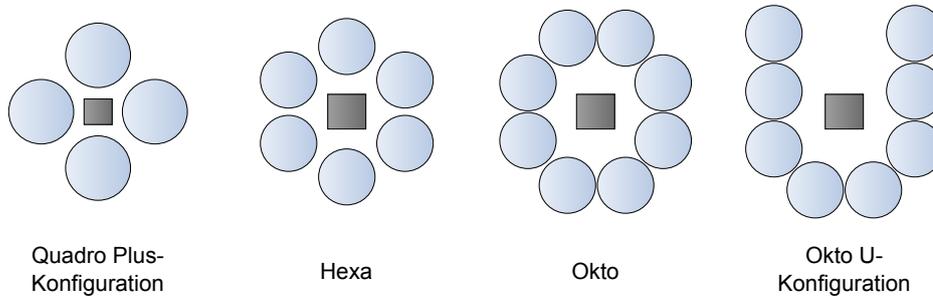


Abbildung 2.3: Multikopter-Bauarten und deren Rotoranordnungen

Quadrokopter besitzen gegenüber anderen Drehflüglern zahlreiche Vorteile. Die Anzahl beweglicher Bauteile ist sehr gering. Bei den üblicherweise drehzahlgesteuerten Koptern sind die einzigen mechanischen Verschleißteile die Kugellager der Motoren, wodurch sie sehr wartungsarm sind. Ein weiterer Vorteil des einfachen Aufbaus ist die preiswerte Instandsetzung nach einem Absturz. In den meisten Fällen beschränken sich die Beschädigungen auf gebrochene Rotoren und verbogene Ausleger. Aufgrund des meist verwendeten direkten Antriebes, bei welchem jeder Rotor direkt von einem eigenen Motor angetrieben wird, sind Quadrokopter energieeffizienter als Hubschrauber.

2.2.1 Realisierung der Steuerung

Laut [Bü10] wird die Steuerung bei Quadrokoptern, ähnlich wie bei Hubschraubern, durch gezielte Neigung des Fluggerätes realisiert. Bei Quadrokoptern wird diese Neigung alleine durch Schubkraftänderung der einzelnen Rotoren erreicht. Wie im Abschnitt Steuerung bereits erwähnt, wird die Schubkraftänderung eines Rotors entweder durch Drehzahländerung oder durch eine synchrone Änderung der Anstellwinkel der Rotorblätter erreicht. Die Umsetzung mithilfe der Drehzahländerung ist die übliche und auch preiswertere Variante, da bei ihr weniger mechanische Bauteile nötig sind. Erst bei größeren Rotordurchmessern wird auf eine Steuerung über die Anstellwinkel zurückgegriffen. Grund hierfür ist die höhere rotierte Masse. Bei Steuervorgängen, unter

Verwendung von Drehzahländerung, müsste diese Masse ständig beschleunigt und abgebremst werden. Regelvorgänge würden träge werden, mehr Energie erfordern und die Antriebsaggregate stärker belasten. Im restlichen Teil der Arbeit wird von Drehzahlgesteuerten Quadroptern in Plus Konfiguration ausgegangen, wenn dies nicht anders erwähnt wird.

Wie in Abbildung 2.4 zu sehen ist, werden die Rotoren so angeordnet, dass sie paarweise gegenläufig sind. Entstehende Giermomente heben sich gegenseitig auf. Auch bei Flugmanövern werden die Rotoren so angesteuert, dass die entstehenden Giermomente im Gleichgewicht sind. Die Steuerung wird, ausgehend vom Schwebезustand, folgendermaßen umgesetzt[MW11c].

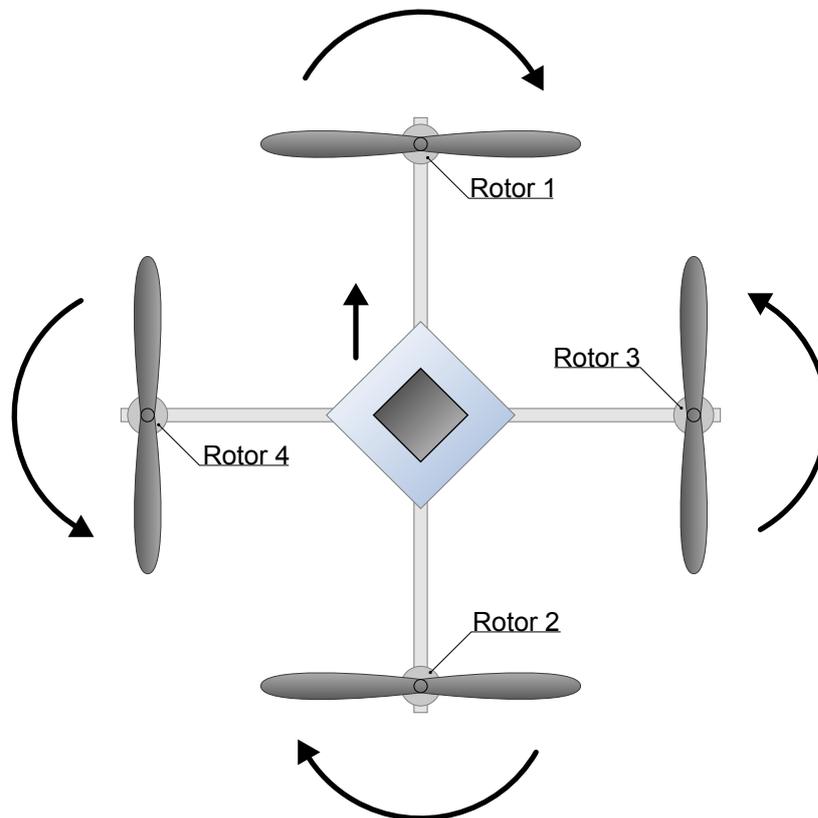


Abbildung 2.4: Drehrichtung der Rotoren eines Quadropters

Schweben Alle Rotoren werden mit der gleichen Drehzahl angetrieben. Die Summe der Giermomente ist null. Die Höhe wird durch synchrones Ändern der Drehzahl aller Rotoren gesteuert.

Nicken Ein Nicken nach vorne wird dadurch erreicht, dass an Rotor zwei die Drehzahl erhöht und an Rotor eins gesenkt wird. Die Drehzahlen der beiden Rotoren werden dabei so geändert, dass die Summe der beiden Giermomente konstant bleibt. Rotoren drei und vier behalten ihre Drehzahl.

Rollen Die Rollbewegung wird nach dem gleichen Prinzip wie die Nickbewegung umgesetzt. Mit dem Unterschied, dass die Drehzahlen der Rotoren drei und vier geändert werden und die Rotoren eins und zwei gleichbleibend angetrieben werden.

Gieren Für eine reine Gierbewegung, also eine Drehung um die Hochachse, werden die Giermomente der Rotoren genutzt. Eine Drehung, von oben gesehen nach rechts, erfolgt durch gemeinsames Senken der Drehzahl der rechtsdrehenden Rotoren. An den linksdrehenden Rotoren wird die Drehzahl erhöht. In Folge dessen überwiegt das Giermoment der linksdrehenden Rotoren. Es wird eine Gierbewegung nach rechts ausgelöst.

Durch Mischen dieser Grundmanöver sind die verschiedensten Flugfiguren möglich. Die Ansteuerung der Motoren wird durch eine Steuerelektronik vorgenommen. Sie wertet die empfangenen Steuerkommandos aus und berechnet, wie stark die Rotoren angetrieben werden müssen.

2.2.2 Stabilisierung

Wie bereits in Abschnitt 2.1.2 auf Seite 11 erwähnt, sind Drehflügler dynamisch instabil. Quadrokopter sind ohne eine Stabilisierung nur schwer zu kontrollieren. Um dennoch ein stabiles Flugverhalten zu erreichen, kommen elektronische Sensorsysteme zum Einsatz. Die wichtigsten Sensoren sind Gyroskope und Beschleunigungssensoren.

Gyroskop Mit einem Gyroskop wird die Drehrate (Winkelgeschwindigkeit) um eine Achse ermittelt. Eine Translation hat keine Auswirkung auf die Messwerte. Dreht

sich das Gyroskop entsprechend seiner Messachse, wird ein Signal mit einer Stärke relativ zu der Drehgeschwindigkeit erzeugt. Durch Integration dieser Messwerte kann die Rotation ermittelt werden, welche seit Beginn der Messung vollzogen wurde. Ein Nachteil dieser Sensoren ist, dass sich durch Sensordrift verursachte Messfehler über die Zeit aufsummieren. Eine genaue Bestimmung der Fluglage über einen längeren Zeitraum ist demzufolge nicht möglich. In Quadrokoptern kommen drei Gyroskope zum Einsatz, um die Rotation um alle drei Raumachsen zu erfassen.

Beschleunigungssensor Beschleunigungssensoren erfassen die Beschleunigung, welche auch unter der Bezeichnung g-Kraft bekannt ist. Dies geschieht durch Messung der Trägheitskraft, welche auf eine bekannte Testmasse einwirkt. Diese Messwerte können dazu verwendet werden, die aktuelle Rotation im Raum zu ermitteln. Im Gegensatz zu Gyroskopen liefern Beschleunigungssensoren absolute Messwerte unabhängig von der Lage zum Initialisierungszeitpunkt. Der Nachteil gegenüber Gyroskopen ist, dass sich Rotationen sowie Translationen auf die Messwerte auswirken und nicht eindeutig voneinander getrennt werden können. Die eingesetzten Sensoren erfassen die Beschleunigungskräfte aller drei Raumachsen.

Zusammen bilden diese Sensoren eine inertielle Messeinheit (engl. *inertial measurement unit* (IMU)). Mit ihr können die Beschleunigungen der sechs Freiheitsgrade gemessen werden. Die Steuerelektronik vergleicht diese Messwerte mit den Sollwerten, welche aus den Steuerkommandos generiert werden. Abweichungen werden durch Regelungsalgorithmen ausgeglichen.

3 Analyse der Systemumgebung

In diesem Kapitel wird die Zielplattform MikroKopter, siehe Abbildung 3.1, genauer beschrieben. Es soll ein grober Überblick über die Hard- und Software sowie die Flugleistungen vermittelt werden. Anschließend werden die, für die erweiterte Steuerung, relevanten Komponenten und Sachverhalte analysiert. Das beinhaltet eine detaillierte Beschreibung der Datenschnittstelle. Ergänzend wird die Reaktion des MikroKopters auf Steuerkommandos untersucht.



Abbildung 3.1: MikroKopter

3.1 Allgemeines

Das MikroKopter-Projekt wurde am 24.10.2006 von Holger Buss und Ingo Buster ins Leben gerufen[MW10]. Anfangs richtete sich das Projekt an Bastler und Modellbauer. Mittlerweile ist das Projekt kommerziell ausgelegt und läuft über die Firma *HiSystems*.

Es sind verschiedene Bauformen, vom kleinen Quadrocopter bis zum Oktokopter, als Bausatz und Teilweise auch flugfertig montiert, verfügbar. Der Quellcode ist teilweise verfügbar und darf für nicht kommerzielle Zwecke modifiziert werden.

3.2 Flugverhalten

Der MikroKopter hat einige Besonderheiten die von anderen Fluggeräten abweichen. Dieser Abschnitt soll dem Leser eine grobe Vorstellung über des Flugverhalten des MikroKopters vermitteln.

Lagestabilisierung

Ein wesentliches Merkmal des Flugverhaltens ist die Lagestabilisierung. Anders als bei einem Hubschrauber wird nicht die Drehrate um eine Achse, sondern der Winkel des Kopters gesteuert. Das bedeutet, der MikroKopter nimmt bei einer Betätigung des Nick- oder Roll-Steuerknüppels einen Winkel proportional zu der Knüppelstellung ein. Wird der Steuerknüppel wieder in seine Mittelposition geführt, kehrt der MikroKopter auch in eine waagerechte Lage zurück. Dieser Flugmodus wird auch als ACC-Modus bezeichnet, weil hierfür ein Beschleunigungssensor(Accelerometer) notwendig ist.

Trägheit

Wie jeder Körper unterliegt auch der MikroKopter einer Massenträgheit. Befindet sich der MikroKopter in einer Bewegung und hat dabei eine waagerechte Lage eingenommen, so wird er diese Bewegung beibehalten. Einzig der Luftwiderstand wirkt diesem Verhalten entgegen. Um den Kopter nach einem Flugmanöver wieder zum Stillstand zu bringen, muss die Bewegungsenergie durch gezieltes Gegensteuern abgebaut werden.

Drift

Anders als häufig vermutet, kann der MikroKopter seine Position nicht durch alleinige Nutzung von Gyroskopen und Beschleunigungssensoren halten. Positioniert man den fliegenden MikroKopter an einer Stelle und greift anschließend nicht mehr in das Flugeschehen ein, so wird er seine Position langsam verlassen.

In welchem Ausmaß diese Abdrift eintritt ist von vielen Faktoren abhängig. Die Hauptursache dieser Abdrift sind äußere Einflüsse, wie Wind und Luftverwirbelungen. Doch selbst in einer vollständig windstillen Umgebung ist ein Abdriften zu beobachten. Gründe hierfür sind unter anderem Ungenauigkeiten und Messfehler der Sensorik.

3.3 Hardware

Der verwendete MikroKopter ist ein QuadroKopter in Plus-Konfiguration. Die Komponenten sind auf einem Aluminiumkreuz montiert. Abbildung 3.2 auf der nächsten Seite zeigt eine schematische Darstellung der Komponenten. Der Achsabstand zweier gegenüberliegender Motoren beträgt 40cm. In der Mitte der Kontruktion befindet sich ein Platinenturm, siehe Abbildung 3.3 auf Seite 23, welcher die elektronischen Komponenten beherbergt. Tabelle 3.1 zeigt die wesentlichen technischen Merkmale und die Flugleistungen.

MikroKopter: Quadro	
Abmessungen ohne Propeller und Antennen in cm (h*b*t)	46 * 46 * 15
Gewicht flugfertig	750g
Geschwindigkeit	ca. 100km/h
Flugdauer	ca. 8 - 20min.
Flughöhe	Funk- oder Sichtweite, genaue Grenze ungetestet
Reichweite	ca. 1km (Funkreichweite)
Nutzlast (sinnvoll)	ca. 500g
Schubkraft	ca. 2000g
Motoren	4 * Brushless
Sensorik	Gyroskope, Beschleunigungssensoren, Barometer, Kompass, GPS
Stromaufnahme	schweben ca. 7A / Vollast ca. 50A

Tabelle 3.1: Technische Daten des verwendeten MikroKopters

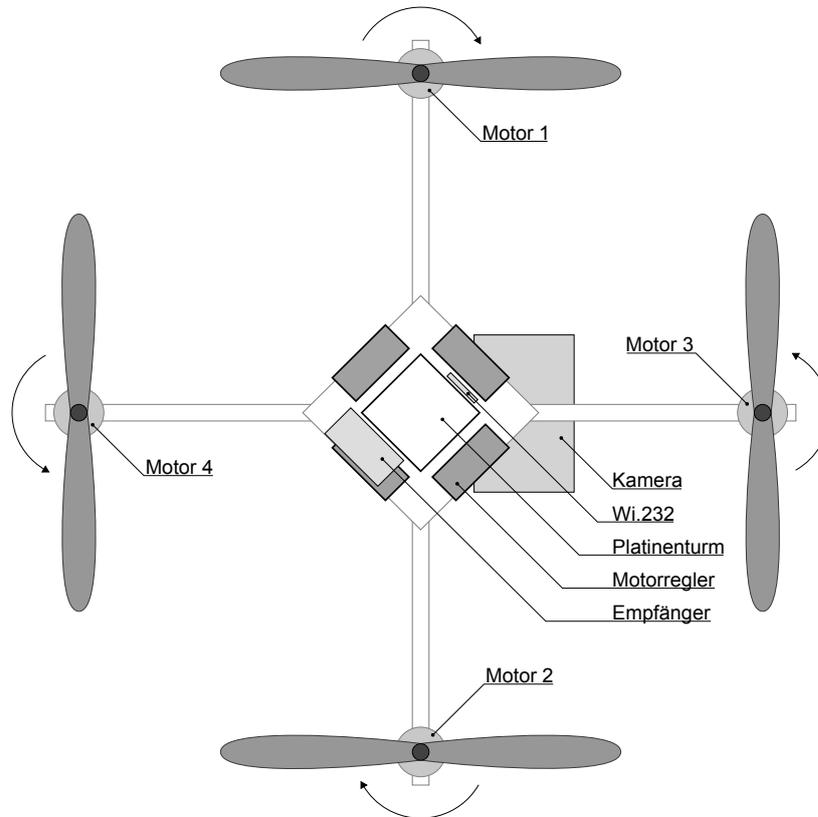


Abbildung 3.2: Schematische Darstellung der Komponenten des MikroKopters

3.3.1 FlightCTRL

Die FlightCTRL ist die Kernkomponente des MikroKopters. Auf ihr befindet sich die Sensorik, um die aktuelle Fluglage zu ermitteln. Es kommen drei Gyroskope und ein Dreiachs-Beschleunigungssensor zum Einsatz. Zusätzlich befindet sich auf der Platine ein Barometer, welches den Luftdruck erfasst. Anhand des Luftdruckes ist es möglich die aktuelle Flughöhe zu bestimmen. Die Datenverarbeitung erfolgt durch einen Mikrocontroller vom Typ *Atmel ATmega644P*. Genaue Bezeichnungen sind der Tabelle 3.2 auf der nächsten Seite zu entnehmen.

Die FlightCTRL übernimmt die wesentlichen Aufgaben um den MikroKopter zu steuern. Sie nimmt das Signal des Fernsteuerempfängers entgegen und wertet es aus. Außerdem werden die Sensordaten verarbeitet. Aus den Sensordaten und den

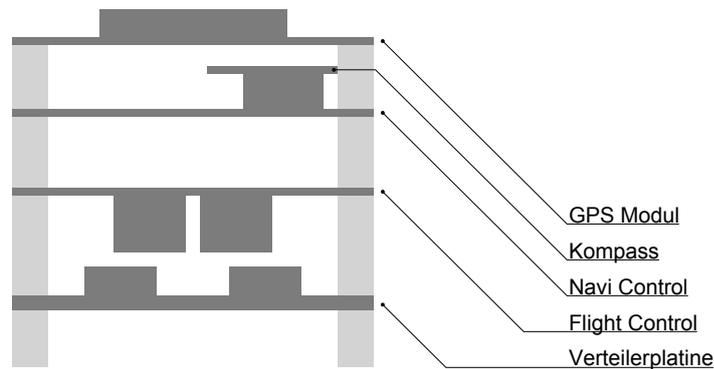


Abbildung 3.3: Platinenturm des MikroKopters

empfangenen Steuerkommandos werden die Steuersignale für die Motorregler generiert und an diese übergeben.

Die FlightCTRL besitzt außerdem einige Schnittstellen. Diese bieten z. B. die Möglichkeit Modellbauservos für eine aktive Lagekorrektur einer Fotokamera anzuschließen. Unter den Schnittstellen befindet sich auch eine serielle RS-232 Schnittstelle. Sie wird hauptsächlich dazu verwendet die Flugparameter des MikroKopters zu konfigurieren. Über diese Schnittstelle können auch externe Steuerkommandos eingegeben und Sensordaten abgefragt werden. Eine genauere Analyse erfolgt in Abschnitt 3.5 auf Seite 30.

