

Der roflCopter - Ein Heli dreht durch
Jugendforscht Technik 2017

Frederik Dunschen

15. Januar 2017

Zusammenfassung

Was bedeutet es, ein eigenes Konzept für ein Fluggerät zu entwerfen, zu konstruieren und letztendlich die Algorithmen für die Flugsteuerung zu implementieren? Dies untersuche ich in diesem Projekt anhand des RoflCopters. Der RoflCopter ist ein von mir entworfenes Kleinstfluggerät, das auf dem Prinzip eines Monocopters basiert. Das bedeutet, dass der Auftrieb über zwei Rotorblätter erfolgt, die am Gehäuse des Copters befestigt sind. Das gesamte System wird von zwei Motoren an einem Motorausleger in Rotation versetzt und der Auftrieb über den Anstellwinkel der Rotorblätter gesteuert. Um seine Lage bzw. Position im Raum zu ändern, besitzt der RoflCopter zwei Steuerklappen an den beiden Rotorblattspitzen. Doch dieser Aufbau allein reicht nicht aus, um ein flugfähiges Gerät zu erhalten. Dazu ist auch ein Algorithmus nötig, der Steuerbefehle einließt, die nötigen Steueraktionen berechnet und schließlich, abgestimmt auf die Hardware, ausführt.

Inhaltsverzeichnis

1	Einleitung	3
2	Projektablauf	5
3	Aufbau und Funktion	6
3.1	Technischer Aufbau	6
3.2	Flugrelevante Funktionen	8
3.3	Hardware und Komponenten	9
3.4	Kein Flug ohne Software	10
4	Algorithmik	11
4.1	Ansteuerung