FlightCTRL	
Version	FlightCTRL V2.0
Mikrocontroller/Takt	ATmega644P / 20MHz
Gyroskope	3 * MEMS ADXRS610
Beschleunigungssensor	LIS344ALH
Barometer	MPX4115A
Schnittstellen	RS-232(57600Baud), I ² C-Bus, SPI ¹ , PPM ² , 5 * Servo, 2 * Schaltausgang
Ansteuerbare Motoren	max. 12
Rotorkonfigurationen	frei konfigurierbar

Tabelle 3.2: Technische Daten der FlightCTRL[MW11a][MW13]

¹Serial Peripheral Interface

²PPM- Summensignal vom RC- Empfänger

3.3.2 NaviCTRL

Die NaviCTRL erweitert den MikroKopter um die Möglichkeit GPS-Positionsdaten zu empfangen. Diese Daten werden dazu verwendet Wegpunkte abzufliegen, die Position zu halten (PositionHold) oder zu der Startposition zurückzufliegen (CommingHome). Außerdem wird die Funktion *FollowMe* ermöglicht, bei welcher der MikroKopter einem zweiten GPS-Modul, das seine Position zum MikroKopter sendet, folgt. Zusätzlich wird auch die *CareFree*-Steuerung zuschaltbar. Bei dieser Steuerung muss der Pilot nicht mehr darauf achten in welche Himmelsrichtung der Kopter gerade gegiert ist. Ein eingehendes Nick-Kommando wird beispielsweise immer in eine Flugbewegung nach Norden umgesetzt. Genauere Informationen befinden sich in [MW12a].

Die GPS-Daten werden von einem ublox-LEA6S Modul geliefert, das sich an oberster Stelle des Platinenturmes befindet. Auf der NaviCTRL befindet sich außerdem ein digitaler 3D-Kompass, um die absolute Lage um die Gierachse zu ermitteln. Die Verarbeitung der GPS- und Kompassdaten übernimmt ein ARM9-Mikrocontroller. In Tabelle 3.3 werden die technischen Daten nochmals zusammengefasst.

NaviCTRL	
Version	NaviCTRL V1.1
Mikrocontroller	STR911FAM44X6
GPS-Modul	ublox-LEA6S
Funktionen	Wegpunkteflug, PositionHold, CommingHome, CareFree, FollowMe
Aktionsradius	Wegpunkteflug und FollowMe auf 250m begrenzt
max. Wegpunkte	30
Sonstiges	Micro SD-Kartenslot für Datenlogging
Schnittstellen	2 * RS-232(57600Baud), SPI, USB
Hallsensoren	3 * KMZ51

Tabelle 3.3: Technische Daten der NaviCTRL[MW12f][MW12e]

Ist eine Funktionen der NaviCTRL aktiv, werden Steuerdaten in Form von Nick-, Roll- und Gierkommandos generiert und an die Debug-Schnittstelle der FlightCTRL gesendet. Die FlightCTRL übernimmt weiterhin die Funktion der Stabilisierung. Die Steuerkommandos der NaviCTRL werden wie Kommandos der Fernsteuerung behandelt. Näheres dazu folgt in Abschnitt 3.6 auf Seite 39.

Die NaviCTRL besitzt eine weitere serielle Datenschnittstelle, welche die Funktion der nun belegten Schnittstelle der FlightCTRL übernimmt. An dieser Datenschnitt-

stelle empfangenen Datenpakete werden von der NaviCTRL an die entsprechenden Baugruppen des MikroKopters weitergeleitet. Mit Hilfe einer Adresse, die in jedem Datensatz enthalten ist, kann bestimmt werden, zu welchem Modul die Daten gesendet werden bzw. von welchem sie stammen. Dieser Sachverhalt wird in Abschnitt 3.5 auf Seite 30 genauer betrachtet.

3.3.3 Antrieb

Der Antrieb des verwendeten MikroKopters erfolgt durch vier bürstenlose Gleichstrommotoren. Diese Motorenbauart besitzt eine hohe Energieeffizienz und unterliegt nur einem geringen Verschleiß. Ein Nachteil dieser Motoren ist die aufwendigere Ansteuerungselektronik.

Anders als bei einem Bürstenmotor muss eine Elektronik die Kommutierung³ zwischen den drei Motorwicklungen übernehmen. Die Kommutierung muss so erfolgen, dass ein zur Motordrehzahl synchrones Drehfeld erzeugt wird. Der Zeitpunkt der Kommutierung wird dabei durch eine Spannungsmessung an den gerade nicht bestromten Wicklungen ermittelt. Die Drehzahl der Motoren wird über eine, dem Drehfeld überlagerte, Pulsweitenmodulation gesteuert[MW11b].

Um diese Aufgaben zu erfüllen, verfügt jeder Motor des MikroKopters über einen eigenen Motorregler. Die Motorregler sind über ein gemeinsames Bussystem⁴ mit der FlightCTRL verbunden, über das sie Steuerdaten für die Drehzahl erhalten. Jeder Motorregler besitzt eine eindeutige Adresse um gezielt angesteuert zu werden. Gespeist wird der Antrieb durch einen Lithium-Polymer-Akkumulator. Weitere technische Daten des Antriebes befinden sich in Tabelle 3.4 auf der nächsten Seite.

3.3.4 Wi.232 Datenfunk

An der freien seriellen Datenschnittstelle der NaviCTRL ist ein *Radiotronix Wi.232* Modul angeschlossen. Mit dessen Hilfe wird eine drahtlose bidirektionale Datenübertragung zu einer Gegenstation, welche ebenfalls mit einem solchen Modul ausgestattet sein muss, ermöglicht. Die Verbindung zwischen den Modulen wird von diesen selbst ausgehandelt. Für den MikroKopter und die entsprechende Gegenstation sind die

³Umschalten auf die nächste Motorwicklung

⁴I²C-Bus

Motoren/Propeller	
Bezeichnung	E-sky EK5-0002B
UPM/V	1000
Wicklungsstrom	max. 8A (dauerlast)
Propeller	EPP1045 (10x4.5 Zoll)
Motorregler	
Version	BL-Ctrl 1.2
Belastbarkeit	max. 12A (begrenzt)
Betriebsspannung	max 16V
Ansteuerung	I ² C-Bus oder PPM
Stromversorgung	
Technologie	Lithium Polymer
Kapazität / Nennspannung	2200mAh / 11.1V
Strombelastbarkeit	55A (dauerlast) / max. 110A (kurzzeitig)

Tabelle 3.4: Technische Daten des Antriebes[MW11b]

Module somit transparent und brauchen bei einer Datenübertragung nicht speziell berücksichtigt zu werden. Die eingesetzten Module senden im, für Europa freigegebenen, 860MHz SRD-Band⁵. Die Reichweite beträgt laut Hersteller 1500 Meter. In der Praxis wird mit einer einfachen Drahtantenne eine Reichweite von ca. 500 Meter erreicht. Weitere technische Daten können dem Datenblatt[Rad07] und der Tabelle 3.5 entnommen werden.

Wi.232 Funkmodul	
Modul	Wi.232EUR-R
Baudrate	2400- 115200
Reichweite	500m Drahtantenne / >6Km 12dB Yagiantenne am Boden
Sendeleistung	max. 25mW, abhängig vom Kanal
Kanäle	2 Wideband, 6 Narrowband
Frequenzband	868- 870 MHz

Tabelle 3.5: Technische Daten des Wi.232 Funkmoduls[MW12g][Rad07]

Die drahtlose Datenverbindung kann dazu eingesetzt werden, aktuelle Sensordaten des fliegenden MikroKopters zu einer Bodenstation zu übertragen. Ebenso ist es

⁵Short Range Devices

möglich, von einer Bodenstation generierte, Steuerkommandos und GPS- Routen an den MikroKopter zu senden.

3.3.5 Kamera/Aufhängung

An dem rechten Ausleger des MikroKopters ist eine Kamera mit Blickrichtung nach unten montiert. Es handelt sich um eine Netzwerkkamera des Herstellers AXIS-Communications. Ein integriertes W-Lan-Modul ermöglicht es, Videodaten während des Fluges zu einer Bodenstation zu senden. Um ein möglichst ruhiges Videobild zu erhalten ist die Kamera vom Rest des Systems mechanisch entkoppelt. Reichweitentests haben bei einer akzeptablen Bildqualität und 30 Bildern/s ca. 200m ergeben.

Kamera	
Bezeichnung	AXIS M1011-W
Übertragungsstandard	IEEE 802.11g/b
Bildsensor	1/4"RGB CMOS
Stromversorgung	4,9 - 5,1V, max. 6,5W
Blickwinkel	ca. 69° x 54,5° (entzerrt)
Reichweite	ca. 200m (abhängig v. Kompression)
Auflösung/Bildrate	640 x 480, 30 Bilder/s
Videokomprimierung	H.264, Motion JPEG, MPEG-4
Streamingprotokoll	RTSP, RTP

Tabelle 3.6: Technische Daten der Kamera[COM10]

3.3.6 Verteilerplatine

An unterster Stelle des Platineturms befindet sich die Verteilerplatine. Die Hauptaufgabe der Platine besteht darin, den I²C-Bus und die Stromleitungen an die Motorregler zu verteilen. Ein elektronischer Leistungsschalter ermöglicht das schnelle An- und Abschalten der Hauptstromversorgung. Außerdem beherbergt die Platine Schaltverstärker für die LED- Beleuchtung. Mikroschalter ermöglichen zusätzlich eine Zwangsabschaltung der Beleuchtung und des Summers.

3.4 MikroKopter-Tool

Das MikroKopter-Tool ist ein Windowsprogramm mit dem die Baugruppen des MikroKopters parametrieren und analysiert werden können. Es ist Bestandteil des MikroKopter-Projektes und steht zur kostenlosen Nutzung zur Verfügung. Die notwendige Datenkommunikation erfolgt über die serielle Schnittstelle der NaviCTRL. Es ist unterteilt in das eigentliche MikroKopter-Tool und das MikroKopterTool-OSD. Im Folgenden wird der komplexe Funktionsumfang nur angerissen. Weiterführende Informationen können dem MikroKopter Wiki[MW12b] entnommen werden.

3.4.1 MikroKopter-Tool

Das MikroKopter-Tool, das in Abbildung 3.4 auf der nächsten Seite zu sehen ist, wird hauptsächlich dazu verwendet den MikroKopter zu parametrieren. Es können unter anderem die internen Regelparameter eingestellt und die Empfindlichkeit auf Steuerbefehle justiert werden. Es ist somit möglich auf das Flugverhalten des MikroKopters Einfluss zu nehmen. Der Kopter kann sehr gutmütig, für z. B. Fotoflüge, aber auch sehr agil, für sportliches Fliegen, konfiguriert werden. Weiterhin kann die Anzahl und Anordnung der verwendeten Motoren eingestellt werden. Die Zuordnung der Steuerkanäle zu jeweils einer Funktion wird hier ebenfalls vorgenommen.

Ein weiterer Verwendungszweck des MikroKopter-Tools ist das Anzeigen von Sensordaten. Es kann zwischen den verschiedenen Baugruppen des MikroKopters, wie z. B. FlightCTRL und NaviCTRL, gewählt werden. Die Sensorwerte werden in numerischer Form auf der linken Seite oder wahlweise als zeitlicher Verlauf im oberen Bereich dargestellt.

Das MikroKopter-Tool bietet auch die Möglichkeit zwölf weitere serielle Kanäle zu nutzen. Diese Kanäle dienen als Erweiterung zu den Kanälen des Fernsteuersenders und können in der Kanalzuordnung ebenfalls Funktionen zugeordnet werden.

Weitere Funktionen sind das Anzeigen von Fehlercodes der MikroKopter-Baugruppen und das Flashen der Firmware. [MW12c]

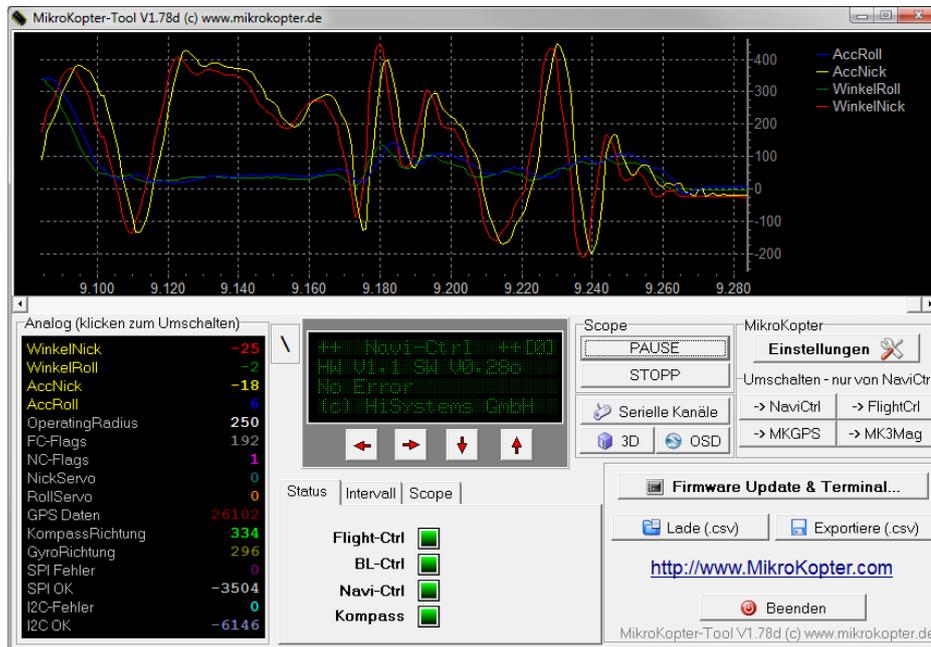


Abbildung 3.4: Benutzeroberfläche des MikroKopter-Tools

3.4.2 MikroKopterTool-OSD

Mit dem OSD des MikroKopter-Tools, siehe Abbildung 3.5 auf der nächsten Seite, wird der Wegpunkteflug des MikroKopters konfiguriert. Zu diesem Zweck wird ein Bild, welches eine Karte oder ein Satellitenbild enthält, geladen. Dem Bild müssen GPS-Koordinaten beiliegen, die den Bereich definieren der vom Bild abgedeckt wird. Diese Koordinaten sind entweder im Bild enthalten oder müssen im OSD eingegeben werden.

Mit Hilfe des Wegpunkt-Generators, welcher im OSD enthalten ist, können Wegpunkt-muster erstellt werden. Es sind verschiedene Muster, die z. B. einen Kreis darstellen oder ein Gebiet abdecken, möglich. Zu jedem Wegpunkt können die Flughöhe, Steigrate, Geschwindigkeit, die Verweildauer und der Fangradius definiert werden. Den Wegpunkten können weiterhin zwei, an der FlightCTRL auszuführende, Servoaktionen übergeben werden. Dies kann beispielsweise zum Ausrichten und Auslösen einer Kamera verwendet werden. Es können maximal 30 Wegpunkte definiert werden. Aus Sicherheitsgründen ist der Aktionsradius des GPS-gestützten Fluges auf 250m begrenzt.

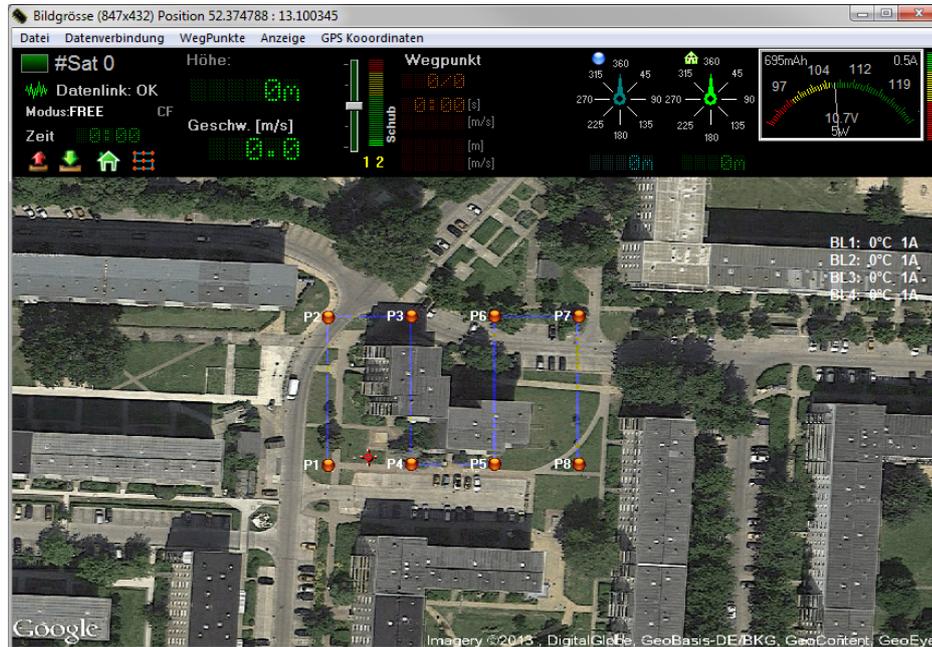


Abbildung 3.5: Benutzeroberfläche des MikroKopterTool-OSD

Die erstellte Wegpunktliste wird zum MikroKopter übertragen und in der NaviCTRL gespeichert. Durch Auslösen eines Funktionskanals der Fernsteuerung oder mit Hilfe eines seriellen Kanals wird der Wegpunkteflug eingeleitet. Eine Abhängigkeit zu dem MikroKopterTool-OSD besteht nicht mehr. Während des Fluges zeigt das OSD Telemetriedaten an, wie beispielsweise die aktuelle Geschwindigkeit und die Entfernung von der Startposition.[MW12d]

3.5 Datenschnittstelle

In den Kapiteln NaviCTRL auf Seite 24 und MikroKopter-Tool auf Seite 28 war bereits die Rede von einer seriellen Schnittstelle, welche für die Datenkommunikation eingesetzt wird. An dieser Stelle wird das verwendete Übertragungsprotokoll, die verschiedenen Kommandos und deren Kodierung genauer betrachtet. Die Informationen stammen aus eigenen Untersuchungen und dem MikroKopter Wiki[MW12h].

3.5.1 Protokoll

Das Kommunikationsprotokoll, siehe Abbildung 3.6, ist wie folgt aufgebaut. Ein neuer Datensatz wird mit dem *Start-Byte* '#' gekennzeichnet. Es folgt ein Byte, das die Baugruppe des MikroKopters definiert, für welche der Datensatz bestimmt ist bzw. von welcher es stammt. Tabelle 3.7 zeigt die hierfür möglichen Adressen. Das nächste Byte ist das *ID-Byte*. Es sagt aus, um was für ein Kommando oder um welche Antwort es sich handelt. Eine Übersicht der definierten ID's ist in Tabelle 3.8 auf der nächsten Seite zu sehen. Dem ID-Byte folgen die eigentlichen *Nutzdaten* variabler Länge, welche mit einer Base64 ähnlichen Kodierung versehen sind. Die nächsten zwei Bytes stellen eine *Checksumme* dar, um Fehler in der Datenübertragung zu erkennen. Sie werden aus dem Adress-Byte, dem ID-Byte und den Nutzdaten gebildet. Das Ende des Datensatzes wird mit einem *Stopp-Byte* '\r' gekennzeichnet.

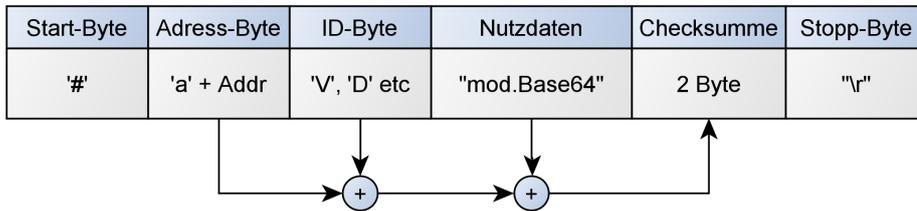


Abbildung 3.6: Übertragungsprotokoll der Seriellen Schnittstelle der NaviCTRL

Adresse	Baugruppe	resultierendes Adressbyte
1	FlightCTRL (FC)	'b'
2	NaciCTRL (NC)	'c'
3	Kompass (MK3MAG)	'd'
5	Motorregler (BL-CTRL)	'f'

Tabelle 3.7: Adressen der MikroKopter Baugruppen

Beschreibung	ID	Adr.	Daten	ID	Adr.	Daten
Allgemeine Kommandos	an MikroKopter gesendet			Antwort		
Debugwert-Bezeichner abrufen	'a'	Any	Index	'A'	Slave	Bezeichnung
Externe Kontrolle	'b'	Any	Steuerwerte	'B'	Slave	echo Frame
Display abfragen (Abo.)	'h'	Any	Index, Intervall	'H'	Slave	Displaytext
Display abfrage (einmalig)	'l'	Any	Index	'L'	Slave	Displaytext
Versionsabfrage	'v'	Any	-	'V'	Slave	Versionsnr.
Debug-Daten abfragen (Abo.)	'd'	Any	Intervall	'D'	Slave	Debug-Daten
Reset	'R'	Any	-	-	-	-
Wertabfrage Externe Kontrolle	'g'	Any	-	'G'	Slave	Steuerwerte
FlightCTRL Kommandos	an FlightCTRL gesendet			Antwort von FlightCTRL		
Kompass Ausrichtung	'K'	FC	Winkel	'k'	MK3	Fluglage
Motortest	't'	FC	Motor Testwerte	'T'	FC	-
Parametersatz abfragen	'q'	FC	Parameterindex	'Q'	FC	Parametersatz
Parametersatz schreiben	's'	FC	Parametersatz	'S'	FC	Parameterindex
RC-Kanäle abfragen	'p'	FC	-	'P'	FC	RC-Kanalwerte
3D-Daten abfragen (Abo.)	'c'	Any	Intervall	'C'	FC	3D-Data
Mixertabelle abfragen	'n'	FC	-	'N'	FC	Mixertabelle
Mixertabelle schreiben	'm'	FC	MixerTabelle	'M'	FC	gültig / ungültig
Parametersatz wechseln	'f'	FC	Parameterindex	'F'	FC	Parameterindex
Seriellen Kanal setzen	'y'	FC	Kanalwerte	-	-	-
BL-CTRL Param. lesen	'u'	FC	Adr. Motorregler	'U'	FC	BL Parametersatz
BL-CTRL Param. schreiben	'w'	FC	BL Parametersatz	'W'	FC	Status
NaviCTRL Kommandos	an NaviCTRL gesendet			Antwort von NaviCTRL		
Verbindungstest	'z'	NC	Testdaten	'Z'	NC	echo: Testdaten
Fehlertext abfragen	'e'	NC	-	'E'	NC	Fehlertext
Zielposition senden	's'	NC	Wegpunkt-Strukt.	-	-	-
Wegpunkt senden	'w'	NC	Wegpunkt-Strukt.	'W'	NC	Wegpunkt-Anz.
Wegpunkt abrufen	'x'	NC	Wegpunkt Index	'X'	NC	Wegpunkt-Strukt.
OSD- Daten abrufen	'o'	NC	Intervall	'O'	NC	OSD-Daten
UART umschalten	'u'	NC	Baugruppe des	-	-	-
3D-Daten abfragen (Abo.)	'c'	Any	Intervall	'C'	NC	3D-Data
Parameter setzen/abfragen	'j'	NC	ID, Parameter	'J'	NC	echo: ID, Wert
BL-CTRL Statusabfrage	'k'	Any	Intervall	'K'	Slave	BL-CTRL Status

Tabelle 3.8: Kommandos und die zugehörigen ID's der MikroKopter Baugruppen.

3.5.2 Relevante Kommandos

Einige der in Tabelle 3.8 auf der vorherigen Seite aufgeführten Kommandos scheinen im Hinblick auf die Aufgabenstellung besonders relevant zu sein. Sie werden im Folgenden, zusammen mit ihren Besonderheiten, genauer vorgestellt.

Umschalten der UART-Adresse (ID 'u')

Wie in Abschnitt 3.3.2 auf Seite 24 erwähnt, leitet die NaviCTRL die an der Datenschnittstelle eingehenden Datenpakete an die adressierten Baugruppen weiter. Entsprechend müssen die Antworten der Baugruppen ebenfalls, zurück durch die NaviCTRL, an die Datenschnittstelle geschleust werden. Zu diesem Zweck existiert ein Kommando, das die NaviCTRL anweist, die Antworten einer bestimmten Baugruppe durchzuleiten. Diese Weiterleitung ausgehender Antworten ist dauerhaft aktiv, bis eine andere Quelladresse gewählt wird. Das Kommando beinhaltet lediglich ein Byte, welches die Baugruppe bestimmt, von welcher die Antworten durchgeschleust werden sollen.

Das Zurückschalten zu der NaviCTRL erfolgt durch Senden eines sogenannten „magic packet“. Das Senden dieses Paketes weicht von dem üblichen Übertragungsprotokoll ab. Es wird nur das unveränderte fünf Byte lange Paket gesendet. Tabelle 3.9 zeigt die Baugruppen und die zugehörigen Adresswerte beziehungsweise das „magic packet“.

Baugruppe	Wert	Bemerkung
FlightCTRL (FC)	0	
Kompass (MK3MAG)	1	
GPS (MKGPS)	2	
NaviCTRL (NC)	„0x1B,0x1B,0x55,0xAA,0x00“	„magic packet“

Tabelle 3.9: Zieladressen und „magic packet“ für das Umschalten der UART-Adresse

Externe Kontrolle (ID 'b')

Mit diesem Kommando ist es möglich den MikroKopter, zusätzlich zur Fernsteuerung, auch über die Datenverbindung zu steuern. In der Konfiguration des MikroKopters kann ein Kanal der Fernsteuerung definiert werden welcher die Steuerung über die Datenschnittstelle freigibt oder blockiert. Somit ist es dem Piloten jederzeit möglich

die Kontrolle zu übernehmen. Ein gesendetes Steuerkommando ist so lange aktiv, bis ein weiteres gesendet wird oder die externe Kontrolle per Schaltkanal deaktiviert wird.

In Tabelle 3.10 sind die im Kommando enthaltenen Daten aufgeschlüsselt. Die Werte von *Nick*, *Roll* und *Gier* werden den empfangenen Steuerkommandos der Fernsteuerung aufaddiert. Siehe Abschnitt 3.6 auf Seite 39. Der Wert *Frame* wird nach Empfang des Kommandos als Bestätigung zurück gesendet.

Name	Datentyp	Bemerkung
Digital	u8[2]	ungenutzt
RemoteTasten	u8	ungenutzt
Nick	s8	Nickbewegung
Roll	s8	Rollbewegung
Gier	s8	Gierbewegung
Gas	u8	Gasstick (begrenzung auf RC-Gas)
Height	s8	Sollwert für Höhenregelung
free	u8	ungenutzt
Frame	u8	Bestätigungsframe
Config	u8	ungenutzt
Nutzdaten: 11 Byte, Kodiert+Protokoll: 22 Byte		

Tabelle 3.10: Inhalt des Kommandos *Externe Kontrolle*

Debug-Daten (ID 'd')

Dieses Kommando gehört zu den allgemeinen Kommandos und wird von den Baugruppen FlightCTRL, NaviCTRL und MK3MAG unterstützt. Die Antwort enthält wichtige Statusdaten des MikroKopters. Für die Aufgabenstellung sind insbesondere die Sensorwerte der Gyroskope und des Barometers interessant. In Tabelle 3.11 ist die Antwort auf das Kommando zu sehen. Die Bedeutung der enthaltenen Daten wird auszugsweise in Tabelle 3.12 auf der nächsten Seite gezeigt.

Name	Datentyp	Bemerkung
Digital	u8[2]	ungenutzt
Analog	s16[32]	Debug-Daten
Nutzdaten: 66 Byte, Kodiert+Protokoll: 94 Byte		

Tabelle 3.11: Antwort auf ein Debug-Daten Kommando

Abalog[n]	FlightCTRL	NaviCTRL	MK3MAG
0	WinkelNick	WinkelNick	Magnet X
1	WinkelRoll	WinkelRoll	Magnet Y
2	AccNick	AccNick	Magnet Z
3	AccRoll	AccRoll	RohMagnet X
4	GyroGier	OperationRadius	RohMagnet Y
5	Höhenwert	FC-Flags	RohMagnet Z
6	AccZ	NC-Flags	Lage Nick
7	Gas	NickServo	Lage Roll
8	Kompasswert	RollServo	Magnet X Offset
⋮	⋮	⋮	⋮
30	GPS Nick	GPS Roll	I2C-Fehler
31	GPS Roll	Anz. Sat	I2C OK

Tabelle 3.12: Auszug aus dem Inhalt des *Analog* Arrays der Debug-Daten Antwort der jeweiligen Baugruppen

Eine Besonderheit dieses Kommandos ist, dass es sich um ein Abonnement der Antwortdaten handelt. Empfängt eine Baugruppe dieses Kommando, sendet sie ihre Debug-Daten kontinuierlich zurück. Mit dem Kommando wird auch die gewünschte Pausenzeit zwischen den automatisch gesendeten Antworten übermittelt. Das Abonnement ist vier Sekunden lang gültig. Wird innerhalb dieser Zeit keine neue Anfrage gestellt wird das Senden der Antwort eingestellt.

Es kann immer nur die Antwort einer einzigen Baugruppe zurück gesendet werden. Welche das aktuell ist, wird durch die ausgewählte UART-Adresse bestimmt.

Serielle Kanäle (ID 'y')

Wie in Abschnitt 3.4 auf Seite 28 erwähnt, ist es möglich, zusätzliche serielle Kanäle zu übertragen. Die Kanäle können in der MikroKopter-Konfiguration, ebenso wie die Kanäle der Fernsteuerung, verschiedenen Funktionen zugeordnet werden. Eine Ausnahme davon stellen die Steuerfunktionen Nick, Roll, Gier und Gas dar. Der Aufbau des Kommandos ist in Tabelle 3.13 auf der nächsten Seite zu sehen.

Die übertragenen Werte der Kanäle sind im MikroKopter so lange aktiv, bis ein anderer Wert übertragen wird. Ein Abschalten der Verarbeitung dieser Kanäle, wie es bei der externen Kontrolle möglich ist, wurde hier nicht vorgesehen. Bricht während

Name	Datentyp	Bemerkung
SerialPoti	s8[12]	Werte der seriellen Zusatzkanäle
Nutzdaten: 12 Byte, Kodiert+Protokoll: 22 Byte		

Tabelle 3.13: Aufbau des Kommandos *Serielle Kanäle*

des Fluges die Datenverbindung ab, bleiben die zuletzt gesendeten Werte weiterhin aktiv und es kann kein Einfluss mehr auf sie genommen werden. Dieser Umstand ist vermutlich auch der Grund dafür, dass die seriellen Kanäle keinen Steuerfunktionen zugeordnet werden können.

Name	Datentyp	Bemerkung
Longitude	s32	Längengrad * 10^7 , $13.1012973^\circ = 121012973$
Latitude	s32	Breitengrad * 10^7 , $52.3747139^\circ = 523747139$
Altitude	s32	Flughöhe
Status	u8	Gültigkeit, 0=Ungültig, 1=Neu, 2=Verarbeitet
Headung	s16	Ausrichtung, 0=egal, 1-360 Richtung, <0 = Index POI
ToleranceRadius	u8	Fangradius, wenn erreicht dann nächster Wegpunkt
HoldTime	u8	Verweildauer bei Wegpunkt
Event_Flag	u8	unbenutzt
Index	u8	Identifizierung verschiedener Wegpunkte
Type	u8	Typ des Wegpunktes, 255=ungültig, 0=WP, 1=POI
WP_EventChannelValue	u8	Servokanal
AltitudeRate	u8	Steigrate bei Änderung der Flughöhe
Speed	u8	Geschwindigkeit zwischen Wegpunkten
CamAngle	u8	Winkel des Kameraservos
Name	u8[4]	Name des Wegpunktes
reserve	u8[2]	unbenutzt
Nutzdaten: 30 Byte, Kodiert+Protokoll: 46 Byte		

Tabelle 3.14: Aufbau der Wegpunktstruktur

Senden von Wegpunkten (ID 'w')

Mit Hilfe dieses Kommandos werden GPS-Wegpunkte an die NaviCTRL gesendet. Das Kommando verwendet die Datenstruktur aus Tabelle 3.14. Die bereits gespeicherten Wegpunkte können durch Senden eines ungültigen Wegpunktes gelöscht werden. Dies

geschieht indem der Wert von *Type* auf 255 gesetzt wird. Als Antwort auf einen empfangenen Wegpunkt sendet die NaviCTRL die Anzahl der gespeicherten Punkte und den Wert von *Index* des zuletzt empfangenen Punktes als Bestätigung zurück.

Es ist auch möglich den Typ (*Type*) eines Wegpunktes als POI (Point of Interest) zu deklarieren. Der negative *Index* des als POI deklarierten Punktes kann anschließend bei *Heading* eingesetzt werden. Dies bewirkt ein Ausrichten des MikroKopters auf die GPS-Koordinaten des angegebenen POI. Das Verhalten kann bei einem Fotoflug gewünscht sein, bei dem ein feststehendes Objekt aufgenommen werden soll.

3.5.3 Kodierung

Der Block im Übertragungsprotokoll, welcher die Nutzdaten enthält, wird vor der Übertragung mit einer angepassten Base64-Kodierung versehen. Die im MikroKopter verwendete Implementierung, siehe Abbildung 3.7, unterscheidet sich etwas von der vorgeschlagenen nach RFC4648[Jos06].

```
1 void SendOutData(unsigned char cmd, unsigned char addr, unsigned char *snd, unsigned
  char len){
2     unsigned int pt = 0;
3     unsigned char a,b,c;
4     unsigned char ptr = 0;
5     SendeBuffer[pt++] = '#'; // Start-Byte
6     SendeBuffer[pt++] = 'a' + addr; // Adress
7     SendeBuffer[pt++] = cmd; // Command
8     while(len){
9         //MikroKopter Base64 Kodierung
10        if(len) { a = snd[ptr++]; len--;} else a = 0;
11        if(len) { b = snd[ptr++]; len--;} else b = 0;
12        if(len) { c = snd[ptr++]; len--;} else c = 0;
13        SendeBuffer[pt++] = '=' + (a >> 2);
14        SendeBuffer[pt++] = '=' + (((a & 0x03) << 4) | ((b & 0xf0) >> 4));
15        SendeBuffer[pt++] = '=' + (((b & 0x0f) << 2) | ((c & 0xc0) >> 6));
16        SendeBuffer[pt++] = '=' + (c & 0x3f);
17    }
18    AddCRC(pt);
19 }
```

Abbildung 3.7: Codeauszug der Base64 Implementierung des MikroKopters

Die Nutzdaten eines Kommandos werden in ein Bytearray kopiert. Es werden jeweils drei Byte des Arrays in vier Blöcke zu 6-Bit aufgeteilt. Die entstehenden Blöcke besitzen nun einen Wertebereich von 0-63. Abbildung 3.8 auf der nächsten Seite veranschaulicht

diesen Sachverhalt. Jeder dieser 6-Bit Blöcke wird anschließend mit 61('=') addiert und dem zu sendenden Nutzdatenblock(Byte-Array) angehängen. Dieser Vorgang wird wiederholt, bis alle Bytes auf diese Weise Kodiert wurden. Ist die Anzahl der zu kodierenden Bytes nicht durch drei teilbar, werden vor der Kodierung entsprechend viele Null-Füllbytes ergänzt.

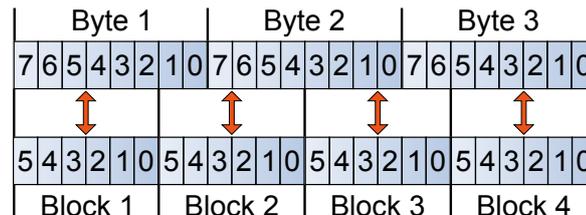


Abbildung 3.8: Base64 Kodierung

Die so kodierten Nutzdaten belegen nur noch einen Wertebereich von 61-124. Für den Fall, dass die unkodierten Nutzdaten durch drei teilbar sind, steigt durch die Kodierung die Größe um 33%. Bestehen die Nutzdaten aus nur einem Byte, vervierfacht sich die Datenmenge sogar.

Der Grund für die verwendete Kodierung ist nicht klar erkennbar. Auffällig ist jedoch, dass die verwendeten Start- und Stoppzeichen des Übertragungsprotokolls außerhalb des Wertebereiches liegen, welcher von der Kodierung genutzt wird. Es liegt die Vermutung nahe, dass die Erschaffung von eindeutigen Start- und Stoppmarkierungen der Hauptgrund für die Kodierung ist.

3.5.4 Checksumme

Im Übertragungsprotokoll sind die vorletzten zwei Bytes als Checksumme definiert. Mit deren Hilfe, können mit hoher Wahrscheinlichkeit Fehler in der Datenübertragung erkannt werden. In Abbildung 3.9 auf der nächsten Seite ist die vom MikroKopter verwendete Implementierung zu sehen.

Die Checksumme wird aus dem Adress-Byte, dem ID-Byte und den kodierten Nutzdaten gebildet. Aus diesen Daten wird die Quersumme gebildet und Modulo 4096 berechnet. Das Ergebnis wird anschließend durch 64 dividiert. Das erste Byte der Checksumme wird aus der Addition von 61('=') mit dem ganzzahligen Quotienten

gebildet. Das zweite Byte aus der Addition von 61(=') mit dem ganzzahligen Rest der Division.

Auch diese beiden Bytes haben die Eigenschaft, dass der genutzte Wertebereich nur einen Bereich 61-124 abdeckt. Somit sind die einzigen beiden Zeichen, die außerhalb dieses Wertebereiches liegen das Start- und das Stoppzeichen. Die Vermutung, welche in der Sektion Kodierung aufgestellt wurde, wird somit noch bestärkt.

```
1 void AddCRC(unsigned int DataBufferLength){
2     unsigned int tmpCRC = 0, i;
3     for(i = 0; i < DataBufferLength; i++){
4         tmpCRC += TxdBuffer[i];
5     }
6     tmpCRC %= 4096;
7     TxdBuffer[i++] = '=' + tmpCRC / 64;
8     TxdBuffer[i++] = '=' + tmpCRC % 64;
9     TxdBuffer[i++] = '\r';
10 }
```

Abbildung 3.9: Implementierung der Checksummenbildung des MikroKopters

3.6 Hierarchie der Steuerkommandos

Im MikroKopter existieren drei verschiedene Quellen von denen Steuerbefehle stammen können. Die Hauptquelle für Steuerbefehle ist der RC-Empfänger, der die Steuerbefehle der Fernsteuerung empfängt. Des Weiteren generiert die NaviCTRL, bei einer aktivierten GPS-Funktion, ihre eigenen Befehle, welche sie an die FlightCTRL sendet. Die dritte Möglichkeit Steuerbefehle zu empfangen ist die externe Kontrolle, die in Abschnitt 3.5.2 auf Seite 33 erläutert wurde. Die Verarbeitung dieser drei Quellen erfolgt in der FlightCTRL.

Nick, Roll, Gier

In Abbildung 3.10 auf der nächsten Seite wird die Verarbeitung von Nick-, Roll- und Gierkommandos der Quellen schematisch dargestellt. Es ist zu sehen, dass die Kommandos der Fernsteuerung immer verarbeitet werden. Die Steuerwerte der NaviCTRL hingegen werden nur verarbeitet, wenn sich die Steuerknüppel der Fernsteuerung

nahe dem Mittelpunkt befinden. Überschreiten die Steuerknüppel den eingestellten Schwellenwert, wird die GPS-Funktionalität deaktiviert. Werden die Steuerknüppel losgelassen, nimmt die GPS-Steuerung ihre vorherige Funktion wieder auf. Nick-, Roll- und Gierkommandos der externen Kontrolle werden nur verarbeitet, wenn der dafür definierte Schalter an der Fernsteuerung aktiviert ist. Sind die Voraussetzungen für die Verarbeitung der externen Kontrolle oder der GPS-Steuerung erfüllt, werden die Werte zu den Steuerwerten der Fernsteuerung aufsummiert.

Die daraus resultierenden Steuerdaten werden an die Lageregelung des MikroKopters weitergeleitet. Der Lageregler berechnet aus den Steuer- und Sensordaten die Sollwerte für die Motorregler und sorgt so dafür, dass der MikroKopter eine Schräglage relativ zu den Steuerwerten einnimmt.

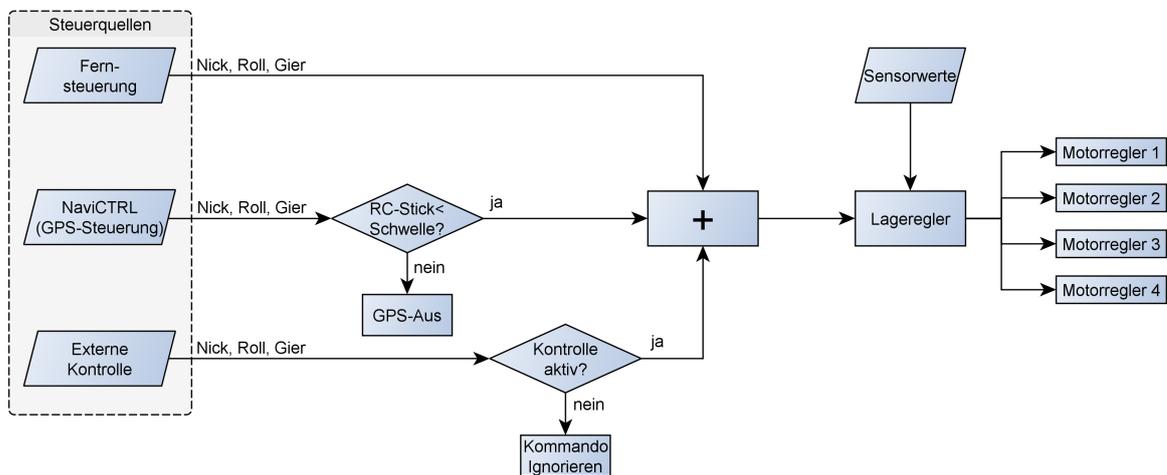


Abbildung 3.10: Verarbeitung von Nick-, Roll- und Gierkommandos der drei Steuerquellen

Gas

Die Gas-Werte werden, wie in Abbildung 3.11 auf der nächsten Seite dargestellt, verarbeitet. In der Abbildung sind nur die Steuerquellen Fernsteuerung und externe Kontrolle aufgeführt. Der Grund hierfür ist, dass die NaviCTRL keine Gas-Werte generiert. Die Gas-Werte der Externen Kontrolle werden ebenfalls nur verarbeitet,

wenn der dafür vorgesehene Schalter aktiviert ist. Ist dies der Fall, wird der kleinere der beiden Gas-Werte verwendet und an die Lageregelung des MikroKopters weitergeleitet.

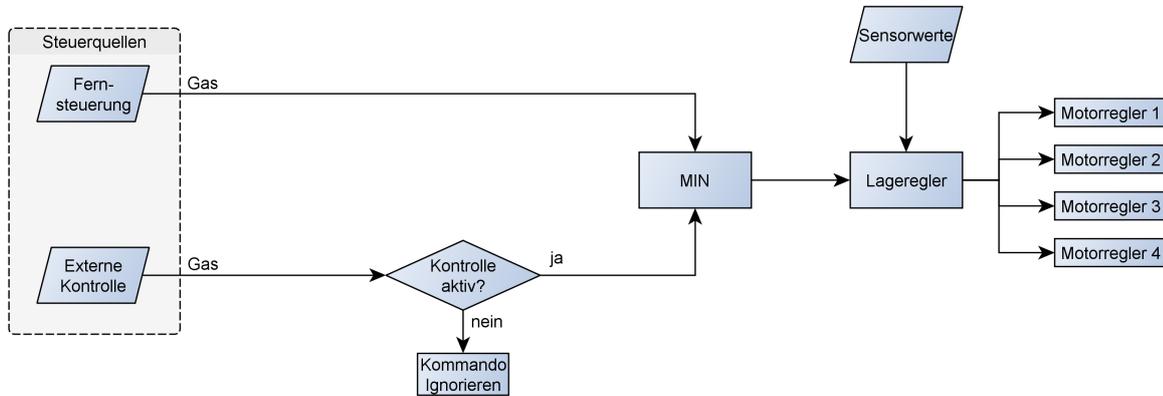


Abbildung 3.11: Verarbeitung des *Gas* Steuerwertes der verschiedenen Quellen

4 Konzeption und Entwurf

Im letzten Kapitel wurde die zugrundeliegende Zielplattform MikroKopter vorgestellt. Aufbauend auf den gewonnenen Erkenntnissen, erfolgt nun die Erarbeitung einer Lösung für den zweiten Teil der Aufgabenstellung. Das beinhaltet zunächst die Herausarbeitung und Analyse der Anforderungen, die an das zu entwickelnde System gestellt werden. Es wird das eingesetzte Testsystem vorgestellt und die darauf durchzuführenden Testszenarien erläutert. Anschließend wird ein Konzept erarbeitet, wie die zuvor dargestellten Anforderungen erfüllt werden können.

4.1 Anforderungsanalyse

Im Vorfeld dieser Arbeit wurde bereits die Entscheidung getroffen, die Aufgabenstellung mit Hilfe einer Bodenstation umzusetzen. Die Sensor- und Videodaten des MikroKopters werden dabei in Echtzeit zu einem am Boden befindlichen Notebook gesendet und dort verarbeitet. Die so generierten Steuerkommandos werden anschließend zum MikroKopter zurück gesendet. In Abbildung 4.1 wird das Konzept grob skizziert.

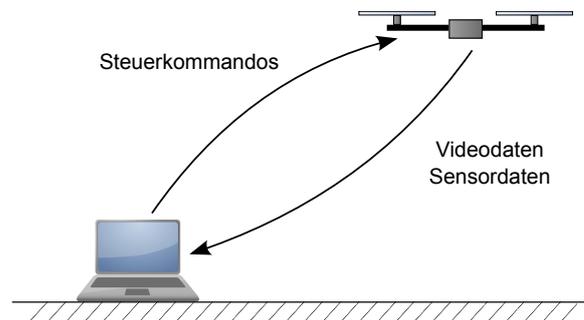


Abbildung 4.1: Grobkonzept

Der Grund für diese Entscheidung ist die bereits vorhandene Hardware zur Video- und Datenübertragung, welche in Abschnitt 3.3 auf Seite 21 vorgestellt wurde. Eine Suche nach geeigneter Hardware, die für eine On-Board Datenverarbeitung geeignet wäre, kann somit entfallen. Diese Entscheidung wurde auch zugunsten einer komfortableren Entwicklungsumgebung getroffen. Nach einer Anpassung der zu entwickelnden Software muss diese nicht erst wieder auf den MikroKopter übertragen werden. Zusätzlich werden durch eine Bodenstation umfangreiche Debugging-Möglichkeiten geschaffen. Bei der folgenden Anforderungsanalyse wird diese Entscheidung bereits berücksichtigt.

4.1.1 Allgemeine Anforderungen

Bodenstation

Wie eingangs erwähnt, soll zur Lösung der Aufgabe eine Bodenstation zum Einsatz kommen. Die Software auf der Bodenstation soll dazu in der Lage sein die aktuellen Daten zu visualisieren. Dazu zählen die Videodaten der Kamera und die Zwischenbilder der Bildverarbeitungskette. Es soll außerdem eine Auswahl wichtiger Telemetriedaten sowie die generierten Steuerdaten angezeigt werden. Bei der Visualisierung der Daten muss berücksichtigt werden, dass der Tester auf den MikroKopter konzentriert sein wird. Die Anzeige der Daten muss demzufolge klar und deutlich erkennbar dargestellt werden. Außerdem soll es möglich sein die relevanten Daten für eine spätere Auswertung aufzuzeichnen.

Es ist zu berücksichtigen, dass die Bodenstation eine Vielzahl an Aufgaben erfüllt. Missionskritische Funktionen wie die Bildverarbeitung, das Empfangen und Senden von Daten sowie das Generieren von Steuerdaten haben Priorität und dürfen nicht behindert werden.

Fallback

Bei einem autonom agierenden System muss gerade in der Entwicklungsphase damit gerechnet werden, dass unvorhergesehene Ereignisse eintreten. Aus Sicherheitsgründen und um Schäden am System zu vermeiden ist es wünschenswert, dass im Fehlerfall jederzeit eingegriffen werden kann. Es muss ausnahmslos zu jedem Zeitpunkt die Möglichkeit bestehen jegliche autonome Funktion zu deaktivieren und die volle Kontrolle

zu übernehmen. Diese Fallback-Möglichkeit sollte idealerweise vollständig unabhängig von dem entwickelten System funktionieren. Im Fall des MikroKopters wäre es wünschenswert, wenn diese Fallback-Möglichkeit über die Fernsteuerung und unabhängig von der Datenübertragung aktivierbar ist.

Datenkommunikation

Die drahtlose Datenübertragung zwischen MikroKopter und Bodenstation ist ein wesentlicher Bestandteil der Aufgabenerfüllung. Sie wird dazu verwendet Telemetriedaten zu erhalten und Steuerkommandos zu senden. Es muss deshalb sichergestellt werden, dass Fehler in der Datenübertragung vermieden oder zumindest erkannt werden. Durch das vom MikroKopter vorgegebene Übertragungsprotokoll wird dieser Punkt bereits erfüllt.

Da es sich bei dem zu entwickelnden System um eine Echtzeitanwendung handelt, sind zusätzlich geringe Latenzen und eine hohe Wiederholungsrate wünschenswert. Idealerweise ist die Wiederholungsrate, mit der Datenpakete gesendet und empfangen werden können, gleich oder höher der Bildrate der Kamera.

4.1.2 Missionskritische Anforderungen

Missionsablauf

Die Aufgabenstellung beinhaltet das Anfliegen einer vorgegebenen GPS-Position, an der sich das Zielobjekt befindet. Anschließend soll mit Hilfe der Kamera eine Relativposition zu dem Objekt eingenommen werden.

Es muss ein geeigneter Missionsablauf definiert werden, der sicherstellt, dass das Objekt überflogen wird und somit in den Blickbereich der Kamera gerät. Dabei sind GPS-Abweichungen zu berücksichtigen, die dies verhindern könnten. Zusätzlich müssen Abbruchbedingungen für den Fall einer erfolglosen Suche geschaffen werden. Der Wegpunktflug der NaviCTRL muss durch das zu entwickelnde System programmiert und genutzt werden können. Der aktuelle Missionsstatus soll durch geeignete Maßnahmen angezeigt werden.

Bildverarbeitung

Die Bildverarbeitung hat die Aufgabe das Zielobjekt im Bildbereich zu lokalisieren. Zu berücksichtigen ist dabei, das vom Weitwinkelobjektiv stark verzerrte Bild. Um eine Bildübertragung mit einer Bildrate von 30 Bildern/s über eine große Distanz zu ermöglichen, muss das Bild stark komprimiert werden. Die Bildverarbeitung muss demzufolge robust gegenüber den auftretenden Kompressionsartefakten sein. Um die Reaktionsgeschwindigkeit des Systems hoch zu halten, ist es erstrebenswert eine Verarbeitungsfrequenz gleich der maximalen Bildrate der Kamera zu erreichen. Dies entspricht einer maximalen Verarbeitungszeit von $1/30$ s, welche für die Berechnung eines einzelnen Bildes zur Verfügung steht.

Zu Entwicklungszwecken werden die Anforderungen an die Objekterkennung vorerst eingeschränkt. Das Objekt soll sich deutlich, z. B. durch seine Farbe, vom Hintergrund abheben und eine kompakte Kontur besitzen. Ist das entwickelte System mit dieser Einschränkung stabil, kann in einem späteren Entwicklungsschritt die Bildverarbeitung auf andere Objekte ausgeweitet werden.

Regelung

Die Regelung ist dafür zuständig, aus der bereits ermittelten Objektposition im Kamerabild und den empfangenen Telemetriedaten, geeignete Steuerdaten zu generieren.

Eine Kernanforderung ist, dass die Genauigkeit der Positionierung gegenüber einer reinen GPS-Lösung erheblich gesteigert wird. Das beinhaltet auch ein stabiles, nicht aufschwingendes Regelverhalten. Die Regelung muss robust gegenüber verschiedenen Anfluggeschwindigkeiten sein, da diese, verursacht durch äußere Einflüsse, nicht immer gleichbleibend ist. Führt der MikroKopter ein Steuerkommando aus, ändert sich seine Fluglage. Diese Änderung der Fluglage bewirkt eine Verschiebung der Objektposition im Kamerabild, was fälschlicherweise als eine Positionsänderung interpretiert werden könnte. Es müssen Maßnahmen getroffen werden, solche Fluglageänderungen zu kompensieren. Des Weiteren muss die Flughöhe berücksichtigt werden. Der Grund hierfür ist, dass die von der Kamera erfasste Fläche des Bodens mit steigender Flughöhe zunimmt. Ein am Rand des Kamerabildes erfasstes Objekt bedeutet in geringer Flughöhe eine kleinere Lageabweichung, als bei einer großen Flughöhe. Darüber hinaus muss auch der Fall berücksichtigt werden, dass die Bildverarbeitung nicht mit jedem Frame

ein gültiges Ergebnis liefert. Kurze Ausfälle müssen überbrückt werden können. In der Praxis muss zusätzlich mit Wind gerechnet werden.

4.2 Testsystem

Um Aussagen über die Qualität der Regelung machen zu können, ist ein geeignetes Testsystem notwendig. Es soll dazu in der Lage sein die Flugbahn des MikroKopters aufzuzeichnen. Um mögliche Störeinflüsse zu minimieren soll diese Aufzeichnung unabhängig von dem System MikroKopter stattfinden. Im Folgenden wird das verwendete Testsystem kurz vorgestellt. Anschließend werden durchzuführende Testszenarien definiert.

4.2.1 Teststrecke und Bewertungstool

In Abbildung 4.2 auf der nächsten Seite, ist eine schematische Zeichnung der Draufsicht der Teststrecke zu sehen. Die im Bild eingezeichnete Fluglinie wird von einer Videokamera betrachtet. Die Position der Kamera relativ zu der Fluglinie und die Länge der erfassten Strecke wurden vermessen. Befindet sich der MikroKopter auf der in der Abbildung eingezeichneten Fluglinie, kann aus der Position im Kamerabild und den Vermessungsdaten der Teststrecke die reale Flugposition ermittelt werden. Dies geschieht mit Hilfe eines speziell für diesen Zweck entwickelten Bewertungstools, in dem die MikroKopter-Position framewise markiert wird.

4.2.2 Testszenarien

Um die Erfüllung der zuvor aufgeführten Anforderungen an die Regelung zu überprüfen, ist eine Reihe von Testfällen notwendig. Im Folgenden werden die jeweiligen Anforderungen und die dazugehörigen Tests aufgeführt. Die Resultate werden mit der eingangs beschriebenen Testmethode aufgezeichnet. Die Regelung muss bei diesen Tests, entsprechend der Teststrecke, auf eine Achse beschränkt werden. Um den MikroKopter innerhalb der Teststrecke zu halten wird die nicht geregelte Achse vom Tester manuell gesteuert.

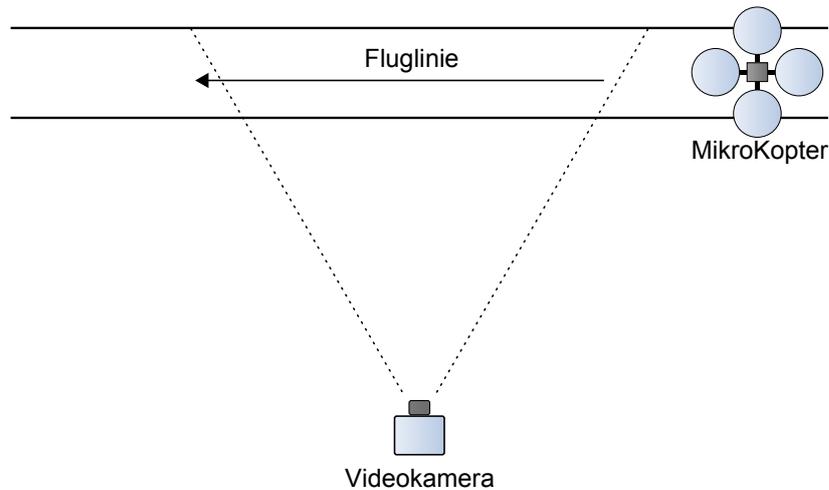


Abbildung 4.2: Draufsicht der Teststrecke

Genauigkeit und Schwingverhalten Der MikroKopter wird über dem Objekt platziert. Anschließend wird die Regelung aktiviert. Dieser Test wird in zwei verschiedenen Flughöhen durchgeführt.

Robustheit gegenüber Höhe und Geschwindigkeit Der MikroKopter wird entlang der Teststrecke manuell in den Bereich des Objektes geflogen. Die manuelle Steuerung muss zu diesem Zweck so modifiziert werden, dass sie deaktiviert wird sobald das Objekt detektiert wurde. Dieser Test wird mit zwei verschiedenen Flughöhen und jeweils zwei verschiedenen Anfluggeschwindigkeiten durchgeführt.

Ausbleiben von Positionsdaten des Objektes Die Bildverarbeitung wird so modifiziert, dass nur zu jedem zweiten Frame die Objektposition an die Regelung weitergeleitet wird. Mit diesen Modifikationen werden die Anflugtests aus dem zuvor aufgeführten Testszenario wiederholt.

Äußere Störeinflüsse Der MikroKopter wird mit aktivierter Regelung über dem Objekt platziert. Anschließend wird mit einem Standventilator in einem Abstand von vier Metern eine Störgröße erzeugt. Um den Wind etwas natürlicher zu gestalten wird die Schwenkfunktion des Ventilators aktiviert.

4.3 Systementwurf

4.3.1 Reglerentwurf

In Abbildung 4.3 ist das Blockschaltbild des Regelkreises zu sehen. Der MikroKopter stellt die Regelstrecke dar. Diese setzt sich aus dem Stell- und Störverhalten zusammen. Das Stellverhalten beschreibt die Reaktion des MikroKopters auf Steuerbefehle, das Störverhalten hingegen die Reaktion auf Störeinflüsse, wie z. B. Wind. Das Messglied hat die Aufgabe die Regelgröße (Istwert) y zu erfassen. Die Regelgröße entspricht dabei der Position des MikroKopters relativ zu dem Zielobjekt. Die Mittel, die zur Erfassung der Regelgröße zur Verfügung stehen, sind das Videobild der On-Board Kamera sowie die Sensordaten des MikroKopters. Die Führungsgröße w stellt dagegen die Sollposition des MikroKopters relativ zu dem Zielobjekt dar. Entsprechend der Aufgabenstellung ist diese null. Die Differenz aus der zurückgeführten Regelgröße y und der Führungsgröße w ist die Regelabweichung e . Sie wird dem Regler zugeführt, der daraus die Stellgröße u in Form von Steuerkommandos generiert. Die Steuerkommandos werden an den MikroKopter gesendet, womit der Regelkreis geschlossen wird. Um Rauschen der erfassten Regelgröße entgegenzuwirken befindet sich in deren Rückführung zusätzlich ein Filter. [Unb01a]

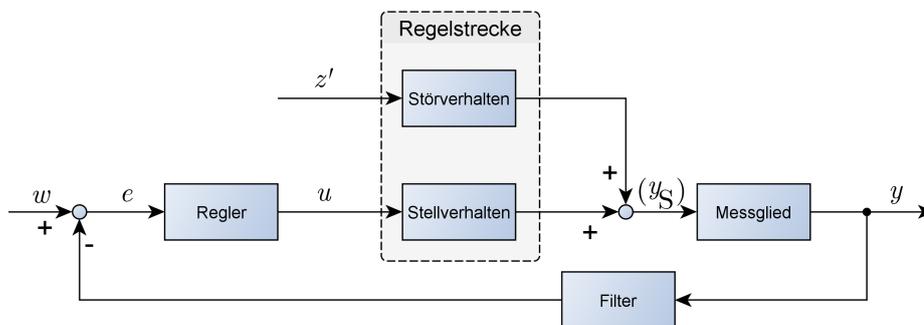


Abbildung 4.3: Blockschaltbild des Regelkreises

Es muss nun eine geeignete Reglerstruktur festgelegt sowie ein passender Regler gefunden werden. Des Weiteren muss eine Gleichung zur Ermittlung der Regelgröße y hergeleitet werden, um daraus die Regelabweichung e berechnen zu können.

Wahl der Reglerstruktur

Der Regler hat die Aufgabe, die in Abbildung 4.4 dargestellte Regelabweichung des MikroKopters zu minimieren. Die Regelabweichung entspricht in diesem Anwendungsfall der negativen Regelgröße y . Um auf die Regelgröße Einfluss zu nehmen stehen die Stellgrößen Nick, Roll, Gier und Gas zur Verfügung. Die Gier-Steuerfunktion kann ignoriert werden, da sie keinen Einfluss auf die Regelabweichung hat. Die Steuerung der Gas-Funktion ist ebenfalls irrelevant für die Aufgabenstellung. Es muss einzig dafür gesorgt werden, dass der MikroKopter eine eingestellte Flughöhe beibehält. Diese Aufgabe übernimmt bereits die im MikroKopter integrierte Höhenregelung. Zielführend für die Aufgabenstellung sind lediglich die Steuerfunktionen Nick und Roll.

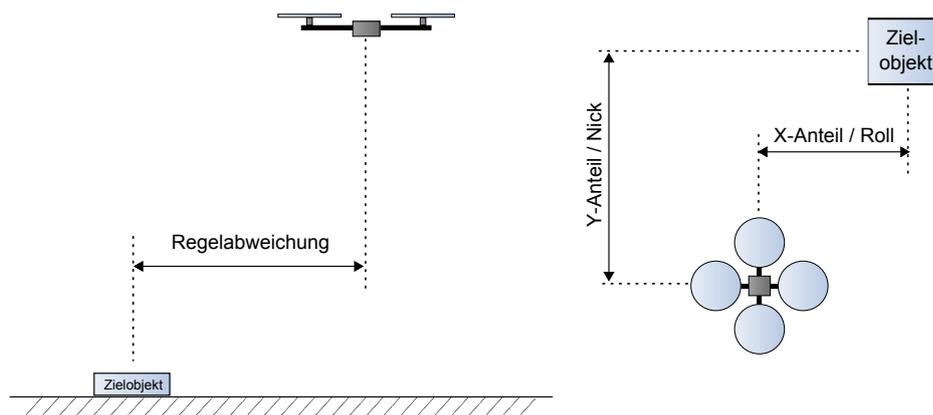


Abbildung 4.4: Darstellung der Regelabweichung und deren Aufteilung in Anteile.

Die Regelgröße kann in zwei getrennt zu betrachtende Anteile aufgeteilt werden. Den *X-Anteil* und den *Y-Anteil*. Dies ist ebenfalls in Abbildung 4.4 dargestellt. Ferner ist festzustellen, dass die zwei relevanten Steuerfunktionen(Stellgrößen) jeweils nur auf einen dieser Anteile Einfluss haben. Es bietet sich demzufolge an, die Regelung in zwei voneinander getrennte Regelkreise aufzuteilen. Jeder dieser Regelkreise besitzt dadurch nur eine Eingangs-(*X-Anteil* / *Y-Anteil*) und eine Ausgangsgröße (*Roll* / *Nick*). Dies wird laut [Unb01b] als Eingrößensystem bezeichnet.

Analyse der Regelstrecke und Auswahl des Reglers

In Abschnitt 3.2 auf Seite 20 wurde bereits auf das Flugverhalten des MikroKopters eingegangen. Um eine Auswahl eines geeigneten Reglers zu treffen wird ergänzend dazu eine Analyse des Stellverhaltens der oben beschriebenen Regelstrecke vorgenommen. Zu diesem Zweck wurde die Sprung- und Impulsantwort, mit Hilfe des Testsystem aus Abschnitt 4.2 auf Seite 47, ermittelt.

Die Impulsantwort, siehe Abbildung 4.5, stellt die Reaktion der Regelgröße auf einen kurzen Impuls der Stellgröße dar. Der Impuls ist ein Steuerkommando mit maximaler Stärke und einer Länge von 250ms. Das resultierende Diagramm stellt den zurückgelegten Weg des MikroKopters in Abhängigkeit von der Zeit dar.

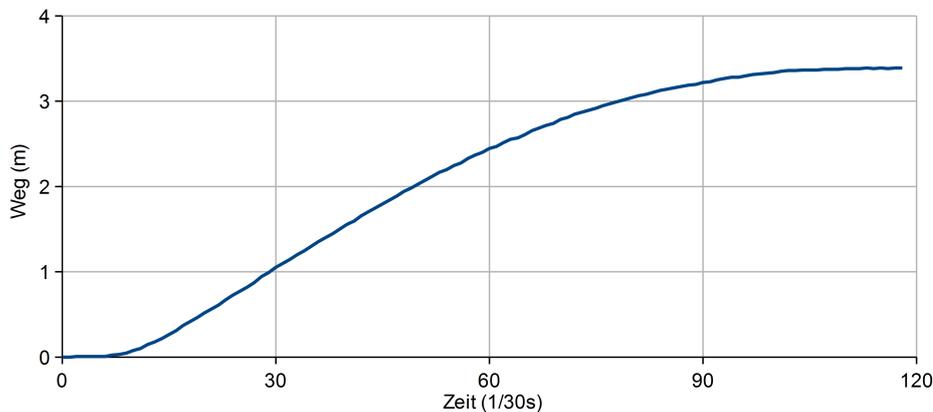


Abbildung 4.5: Impulsantwort auf ein Steuerkommando (250ms, Stärke 127)

In Abbildung 4.6 auf der nächsten Seite ist die Sprungantwort auf ein Steuerkommando mit einer Stärke von ca. 8% zu sehen. Der Abbildung ist zu entnehmen, dass die Regelgröße mit einer Verzögerung auf ein konstantes Steuerkommando reagiert. Nach dieser Verzögerung steigt die Regelgröße gleichbleibend an. Laut [Leo92] handelt es sich um eine Regelstrecke mit verzögerter Integration (IT-Glied).

Um die gegebene Regelstrecke mit vorhandener Integration stabilisieren zu können ist ein Regler mit Vorhaltevermögen nötig. Die Anforderung an die Genauigkeit der Positionierung erfordert zudem einen Regler mit integrierendem Verhalten. Der Proportional-Integral-Differential-Regler (PID-Regler) erfüllt diese Anforderungen.

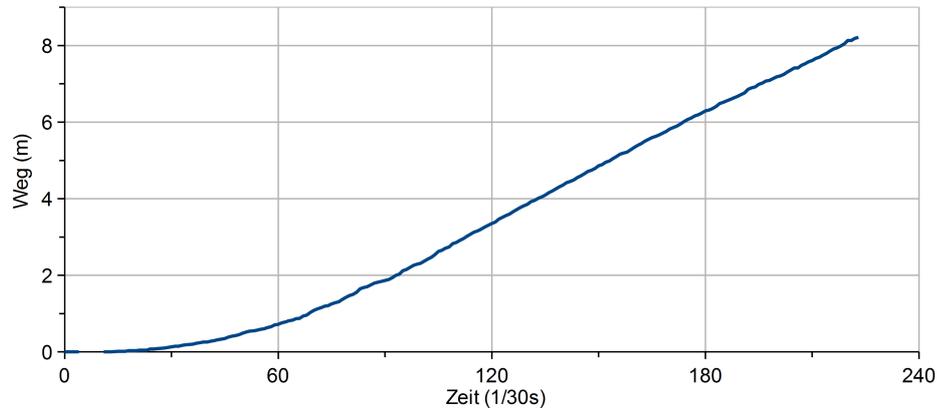


Abbildung 4.6: Sprungantwort auf ein Steuerkommando der Stärke 10

Er ist eine Parallelschaltung der Regler, die im Folgenden beschrieben werden. [Lun12a][Orl07]

P-Regler Die Regelabweichung wird mit einem Verstärkungsfaktor K_p multipliziert und als Stellgröße ausgegeben. Das bedeutet, je größer die Regelabweichung ist, umso größer wird auch die Stellgröße. Insbesondere bei Regelstrecken mit schwacher Dämpfung besteht bei diesem Regler die Gefahr des Aufschwingens.

I-Regler Die Regelabweichung wird über die Zeit aufsummiert (Integral *esum*). Das Integral wird mit einem Faktor K_i versehen und als Stellgröße ausgegeben. Die Stellgröße wird demzufolge so lange erhöht wie eine Regelabweichung besteht. Der I-Regler zeichnet sich dadurch aus, dass er letzte Regelabweichungen eliminieren kann.

PD-Regler Der D-Anteil dieses Reglers reagiert auf die Änderungsgeschwindigkeit (Differential) der Regelabweichung. Das gebildete Differential wird ebenfalls mit einem Verstärkungsfaktor K_d multipliziert. Schnellere Änderungen der Regelabweichung erzeugen ein stärkeres Eingreifen. Dadurch reagiert dieser Regler bereits auf Ankündigungen einer Regelabweichung. Auf die Aufgabenstellung bezogen bewirkt dieser Anteil bereits ein Gegensteuern des MikroKopters bevor das Zielobjekt erreicht wurde.

Aus der Kombination dieser Regler lässt sich die folgende diskrete Reglergleichung bilden[Rob13].

$$u = Kp \cdot e + Ki \cdot Ta \cdot esum + \frac{Kd \cdot (e - ealt)}{Ta} \quad (4.1)$$

Der Parameter Ta der Gleichung stellt die Abtastzeit zwischen den Regelschritten dar. Sie muss bei der Bildung des Integral- und Differenzialanteils berücksichtigt werden. Somit kann der Regler auch auf Änderungen der Abtastzeit reagieren. Dieser Fall tritt ein, wenn das Zielobjekt für einzelne Frames nicht gefunden wird und somit entsprechend viele Regelschritte ausgelassen werden. Solche kurzen Pausenzeiten können mit den zuletzt berechneten Steuerbefehlen überbrückt werden. Die maximal tolerierte Länge der Abtastzeit muss demzufolge begrenzt werden. Wird die maximale Abtastzeit überschritten, kann davon ausgegangen werden, dass sich das Zielobjekt nicht mehr im Blickfeld der Kamera befindet. Der Regler muss dann zurückgesetzt werden. Das bedeutet, es werden das Integral $esum$ und die Regelabweichung des letzten Regelschrittes $ealt$ auf null gesetzt.

Wird das Zielobjekt nach einem Zurücksetzen des Reglers erneut detektiert, so sind zwei Regelschritte notwendig bis der Regler wieder eingreifen kann. Dies ist dadurch begründet, dass die Regelabweichung des letzten Regelschrittes $ealt$ anfangs unbekannt ist.

Herleitung der Regelgröße (neg. Regelabweichung)

Wie eingangs erwähnt, gilt es eine Gleichung zu finden, die als Ergebnis die Regelgröße y liefert. Dabei sind auch die Flughöhe h sowie die Fluglage α des MikroKopters zu berücksichtigen. Abbildung 4.7 dient als Grundlage für die Herleitung. Dort sind ein Modell der Kamera und des Bildsensors (Bildebene) sowie das Zielobjekt zu sehen. Die Regelgröße y ist der Abstand zwischen dem Lot des MikroKopters und dem Zielobjekt.

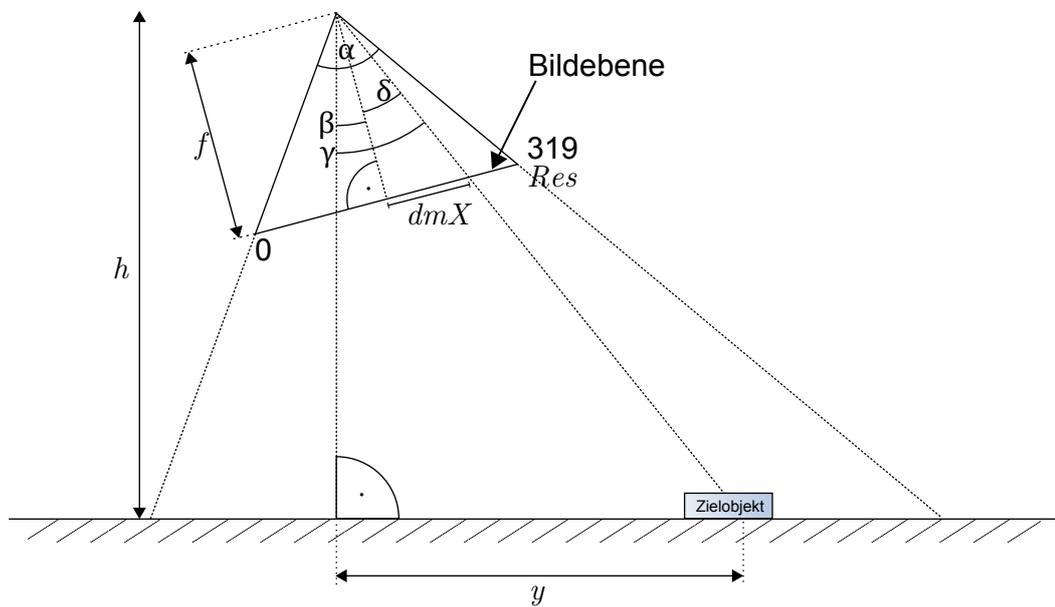


Abbildung 4.7: Kameramodell zur Herleitung der Regelgrößengleichung

Die Regelgröße y berechnet sich aus der Winkeldifferenz γ und der Höhe h wie folgt:

$$y = \tan(\gamma) \cdot h \quad (4.2)$$

Die Winkeldifferenz γ in Formel 4.2 wird aus der Summe der Fluglage β und dem Winkel δ des Zielobjektes zur Kameraachse gebildet. Man erhält somit:

$$y = \tan(\beta + \delta) \cdot h \quad (4.3)$$

Der Winkel δ kann aus der Fokusslänge f und der Pixeldifferenz des Zielobjektes zum Bildmittelpunkt dmX gebildet werden. Eingesetzt in Formel 4.3 erhält man:

$$y = \tan \left[\beta + \arctan \left(\frac{dmX}{f} \right) \right] \cdot h \quad (4.4)$$

Die noch unbekannte Fokusslänge f wird aus der Bildauflösung Res und dem Öffnungswinkel α wie folgt gebildet:

$$f = \frac{0,5Res}{\tan(0,5\alpha)} \quad (4.5)$$

Eingesetzt in Formel 4.4 erhält man:

$$y = \tan \left[\beta + \arctan \left(\frac{dmX \cdot \tan(0,5\alpha)}{0,5Res} \right) \right] \cdot h \quad (4.6)$$

Der letzte unbekannte Wert ist die Pixelabweichung dmX . Sie wird aus der Bildauflösung Res und der detektierten Objektposition im Bild $objX$ berechnet. Eingesetzt in Formel 4.5 erhält man die finale Gleichung zur Ermittlung der Regelgröße y .

$$y = \tan \left[\beta + \arctan \left(\frac{(0,5Res - objX) \cdot \tan(0,5\alpha)}{0,5Res} \right) \right] \cdot h \quad (4.7)$$

4.3.2 Bildverarbeitung

Das zuvor erarbeitete Regelkonzept benötigt die Bildkoordinaten des zu suchenden Zielobjektes. Wie bereits in der Anforderungsanalyse erwähnt, wird in diesem Entwicklungsschritt von einem Objekt ausgegangen, das sich durch seine Merkmale deutlich vom Hintergrund abhebt und eine einfache Kontur besitzt. Ausgehend von diesen Einschränkungen kann die Bildverarbeitungskette, wie in Abbildung 4.8 auf der nächsten Seite dargestellt, ausfallen. Die zugrundeliegenden Überlegungen werden im Folgenden erläutert.

Vorverarbeitung

Das von der Kamera erfasste Bild unterliegt einer starken geometrischen Verzerrung, welche durch das Weitwinkelobjektiv hervorgerufen wird, siehe Abbildung 4.9. Um die ermittelte Objektposition nicht zu verfälschen muss diese beseitigt werden.

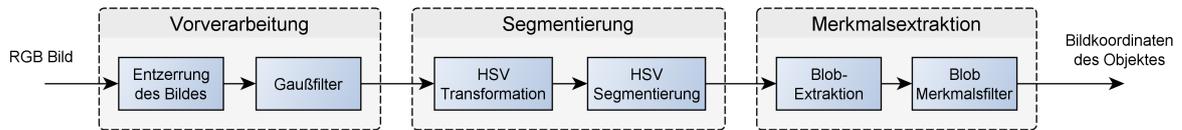


Abbildung 4.8: Bildverarbeitungskette der Objektsuche

Dazu ist es zunächst erforderlich die auftretenden Verzerrungen messtechnisch zu ermitteln. Zu diesem Zweck wird ein Kalibrieremuster mit der Kamera aufgenommen. Anhand des aufgenommenen Musters werden korrespondierende Quell- und Zielkoordinaten ermittelt, aus denen anschließend die intrinsischen Parameter der Kamera berechnet werden. Unter Zuhilfenahme dieser einmalig zu ermittelnden Parameter kann eine Koordinatentransformation durchgeführt werden, um die Verzerrung zu beseitigen. Ausführliche Informationen zu der Thematik befinden sich in [Ope13] und [Sch95]

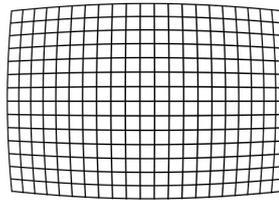


Abbildung 4.9: Illustration der vom Objektiv verursachten Bildverzerrung

Die starke Kompression der Bilddaten verursacht deutlich sichtbare Kompressionsartefakte. In Lichtschwachen Umgebungen neigt der Bildsensor zusätzlich zum Rauschen. Mit Hilfe eines Gaußfilters kann diesen Effekten entgegengewirkt werden.

Farbbasierte Segmentierung

Das zu suchende Objekt hebt sich durch seine Farbinformationen vom Hintergrund ab. Demzufolge bietet sich eine farbbasierte Segmentierung an. Hierfür ist es günstig den gegebenen RGB-Farbraum zunächst in einen HSI-Farbraum zu transformieren. Dieser Farbraum zeichnet sich dadurch aus, dass er, im Gegensatz zum RGB-Farbraum, die Farbinformation von der Helligkeitsinformation trennt. Er setzt sich aus den drei Größen

Farbton (engl.:HUE), Sättigung (engl.: Saturation) und der Helligkeit (engl.:Intensity) zusammen[Bru93].

Durch die vorgenommene Transformation wird eine direkte Filterung anhand der Farb-, Sättigungs- und Helligkeitsinformation des zu suchenden Objektes ermöglicht. Hierfür müssen Intervalle für die drei Größen definiert werden. Die Werte sind so zu wählen, dass sichergestellt wird, dass die HSI-Werte des zu suchenden Objektes innerhalb dieser Intervalle liegen. Anschließend wird das Bild anhand der definierten Intervalle binarisiert. Im Idealfall sind nach der Binarisierung alle Bildpunkte des Objektes weiß und die des Hintergrundes schwarz.

Merkmalsextraktion

Aus dem gewonnenen Binärbild müssen anschließend die Koordinaten des Objektes ermittelt werden. Dies kann mit Hilfe eines Blob-Extraktors erfolgen, der zusammenhängende Bildbereiche(Blobs) extrahiert. Problematisch ist dabei, dass durch die zuvor angewandte Binarisierung nicht sichergestellt werden kann, dass einzig dem Objekt zugehörige Bildpunkte im Binärbild enthalten sind. Im schlimmsten Fall kann es vorkommen, dass sich das Objekt nicht im Blickbereich der Kamera befindet, aber dennoch weiße Bildpunkte im Binärbild vorhanden sind. Um zu erkennen ob es sich bei einem Blob um das gesuchte Objekt handelt, ist es demzufolge nötig weitere Merkmale zu ermitteln. Sinnvolle Merkmale stellen die Fläche des Blobs und seine Kompaktheit dar. Wird aus den ermittelten Werten der Blob mit der größten Fläche ausgewählt und besitzt er zusätzlich eine kleine Kompaktheitszahl, so handelt es sich vermutlich um das gesuchte Objekt. Abschließend muss der Mittelpunkt dieses Blobs bestimmt werden.

4.3.3 Missionsplanung und Abarbeitung

In den letzten beiden Sektionen wurde ein Konzept erarbeitet, mit dem der MikroKopter autonom über dem Zielobjekt positioniert werden kann. Dies erfordert jedoch eine grobe Vorpositionierung um das Zielobjekt in den Blickbereich der Kamera zu bringen. Zu diesem Zweck wird die GPS-Position des Zielobjektes vorgegeben. Unter Einsatz der GPS-Funktionalität der NaviCTRL kann diese Position angefliegen werden.

Missionsplanung

Es muss sichergestellt werden, dass das Zielobjekt überflogen wird. Dazu wird ein Suchmuster, siehe Abbildung 4.10, definiert, das ein bestimmtes Gebiet abdeckt. Um verschiedene Flughöhen zu berücksichtigen muss es möglich sein die im Bild eingezeichnete *Schrittweite* anzupassen. Durch die zusätzliche Angabe einer Schrittzahl wird das Suchgebiet hinreichend genau definiert.

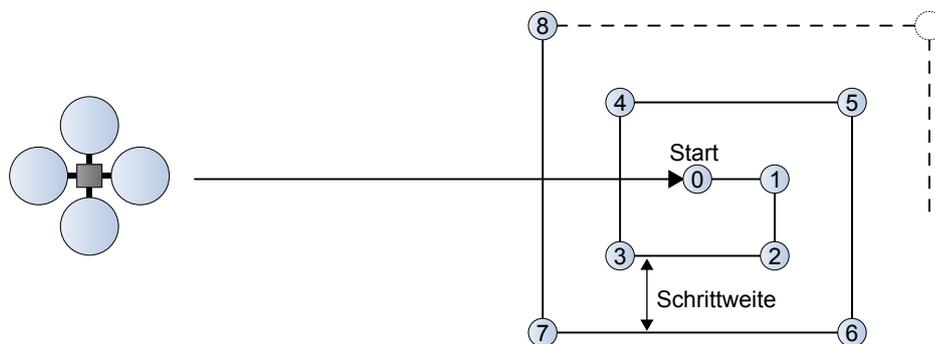


Abbildung 4.10: GPS Suchmuster der Missionsdurchführung

Zur Angabe des Startpunktes im Suchmuster bietet es sich an eine kartenbasierte Variante, ähnlich dem MikroKopterTool-OSD, siehe Abschnitt 3.4 auf Seite 28, zu verwenden. Dazu muss ein Karten- oder Satellitenbild vorliegen, von dem die Eckpunkte in Form von GPS-Koordinaten bekannt sind. Mit Hilfe dieser Eck-Koordinaten kann für jeden beliebigen Bildpunkt des Kartenbildes ein GPS-Koordinatenpaar errechnet werden. Damit ist es möglich, durch einfaches Markieren eines Kartenpunktes, den Startpunkt vorzugeben.

Aus den so definierten Parametern wird anschließend eine Liste von GPS-Koordinaten erstellt. Es kommt die in Abschnitt 3.5.2 auf Seite 36 bereits vorgestellte Datenstruktur zum Einsatz. Die Geschwindigkeit des MikroKopters darf bei Erreichen des Zielobjektes nicht zu hoch sein, da sonst die Gefahr besteht, dass er nicht rechtzeitig abbremst. Durch Setzen des Wertes *Speed* in der Datenstruktur kann dem entgegengewirkt werden.

Missionsdurchführung

Für die Missionsdurchführung ist es erforderlich die GPS-Modi Wegpunkteflug, ComingHome (GPS_CH) und PositionHold (GPS_PH) zu nutzen. Üblicherweise werden diese Modi durch einen dreistufigen Schalter an der Fernsteuerung gewählt. Die drei Stufen des Schalters entsprechen den Funktionen GPS-aus (GPS_FREE), PositionHold und ComingHome. Enthält der Speicher der NaviCTRL eine Wegpunktliste, so wird bei der Schalterstellung ComingHome der Wegpunkteflug eingeleitet. Um in diesem Fall ComingHome zu aktivieren, müssen der Modus GPS_FREE und anschließend ComingHome gewählt werden. Ein erneutes Abfliegen der Wegpunktliste ist erst nach einem Landemanöver oder neuem Übertragen der Wegpunktliste möglich[MW12f]. Um die automatisierte Steuerung der GPS-Modi zu ermöglichen, muss dafür ein serieller Kanal im MikroKopter-Tool konfiguriert werden. Dieser Kanal kann anschließend über die Datenschnittstelle gesteuert werden, siehe Abschnitt 3.5.2 auf Seite 35. Dafür ist es erforderlich die UART-Adresse, siehe Abschnitt 3.5.2 auf Seite 33, auf die FlightCTRL zu legen.

Zusätzlich ist der Fall zu berücksichtigen, dass das Zielobjekt nicht innerhalb des Suchgebietes gefunden wird. Problematisch ist hierbei, dass aufgrund der gewählten UART-Adresse nicht ermittelt werden kann, wann der Wegpunkteflug tatsächlich beendet ist. Das kann umgangen werden, indem die Dauer des Wegpunktefluges, anhand der Parameter des Suchmusters und der eingestellten Fluggeschwindigkeit, geschätzt wird. Die so ermittelte Zeitspanne kann als Abbruchbedingung(Missionstimer) genutzt werden.

Unter Berücksichtigung dieser Überlegungen erfolgt die Missionsabarbeitung, wie in Abbildung 4.11 auf der nächsten Seite dargestellt. Der grün markierte Bereich des Ablaufdiagrammes steht für eine erfolgreiche Missionsausführung. Die dort enthaltene Bedingung „Objekt verloren?“ wird erst erfüllt, wenn das Zielobjekt für eine definierte Zeitspanne nicht mehr detektiert wurde. Dies ist erforderlich um kurzzeitige Objektverluste zu tolerieren, die von starken Steuermanövern hervorgerufen werden können. Bevor ein endgültiger Missionsabbruch stattfindet wird die Objektsuche optional wiederholt.

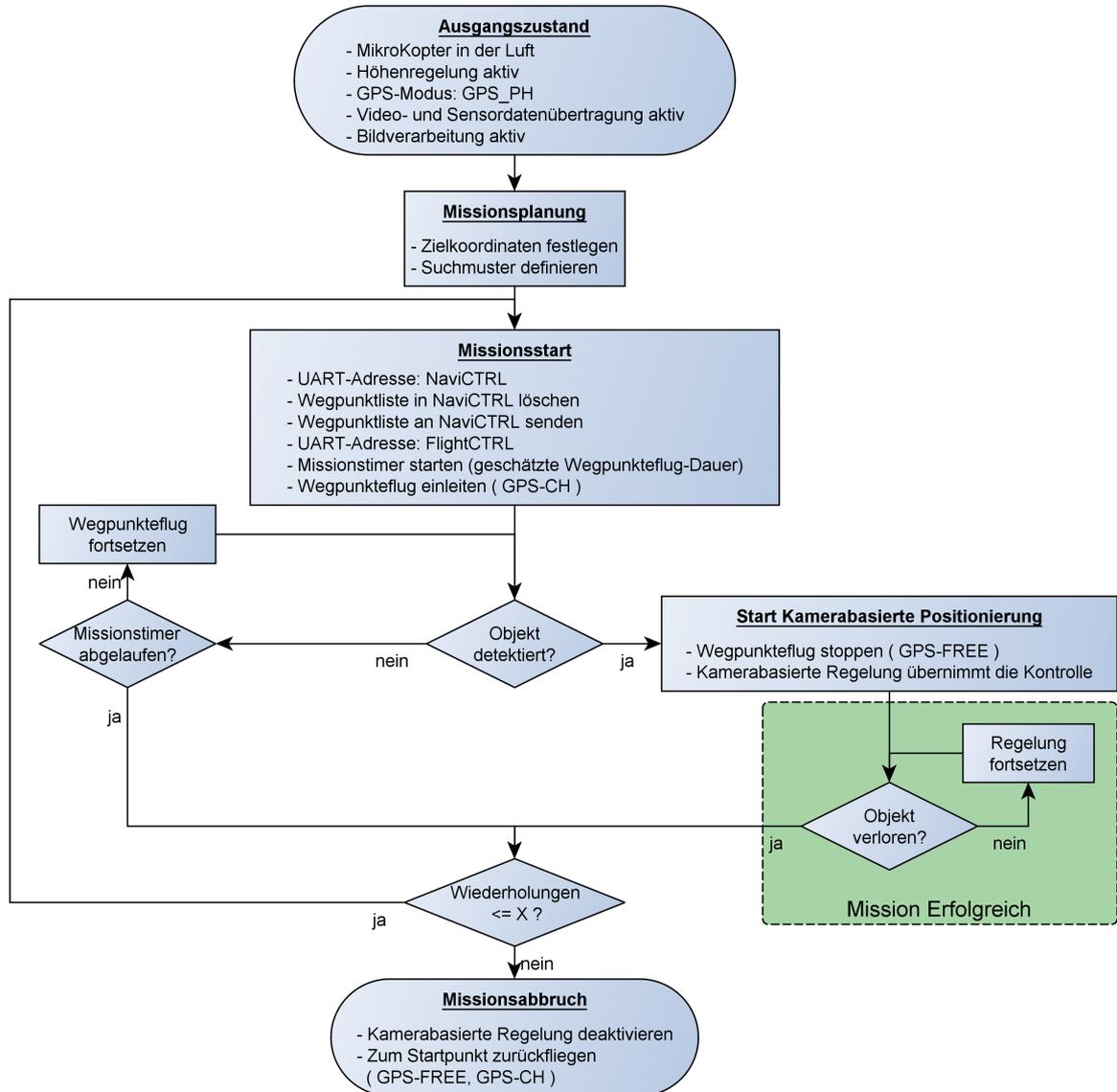


Abbildung 4.11: Ablaufplan der Missionsdurchführung

4.3.4 Modifikationen der FlightCTRL Firmware

In der Anforderungsanalyse, siehe Abschnitt 4.1.1 auf Seite 44, werden Forderungen an die Datenschnittstelle und die Fallback-Möglichkeit gestellt. Um den dort aufgezählten Eigenschaften gerecht zu werden, ist es nötig die Firmware der FlightCTRL zu modifizieren.

Datenübertragung

Aus den bisherigen Kapiteln geht hervor, dass für die kameragestützte Positionierung die Sensordaten des MikroKopters zur Bodenstation übertragen und anschließend Steuerdaten zurückgesendet werden müssen. Die zu diesem Zweck übertragenen Kommandos und Daten sowie deren jeweiligen Datenraten sind in Tabelle 4.1 aufgelistet.

Kommando(ID)	Nutzdaten(Byte)	Übertragen(Byte)	Rate(1/s)	Byterate(1/s)
Externe Kontrolle ('b')	11	22	30	660
Bestätigung ('B')	1	10	30	300
Debug-Daten Abo ('d')	1	10	0,5	5
Debug-Daten ('D')	66	94	30	2820
3785 Byte/s				

Tabelle 4.1: Verwendete Kommandos und deren Größe

Betrachtet man den Aufbau des Kommandos *Externe Kontrolle*, siehe Abschnitt 3.5.2 auf Seite 33, und den Inhalt der *Debug-Daten*, siehe Abschnitt 3.5.2 auf Seite 34, so fällt auf, dass nur ein sehr kleiner Teil dieser Daten tatsächlich benötigt wird. Aus den Debug-Daten werden einzig die Lagedaten *WinkelNick* und *WinkelRoll* sowie die Flughöhe (*Höhenwert*) des MikroKopters benötigt. Über die externe Kontrolle müssen nur Nick- und Roll- Steuerkommandos und der Bestätigungsframe übertragen werden. Es bietet sich demnach an eigene Kommandos und zugehörige Datenstrukturen zu definieren, die nur die benötigten Daten enthalten. Der auf ein Steuerkommando zurückzusendende Bestätigungsframe kann dabei in die minimale Debug-Daten-Struktur integriert werden.

Eine weitere Optimierungsmöglichkeit besteht darin, die Zeitbeschränkung des Debug-Daten Abonnements aufzuheben. Dadurch muss das Kommando ('d') zu deren Abfrage nur einmalig gesendet werden.

Durch diese Anpassungen beschränken sich die regelmäßig zu übertragenden Daten auf zwei verschiedene Datensätze. In Tabelle 4.2 sind diese zusammen mit der resultierenden Datenrate aufgeführt. Es ist zu sehen das die benötigte Datenrate um den Faktor 4,5 gesenkt werden konnte. Das reicht aus, um die geforderten Wiederholungsraten auch in größerer Entfernung einhalten zu können.

Kommando(ID)	Nutzdaten(Byte)	Übertragen(Byte)	Rate(1/s)	Byterate(1/s)
Externe Kontrolle Min	3	10	30	300
Debug-Daten Min	7	18	30	540
840 Byte/s				

Tabelle 4.2: Neu definierte Kommandos und deren Größe

Fallback-Erweiterung

Wie in Abschnitt 3.5.2 auf Seite 33 bereits erwähnt, kann mit einem Schalter der Fernsteuerung die Steuerung über die Datenschnittstelle deaktiviert werden. Damit ist die geforderte Fallback-Möglichkeit bereits teilweise erfüllt.

Die Missionsdurchführung, wie sie auf Seite 59 beschrieben wurde, steuert die GPS-Modi der NaviCTRL über einen seriellen Kanal. Für diese Kanäle wurde seitens der MikroKopter-Entwickler keine Abschaltmöglichkeit über die Fernsteuerung vorgesehen, siehe Abschnitt 3.5.2 auf Seite 35. Wurde der Wegpunkteflug über einen seriellen Kanal aktiviert und bricht anschließend die Datenverbindung ab, besteht keine Möglichkeit mehr den GPS-Flug zu stoppen. Dies ist insbesondere problematisch, wenn fälschlicherweise weit entfernte GPS-Koordinaten übermittelt wurden.

Es ist demnach erforderlich die Funktion des Schalters auf die seriellen Kanäle zu erweitern. Registriert die FlightCTRL ein Deaktivieren des entsprechenden Schalters, muss sie die Werte der seriellen Kanäle zurücksetzen. Damit erhält der Pilot wieder die alleinige Kontrolle über den MikroKopter.

4.3.5 Bodenstation

Die Vielzahl der Aufgaben und die Notwendigkeit, dass die missionskritischen Aufgaben nicht behindert werden dürfen, spricht für eine Aufteilung in mehrere Threads. Es wird

eine Aufteilung in vier Threads vorgenommen. Im Folgenden werden die jeweiligen Aufgaben kurz aufgezählt.

Output-Thread Dieser Thread übernimmt alle Aufgaben die etwas mit dem ausgehenden Datenverkehr auf der seriellen Datenschnittstelle zu tun haben. Das beinhaltet z. B. das Senden von Steuerbefehlen, GPS-Wegpunktlisten und das Umschalten der UART-Adresse. Außerdem soll in diesem Thread die eigentliche Regelung sowie die Missionsdurchführung stattfinden.

Input-Thread Der Thread ist dafür zuständig, alle an der seriellen Datenschnittstelle eingehenden Daten in Empfang zu nehmen. Hier erfolgt auch die Base64 Dekodierung und das Überprüfen der Checksumme. Die dekodierten Daten werden entsprechend ihrer ID in die jeweilige Datenstruktur geschrieben, von wo aus sie von anderen Programmteilen weiterverwendet werden können. Wurde ein neuer Datensatz empfangen wird dies den anderen Threads signalisiert.

BV-Thread Der Thread wartet auf ein eingehendes Bild von der Kamera und verarbeitet dieses. Dazu wird die gesamte Bildverarbeitungskette, wie sie in Abschnitt 4.3.2 auf Seite 55 vorgestellt wurde, abgearbeitet. Außerdem soll dem Output-Thread signalisiert werden, wenn die Berechnung eines Bildes abgeschlossen wurde.

GUI-Thread Hier finden sämtliche Benutzerinteraktionen statt. Dazu zählt auch das Ausgeben der Zwischenschritte der Bildverarbeitung sowie das Anzeigen von Statusmeldungen.

5 Tests

Das im letzten Kapitel erarbeitete Konzept wurde praktisch umgesetzt und muss nun getestet werden. Es wird erläutert, wie das System vor der Inbetriebnahme kalibriert werden muss und wie bei der Bestimmung der Reglerparameter vorgegangen wird. Anschließend wird getestet, wie hoch die Genauigkeit der ermittelten Regelgröße im praktischen Einsatz ist. Abschließend wird das System auf seine Stabilität und Genauigkeit untersucht.

5.1 Parameterjustierung

Das entworfene System besitzt eine Vielzahl an Parametern die justiert werden müssen. Im Folgenden wird kurz auf die wichtigsten von ihnen eingegangen und deren Einstellvorgang erläutert.

Kalibrierung

Wie in Abschnitt 4.3.1 auf Seite 54 dargelegt, benutzt das System zur Ermittlung der Regelgröße die Videodaten der On-Board-Kamera und die Lagedaten des MikroKopters. Wie die Entzerrung des Videobildes erfolgt, wurde bereits in Abschnitt 4.3.2 auf Seite 55 erwähnt. Anschließend muss noch ermittelt werden, was das Objektiv für einen Öffnungswinkel besitzt. Dafür ist es ausreichend die Kamera exakt auf eine Fläche auszurichten und den erfassten Bereich sowie den Abstand zu dieser Fläche zu messen. Aus diesen Werten wird anschließend der Öffnungswinkel berechnet.

Die Lagedaten des MikroKopters werden bei der Berechnung der Regelgröße zur Kompensation der Kameraneigung verwendet. Es ist naheliegend, dass diese Werte genau kalibriert sein müssen. Dazu wird der MikroKopter um einen bekannten Winkel

geneigt. Aus den Lagedaten des MikroKopters und der tatsächlich vorliegenden Lage kann anschließend ein Verstärkungsfaktor hergeleitet werden.

Ebenso wird bei der Kalibrierung des Höhensensors vorgegangen. Zusätzlich zu dem Verstärkungsfaktor muss ein Offset hinzugezogen werden. Dieser Offset ist notwendig, weil durch die Propellerwinde ein Überdruck im Bereich des Luftdrucksensors erzeugt wird.

Regler

Die Regelung besitzt die drei Parameter P , I und D . Deren Bedeutung wurde in Abschnitt 4.3.1 auf Seite 51 bereits erläutert. Zusätzlich ist die Filterbreite der zurückgeführten Regelgröße y zu wählen. Eine verbreitete Methode die Reglerparameter einzustellen ist die von *Ziegler* und *Nichols*. Sie setzt allerdings eine Regelstrecke voraus deren Ausgangsgröße sich, nach einer Änderung der Eingangsgröße, einem statischen Endwert nähert[Lun12b]. Dieser Fall ist hier nicht gegeben, wie die Sprungantwort auf Seite 52 zeigt. Als Folge dessen wird ein empirisches Vorgehen gewählt. Das bedeutet, die Reglerparameter werden schrittweise verändert und die Reaktionen des Systems auf diese Änderungen beobachtet. Anhand dieser Beobachtungen werden die Parameter weiter angepasst.

Das hier verwendete Vorgehen lautet konkret: Alle Parameter des Reglers werden zunächst auf null gesetzt. Der P-Anteil wird langsam erhöht bis der MikroKopter anfängt über dem Zielobjekt hin und her zu schwingen. Dies wird so lange fortgesetzt bis das System gerade noch stabil ist und nicht aus dem Objektbereich rausschwingt. Anschließend wird mit dem D-Anteil fortgefahren, um diese Schwingung zu dämpfen. Wird der D-Anteil zu groß gewählt, kann es passieren das der MikroKopter, aufgrund von Rauschen der zurückgeführten Regelgröße, zu „zucken“ anfängt. Das kann in gewissen Grenzen durch Erhöhen der Filterbreite kompensiert werden. Diese darf jedoch nicht zu groß ausfallen, da sonst die Reaktionsgeschwindigkeit der Regelung zu stark beeinträchtigt wird. Anschließend kann versucht werden den P-Anteil weiter zu erhöhen. Diese Schritte werden so lange wiederholt bis ein zufriedenstellendes Regelverhalten erreicht wird. Abschließend kann durch Erhöhen des I-Parameters versucht werden eine noch bestehende Regelabweichung zu beseitigen.

5.2 Test der Regelgrößenermittlung

Die Qualität der Regelung wird maßgeblich durch die Genauigkeit der ermittelten Regelgröße (Istwert) beeinflusst. Bildet die Regelgröße den tatsächlichen Zustand nicht hinreichend genau ab, kann die Regelung nicht exakt arbeiten. In den folgenden Diagrammen wird die von dem System, mit Hilfe der On-Board-Kamera und der Sensorik, ermittelte Regelgröße mit der tatsächlich vorliegenden verglichen. Zu diesem Zweck wurden Flugtests durchgeführt und der zeitliche Verlauf der Regelgrößen aufgezeichnet. Die tatsächliche Regelgröße wurde mit dem Testsystem ermittelt und wird im Diagramm als *real* bezeichnet. Die berechnete, von der Regelung benutzte, Regelgröße ist im Diagramm als *berechnet* gekennzeichnet. Im Idealfall sind beide Kurven deckungsgleich. Da die berechnete Regelgröße erst ermittelt werden kann, wenn die Kamera das Zielobjekt erfasst, sind die Werte der entsprechenden Kurve zu Beginn jeder Aufzeichnung null.

Abbildung 5.1 zeigt die Auswertung eines langsamen Fluges mit einer Höhe von ca. 4 Metern. Zu Beginn der Messung, bei Sekunde 4, ist eine Abweichung der beiden Kurven zu beobachten. In diesem Bereich führt der MikroKopter ein Bremsmanöver aus, was zu einer Schräglage führt. Im weiteren Verlauf verringert sich die beobachtete Abweichung und verschwindet zum Ende.

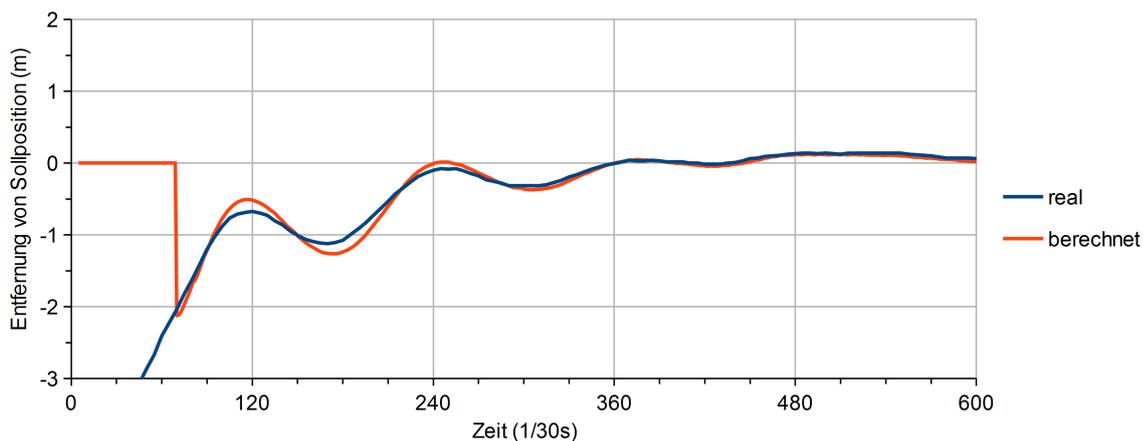


Abbildung 5.1: Vergleich der realen und der berechneten Regelgröße (Langsamer Anflug mit einer Höhe von ca. 4m.)

Die Berechnung der Regelgröße erfolgt unter Berücksichtigung der Flughöhe. Dies wurde überprüft indem der zuvor durchgeführte Test mit einer anderen Flughöhe wiederholt wurde. Wie in Abbildung 5.2 zu sehen ist, besteht ein ähnlicher Zusammenhang zwischen den berechneten und realen Werten wie bei dem zuvor durchgeführten Test. Das ähnlich starke Bremsmanöver verursacht einen ebenso kleinen Fehler der zum Ende abklingt. Die Änderung der Flughöhe hat offensichtlich keinen weiteren Einfluss auf dessen Stärke. Daraus kann geschlossen werden, dass die Berücksichtigung der Flughöhe korrekt erfolgt.

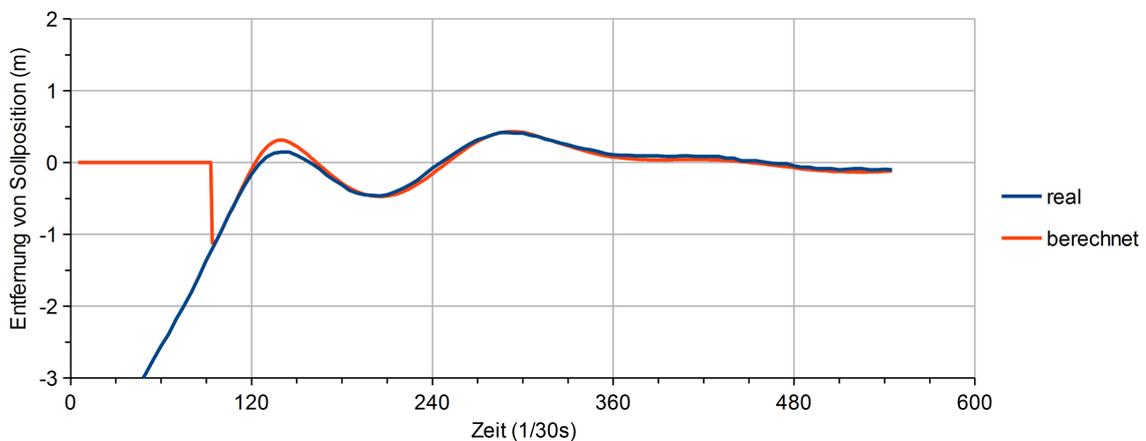


Abbildung 5.2: Vergleich der realen und der berechneten Regelgröße (Langsamer Anflug mit einer Höhe von ca. 2m.)

Ergänzend wurde der erste Flugtest mit einer höheren Geschwindigkeit durchgeführt, um ein stärkeres Bremsmanöver und damit auch eine größere Neigung des MikroKopters zu provozieren. Der daraus resultierende Verlauf der Regelgrößen ist in Abbildung 5.3 auf der nächsten Seite zu sehen. Es ist zu beobachten, dass die Abweichung bei Eintreten des Bremsmanövers deutlich größer ausfällt.

Zusammenfassend kann beobachtet werden, dass die berechnete Regelgröße plausibel zu sein scheint. Verschiedene Flughöhen werden bei der Berechnung ebenso berücksichtigt wie leichte Schräglagen des MikroKopters. Einzig bei starken Bremsmanövern ist ein deutlicher Fehler zu erkennen. Die genaue Ursache dieses Fehlers konnte nicht abschließend geklärt werden. Da der Fehler nur bei hochdynamischen Flugmanövern auftritt, ist es möglich dass er durch fehlerhafte Sensorwerte verursacht wird.

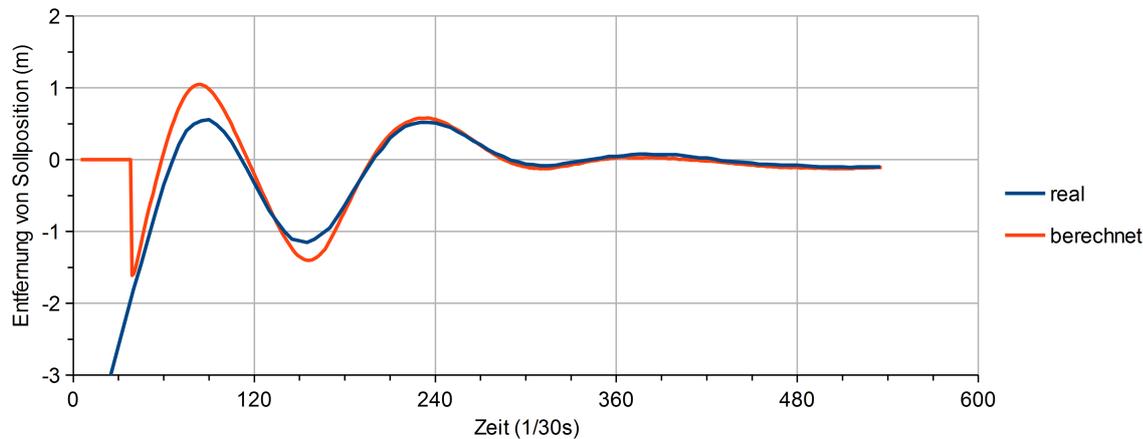


Abbildung 5.3: Vergleich der realen und der berechneten Regelgröße (Schneller Anflug mit einer Höhe von ca. 4m.)

5.3 Ergebnisse

Im Folgenden werden die erzielten Ergebnisse der entwickelten Regelung vorgestellt und analysiert. Es werden dabei insbesondere die Testszenarien berücksichtigt, die in Abschnitt 4.2.2 auf Seite 47 definiert wurden. Die dort beschriebene Vorgehensweise der Testdurchführung findet hier Anwendung.

Die Ergebnisse wurden unter Zuhilfenahme des Testsystems aufgezeichnet und ausgewertet. Die gezeigten Diagramme stellen auf der Y-Achse die relative Position des MikroKopters zu dem Zielobjekt dar. Es werden jeweils zwei Verläufe gegenübergestellt.

Die Fluggeschwindigkeit wird mit *langsam* bzw. *schnell* betitelt, da sie nur ein grober Richtwert ist und nicht genau reproduziert werden kann. Ein langsamer Flug entspricht ca. 5km/h, ein schneller ca.10km/h. Ähnlich wird bei der Flughöhe vorgegangen. Die Angabe *hoch* entspricht einer Flughöhe von ca. 4 Metern und *tief* einer Höhe von etwa 2 Metern.

Robustheit gegenüber Geschwindigkeit

Mit diesem Test soll die Reaktion des Systems auf verschiedene Anfluggeschwindigkeiten untersucht werden. Die entsprechenden Flugbahnen sind in Abbildung 5.4 auf der nächsten Seite zu sehen. Dem Diagramm ist zu entnehmen, dass bei beiden An-

fluggeschwindigkeiten die Position des MikroKopters nach ca. 10 Sekunden stabilisiert wurde.

Dennoch weisen beide Kurven leichte Unterschiede auf. Bei langsamer Anfluggeschwindigkeit ist zu beobachten, dass der MikroKopter bereits kurz vor Erreichen des Zielobjektes zum Stillstand kommt und sich anschließend wieder ein Stück entfernt. Dieses Verhalten deutet auf einen zu hoch gewählten D-Parameter. Ein Senken dieses Parameters hätte allerdings zur Folge, dass der MikroKopter bei hoher Geschwindigkeit nicht rechtzeitig abbremst und über das Objekt hinaus fliegt.

Der Verlauf bei hoher Geschwindigkeit zeigt dagegen ein eher zu schwach gedämpftes Verhalten. Der MikroKopter fliegt zunächst leicht über die Zielposition hinaus, um anschließend wieder zu weit zurück zu fliegen. Dies deutet auf einen zu kleinen D-Parameter oder einen zu hoch gewählten P-Parameter hin. Tatsächlich hat eine Sichtung der Steuerwerte jedoch ergeben, dass das zu starke Gegensteuern durch den D-Anteil verursacht wird. Dies kann durch Latenzen in der Video- und Datenübertragung begründet sein.

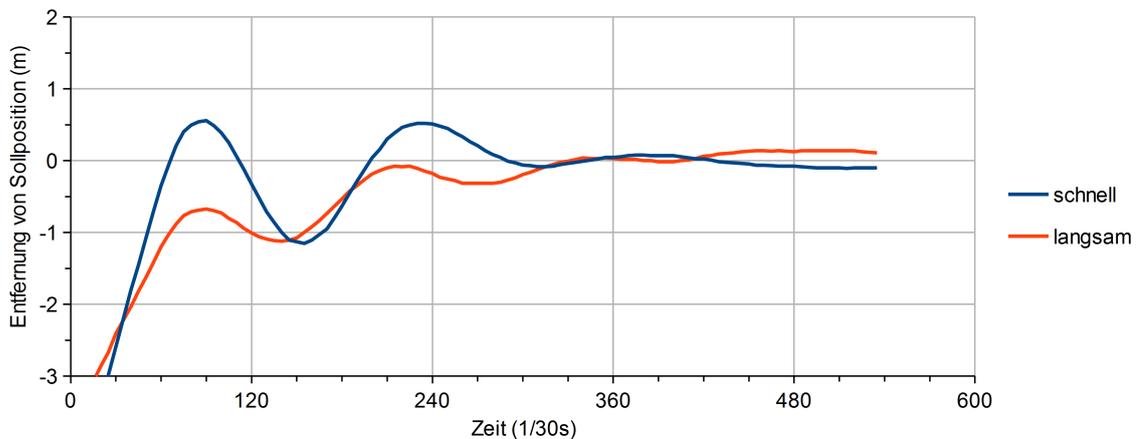


Abbildung 5.4: Vergleich zweier Anfluggeschwindigkeiten bei hoher Flughöhe

Robustheit gegenüber Flughöhe

Dieser Test soll aufdecken, wie das System auf verschiedene Anflughöhen bei gleichbleibender Geschwindigkeit reagiert. Das Problem bei sehr kleinen Flughöhen ist die

kleinere Reaktionszeit die zur Verfügung steht um die Fluggeschwindigkeit abzubauen. Es ist zu erwarten, dass der MikroKopter stärker über das Ziel hinaus fliegt oder es sogar ganz verfehlt.

Die aufgezeichneten Ergebnisse, siehe Abbildung 5.5, bestätigen dies. Es ist zu sehen, dass der MikroKopter bei geringer Flughöhe zunächst etwas weiter über das Zielobjekt hinaus fliegt. Eine weitere Auffälligkeit ist, dass nach dem Bremsmanöver bei tiefer Flughöhe nicht so stark zurückgesteuert wurde wie bei hoher Flughöhe. Eine genauere Analyse hat ergeben, dass das Zielobjekt bei der geringeren Flughöhe bereits für einen kurzen Moment aus dem Blickfeld der Kamera geraten ist. Der Regelkreis wurde nach dem ersten Überfliegen des Objektes demzufolge kurz getrennt, was zu einem verminderten Gegensteuern führte.

Aus den Messungen geht dennoch hervor, dass in beiden Fällen die Position des MikroKopters sauber ausgeregelt wurde. Bis auf den kurzen Objektverlust gab es keine wesentlichen Unterschiede zwischen den Durchläufen. Die Robustheit gegenüber verschiedenen Flughöhen ist also in gewissen Grenzen gegeben. Dazu kommt, dass in der Praxis nur selten mit so geringen Flughöhen zu rechnen ist.

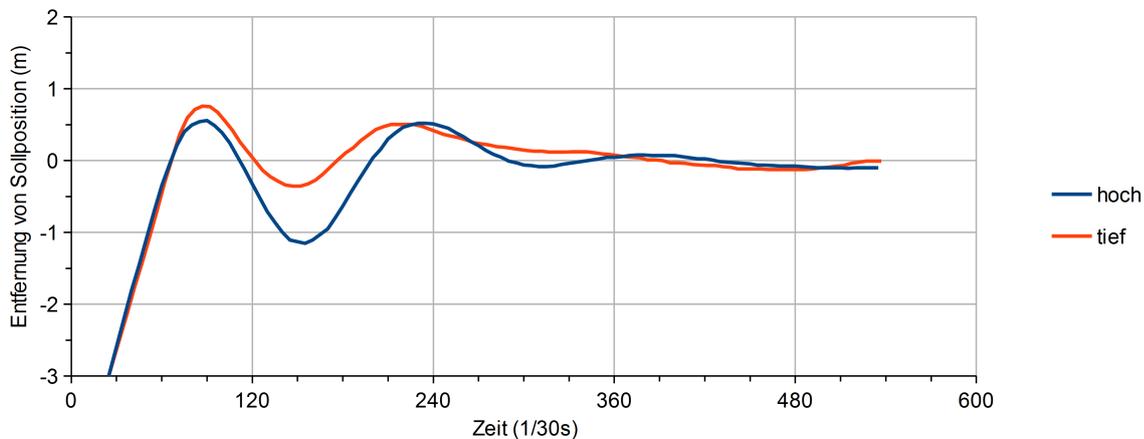


Abbildung 5.5: Vergleich zweier Flughöhen bei schneller Anfluggeschwindigkeit

Robustheit gegenüber äußeren Einflüssen

Wie in Abschnitt 4.2.2 auf Seite 47 beschrieben, wurde bei diesem Test eine Störgröße mit Hilfe eines schwenkenden Standventilators erzeugt. Somit kann überprüft werden,

ob das System auch bei einer wechselnden Störgröße stabil bleibt oder sich aufschwingt. Um die Auswirkungen der Störgröße besser beurteilen zu können wurde der gleiche Test mit ausgeschaltetem Ventilator wiederholt. Die Flughöhe betrug bei beiden Tests ca. 3 Meter.

Wie in Abbildung 5.6 zu sehen ist, beträgt die Schwankung um die Sollposition, ohne eine zusätzliche Störgröße, ca. ± 20 Zentimeter. Mit Zuschalten der Störgröße verdoppelt sich diese auf ca. ± 40 Zentimeter. Das System bleibt weiterhin in einem stabilen Zustand. Da keine weiteren Informationen über die erzeugte Windstärke vorliegen, kann keine Aussage über die Stabilität gegenüber in der Natur vorkommender Winde gemacht werden.

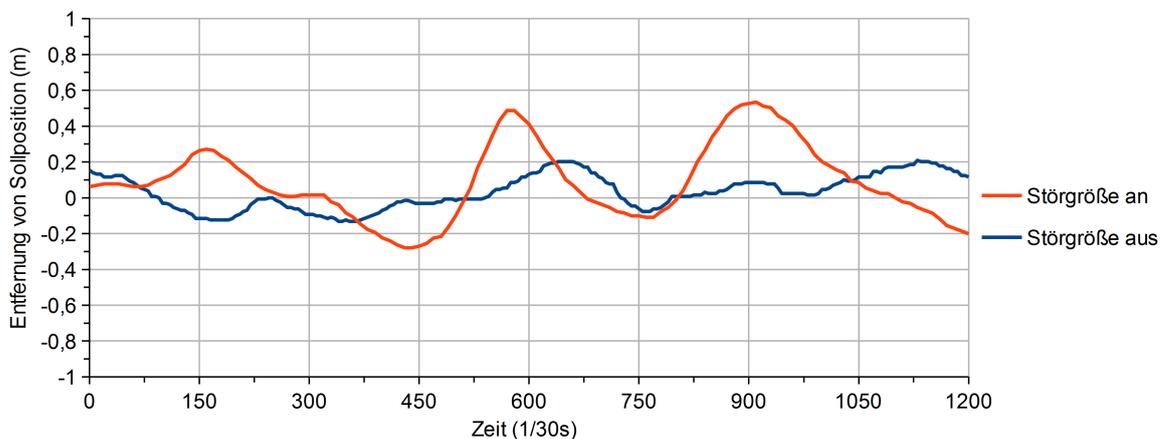


Abbildung 5.6: Genauigkeit der Positionsstabilisierung mit und ohne Störgröße

Robustheit gegenüber kurzer Aussetzer

In Abschnitt 4.3.1 auf Seite 51 wurde beschrieben, wie die Regelung damit umgeht, wenn das Zielobjekt für einzelne Frames nicht gefunden wird. Dieser Fall wurde getestet indem ein künstlicher Objektverlust in jedem zweiten Frame erzeugt wurde. Mit diesen Modifikationen wurde ein Flugtest mit hoher Geschwindigkeit sowie hoher Flughöhe durchgeführt. In Abbildung 5.7 auf der nächsten Seite werden die Resultate einem gleichen Flugtest ohne diese Modifikationen gegenübergestellt. Es ist zu sehen, dass der simulierte *Fehler* in den ersten 10 Sekunden nahezu keine Auswirkungen

zeigt. Lediglich im weiteren Verlauf ist ein leichtes Nachschwingen sowie eine länger andauernde Abweichung von der Sollposition zu beobachten. Versagt die Detektion der Objektposition tatsächlich nur für einzelne Frames, so wird dies demzufolge kaum eine Auswirkung auf die Qualität der Positionierung haben.

Interessant sind die gemachten Beobachtungen vor allem für die Umsetzung einer komplexeren Bildverarbeitung, die eine längere Verarbeitungszeit benötigt.

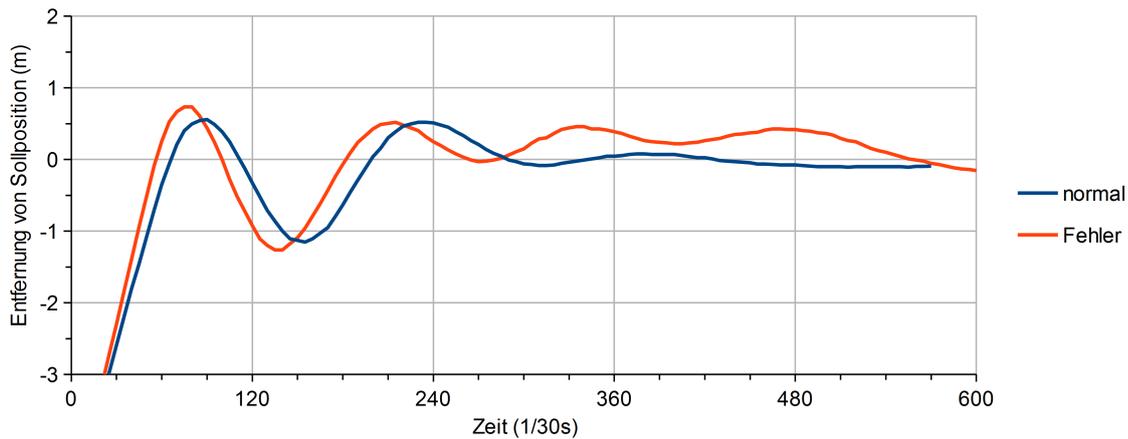


Abbildung 5.7: Test der Robustheit gegenüber Objektverlust

6 Zusammenfassung

In der vorliegenden Arbeit wurden die allgemeinen Grundlagen von Drehflüglern und Quadroptern erläutert. Anschließend wurde die Hard- und Software eines Quadropters vom Typ MikroKopter analysiert. Darauf aufbauend wurde ein System konzipiert und praktisch umgesetzt, das dazu in der Lage ist den MikroKopter autonom, unter Verwendung der On-Board-Kamera, über einem Objekt zu positionieren. Das entwickelte System wurde anschließend im Hinblick auf seine Stabilität und Genauigkeit untersucht. Im folgenden Kapitel werden die Ergebnisse nochmals zusammengefasst und die aufgetretenen Probleme erläutert. Abschließend wird ein Ausblick auf mögliche Erweiterungen des Systems gegeben.

6.1 Auswertung

Die Testergebnisse haben gezeigt, dass die verwirklichte Positionierung bereits sehr stabil ist. Eine endgültige Aussage über die Stabilität und Robustheit kann jedoch nicht getroffen werden, da die Tests nur einen sehr kleinen Höhen- und Geschwindigkeitsbereich abdecken. Dies ist vor allem durch den beengten Testraum begründet, der zur Verfügung stand. Ausgiebige Tests im Freifeld kamen aufgrund der Wetterbedingungen nicht in Frage.

Überhaupt kann gesagt werden, dass die Testdurchführung und Auswertung die meisten Probleme bereitet hat. Es war nur schwer möglich Tests unter gleichen Bedingungen zu wiederholen. Dies gilt vor allem für die Anfluggeschwindigkeit, die möglichst genaue Einhaltung der Flugposition entlang der Teststrecke sowie das Einhalten einer exakten Flughöhe. Erschwerend kam hinzu, dass die weitere Auswertung der aufgenommenen Flugbahnen mit dem Bewertungstool sehr zeitaufwendig war.

Die Auslagerung der Datenverarbeitung auf eine Bodenstation brachte nicht nur die erhofften Vorteile. Bei der Übertragung der Sensor-, Steuer- und Videodaten traten unerwartete Probleme auf. Aufgrund von Hintergrundprozessen, welche regelmäßige W-Lan Netzwerkscans durchführten, kam es zu sporadischen Bildverlusten bei der Videoübertragung. Zusätzlich haben die durch die Datenübertragung verursachten Latenzen die Regelung negativ beeinflusst. Eine genaue Analyse der Stärke der Auswirkungen muss noch erfolgen.

Abschließend konnte gezeigt werden, dass die in der Einleitung genannten Einschränkungen die mit einer reinen GPS-basierten Lösung einhergehen, umgangen werden können. Durch die erreichte Präzision der Positionierung und der Unabhängigkeit von GPS wäre außerdem ein Einsatz in Gebäuden denkbar.

6.2 Ausblick

Mit dem System wurde eine gute Basis für weitere Entwicklungen geschaffen. Zunächst müssen jedoch noch Flugtests im Freien durchgeführt werden, um die Robustheit gegenüber höheren Geschwindigkeiten und Flughöhen zu testen. Mit Hilfe dieser Tests können die Reglerparameter weiter optimiert werden. Ein ausführlicher Test der Missionsdurchführung steht ebenfalls noch aus. Hierbei muss insbesondere überprüft werden ob die berechnete Flugzeit, welche bei dem Missionstimer Verwendung findet, mit der tatsächlichen Flugzeit übereinstimmt. Des Weiteren ist es sinnvoll die Ursache des Fehlers zu finden, der bei der Ermittlung der Regelgröße zu beobachten war, siehe Abschnitt 5.2 auf Seite 67.

Das Testsystem sollte ebenfalls überdacht werden. Das Bewertungstool könnte dahingehend verbessert werden, dass es die aufgezeichnete Flugbahn selbstständig, mit Hilfe von bildverarbeitenden Verfahren ermittelt. Dies kann durch eine spezielle Markierung, welche am MikroKopter angebracht wird, erleichtert werden. Außerdem ist es denkbar die Flugbahn des MikroKopters aus zwei verschiedenen Perspektiven aufzuzeichnen. Dadurch wäre es möglich die Flugposition im Raum zu bestimmen. Ein Testflug müsste somit nicht mehr exakt entlang der Teststrecke stattfinden. Kombiniert mit einer automatischen Auswertung der so aufgezeichneten Videos würde dies eine erhebliche Verbesserung gegenüber dem aktuell bestehenden Testsystem darstellen.

Die Positionsregelung selbst bietet ebenfalls viel Raum für Verbesserungen. Hier ist es ratsam zumindest eine adaptive Regelung umzusetzen, um besser auf verschiedene Fluggeschwindigkeiten und Flughöhen zu reagieren. Zusätzlich sollten weitere Regelkonzepte getestet werden. Vielversprechend könnte der Einsatz einer Zustands- oder Fuzzyregelung sein.

Die Bildverarbeitung wurde in dieser Arbeit sehr einfach gehalten. Sie könnte dahingehend erweitert werden, dass auch frei definierbare Objekte detektiert werden können. Dies könnte zum Beispiel mit einem Template Matching Verfahren erfolgen. Die Positionsstabilisierung könnte auch unabhängig von bestimmten Objekten erfolgen. Interessant sind hierfür z. B. Verfahren die den optischen Fluss bestimmen.

Längerfristig sollte die Datenverarbeitung auf eine On-Board-Lösung portiert werden. Der MikroKopter wäre somit nicht mehr auf die Funkverbindungen angewiesen und könnte vollständig autonom über einen größeren Radius operieren.

Tabellenverzeichnis

3.1	Technische Daten des verwendeten MikroKopters	21
3.2	Technische Daten der FlightCTRL[MW11a][MW13]	23
3.3	Technische Daten der NaviCTRL[MW12f][MW12e]	24
3.4	Technische Daten des Antriebes[MW11b]	26
3.5	Technische Daten des Wi.232 Funkmoduls[MW12g][Rad07]	26
3.6	Technische Daten der Kamera[COM10]	27
3.7	Adressen der MikroKopter Baugruppen	31
3.8	Kommandos und die zugehörigen ID's der MikroKopter Baugruppen. .	32
3.9	Zieladressen und „magic packet“ für das Umschalten der UART-Adresse	33
3.10	Inhalt des Kommandos <i>Externe Kontrolle</i>	34
3.11	Antwort auf ein Debug-Daten Kommando	34
3.12	Auszug aus dem Inhalt des <i>Analog Arrays</i> der Debug-Daten Antwort der jeweiligen Baugruppen	35
3.13	Aufbau des Kommandos <i>Serielle Kanäle</i>	36
3.14	Aufbau der Wegpunktstruktur	36
4.1	Verwendete Kommandos und deren Größe	61
4.2	Neu definierte Kommandos und deren Größe	62

Abbildungsverzeichnis

2.1	Übliche Rotoranordnungen[Bit05b]	12
2.2	Roll-Pitch-Yaw Winkel von Luftfahrzeugen[Wik12c]	14
2.3	Multikopter-Bauarten und deren Rotoranordnungen	15
2.4	Drehrichtung der Rotoren eines Quadropters	16
3.1	MikroKopter	19
3.2	Schematische Darstellung der Komponenten des MikroKopters	22
3.3	Platinenturm des MikroKopters	23
3.4	Benutzeroberfläche des MikroKopter-Tools	29
3.5	Benutzeroberfläche des MikroKopterTool-OSD	30
3.6	Übertragungsprotokoll der Seriellen Schnittstelle der NaviCTRL	31
3.7	Codeauszug der Base64 Implementierung des MikroKopters	37
3.8	Base64 Kodierung	38
3.9	Implementierung der Checksummenbildung des MikroKopters	39
3.10	Verarbeitung von Nick-, Roll- und Gierkommandos der drei Steuerquellen	40
3.11	Verarbeitung des <i>Gas</i> Steuerwertes der verschiedenen Quellen	41
4.1	Grobkonzept	43
4.2	Draufsicht der Teststrecke	48
4.3	Blockschaltbild des Regelkreises	49
4.4	Darstellung der Regelabweichung und deren Aufteilung in Anteile.	50
4.5	Impulsantwort auf ein Steuerkommando (250ms, Stärke 127)	51
4.6	Sprungantwort auf ein Steuerkommando der Stärke 10	52
4.7	Kameramodell zur Herleitung der Regelgrößengleichung	54
4.8	Bildverarbeitungskette der Objektsuche	56
4.9	Illustration der vom Objektiv verursachten Bildverzerrung	56
4.10	GPS Suchmuster der Missionsdurchführung	58
4.11	Ablaufplan der Missionsdurchführung	60
5.1	Vergleich der realen und der berechneten Regelgröße (Langsamer Anflug mit einer Höhe von ca. 4m.)	67
5.2	Vergleich der realen und der berechneten Regelgröße (Langsamer Anflug mit einer Höhe von ca. 2m.)	68

5.3	Vergleich der realen und der berechneten Regelgröße (Schneller Anflug mit einer Höhe von ca. 4m.)	69
5.4	Vergleich zweier Anfluggeschwindigkeiten bei hoher Flughöhe	70
5.5	Vergleich zweier Flughöhen bei schneller Anfluggeschwindigkeit	71
5.6	Genauigkeit der Positionsstabilisierung mit und ohne Störgröße	72
5.7	Test der Robustheit gegenüber Objektverlust	73

Literatur

- [Bit05a] Walter Bittner. *Flugmechanik der Hubschrauber*. Berlin/Heidelberg: Springer Verlag, 2005, S. 174–186.
- [Bit05b] Walter Bittner. *Flugmechanik der Hubschrauber*. Berlin/Heidelberg: Springer Verlag, 2005, S. 1–18.
- [Bru93] Bernd Brueckmann. *Bildverarbeitung und optische Meßtechnik*. München: Franzis Verlag, 1993, S. 57–61.
- [Bü10] Roland Büchi. *Faszination Quadropter*. Baden-Baden: Verlag für Technik und Handwerk, 2010.
- [COM10] AXIS COMMUNICATIONS. *AXIS M10 Netzwerk-Kamera-Serie*. [Online; Stand 24. Februar 2013, 4:15]. 2010. URL: http://www.axis.com/de/files/datasheet/ds_m10_40706_de_1009_lo.pdf.
- [Jos06] Simon Josefsson. *The Base16, Base32, and Base64 Data Encodings*. RFC 4648, 2006. URL: <http://tools.ietf.org/pdf/rfc4648.pdf>.
- [KM12] Niels Klußmann und Arnim Malik. *Lexikon der Luftfahrt*. Berlin/Heidelberg: Springer Verlag, 2012, S. 53–64.
- [Leo92] Werner Leonhard. *Einführung in die Regelungstechnik*. Braunschweig/Wiesbaden: Vieweg Verlag, 1992, S. 31–33.
- [Lun12a] Jan Lunze. *Automatisierungstechnik*. München: Oldenbourg Verlag, 2012, S. 230–235.
- [Lun12b] Jan Lunze. *Automatisierungstechnik*. München: Oldenbourg Verlag, 2012, S. 239–246.
- [MW10] Mikrokopter-Wiki. *Pressemappe*. [Online; Stand 21. Februar 2013, 0:01]. 2010. URL: <http://mikrokopter.de/ucwiki/Mikrokopter-FAQ-Pressemappe?action=recall&rev=8>.
- [MW11a] Mikrokopter-Wiki. *Aufbau einer unbestückten Flight-Ctrl*. [Online; Stand 21. Februar 2013, 2:15]. 2011. URL: <http://www.mikrokopter.de/ucwiki/FCAufbauUnbestueckt?action=recall&rev=60>.
- [MW11b] Mikrokopter-Wiki. *BrushlessCtrl*. [Online; Stand 21. Februar 2013, 4:56]. 2011. URL: <http://mikrokopter.de/ucwiki/BrushlessCtrl?action=recall&rev=127>.

- [MW11c] Mikrokoetter-Wiki. *MikroKopterEinstieg - Etwas Theorie*. [Online; Stand 2. Oktober 2012, 14:34]. 2011. URL: <http://mikrokoetter.de/ucwiki/MikroKopterEinstieg?action=recall&rev=156>.
- [MW12a] Mikrokoetter-Wiki. *CareFree*. [Online; Stand 21. Februar 2013, 3:49]. 2012. URL: <http://www.mikrokoetter.de/ucwiki/CareFree?action=recall&rev=37>.
- [MW12b] Mikrokoetter-Wiki. *MikroKopter-Software*. [Online; Stand 23. Februar 2013, 1:18]. 2012. URL: <http://mikrokoetter.de/ucwiki/Software?action=recall&rev=18>.
- [MW12c] Mikrokoetter-Wiki. *MikroKopterTool*. [Online; Stand 22. Februar 2013, 23:28]. 2012. URL: <http://mikrokoetter.de/ucwiki/MikroKopterTool?action=recall&rev=206>.
- [MW12d] Mikrokoetter-Wiki. *MikroKopterTool-OSD*. [Online; Stand 23. Februar 2013, 23:30]. 2012. URL: <http://mikrokoetter.de/ucwiki/MikroKopterTool-OSD?action=recall&rev=45>.
- [MW12e] Mikrokoetter-Wiki. *MK3Mag*. [Online; Stand 21. Februar 2013, 2:29]. 2012. URL: <http://www.mikrokoetter.de/ucwiki/MK3Mag?action=recall&rev=111>.
- [MW12f] Mikrokoetter-Wiki. *NaviCtrl V1.1*. [Online; Stand 21. Februar 2013, 3:28]. 2012. URL: http://www.mikrokoetter.de/ucwiki/NaviCtrl_1.1?action=recall&rev=94.
- [MW12g] Mikrokoetter-Wiki. *RadioTronix Wi.232*. [Online; Stand 22. Februar 2013, 5:42]. 2012. URL: <http://www.mikrokoetter.de/ucwiki/RadioTronix?action=recall&rev=75>.
- [MW12h] Mikrokoetter-Wiki. *SerialProtocol*. [Online; Stand 24. Februar 2013, 2:12]. 2012. URL: <http://www.mikrokoetter.de/ucwiki/en/SerialProtocol?action=recall&rev=35>.
- [MW13] Mikrokoetter-Wiki. *FlightCtrl V2.0*. [Online; Stand 21. Februar 2013, 2:18]. 2013. URL: http://www.mikrokoetter.de/ucwiki/FlightCtrl_ME_2_0?action=recall&rev=37.
- [Ope13] OpenCV. *Camera Calibration and 3D Reconstruction*. [Online; Stand 1. März 2013, 3:34]. 2013. URL: http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#bouguetmct.

- [Org03] International Civil Aviation Organization. *Annex 7 to the Convention on International Civil Aviation, Aircraft Nationality and Registration Marks*. [Online; Stand 20. Februar 2013, 6:09]. 2003. URL: http://www.bazl.admin.ch/dokumentation/grundlagen/02643/index.html?lang=de&download=NHZLpZe;g7t,lnp6IONT!U04212Z6ln:1acy4Zn4Z2;qZpn02!Yuq2Z6:gpJCDeoB;8fmym1!62epYb:g2c_JjK;bNoKS!n6A--.
- [Orl07] Peter F. Orlowski. *Praktische Regeltechnik*. Berlin/Heidelberg: Springer Verlag, 2007, S. 60–109.
- [Rad07] Radiotronix. *Wi232EUR User's Manual*. [Online; Stand 22. Februar 2013, 5:25]. 2007. URL: http://www.radiotronix.com/datasheets/new/eur_um.pdf.
- [Rob13] RoboterNETZ. *Regelungstechnik*. [Online; Stand 28. Februar 2013, 21:27]. 2013. URL: <http://www.rn-wissen.de/index.php?title=Regelungstechnik&oldid=21546>.
- [Sch95] Reiner Schmid. *Industrielle Bildverarbeitung*. Braunschweig/Wiesbaden: Vieweg Verlag, 1995, S. 174–186.
- [Unb01a] Heinz Unbehauen. *Regelungstechnik I*. Wiesbaden: Vieweg Verlag, 2001, S. 11–13.
- [Unb01b] Heinz Unbehauen. *Regelungstechnik I*. Wiesbaden: Vieweg Verlag, 2001, S. 31.
- [vol13] e volo. *Firmenpräsentation*. [Online; Stand 20. Februar 2013, 22:03]. 2013. URL: <http://www.e-volo.com/de/>.
- [Wik12a] Wikipedia. *Drehflügler*. [Online; Stand 13. September 2012, 13:46]. 2012. URL: <http://de.wikipedia.org/w/index.php?title=Drehfl%C3%BCgler&stableid=106564946>.
- [Wik12b] Wikipedia. *European Geostationary Navigation Overlay Service*. [Online; Stand 9. September 2012, 22:43]. 2012. URL: http://de.wikipedia.org/w/index.php?title=European_Geostationary_Navigation_Overlay_Service&stableid=105432043.
- [Wik12c] Wikipedia. *Roll-Nick-Gier-Winkel*. [Online; Stand 17. September 2012, 21:10]. 2012. URL: <http://de.wikipedia.org/w/index.php?title=Roll-Nick-Gier-Winkel&stableid=107718320>.
- [Wik13] Wikipedia. *Quadrocopter*. [Online; Stand 7. Februar 2013, 15:37]. 2013. URL: <http://de.wikipedia.org/w/index.php?title=Quadrocopter&stableid=112694214>.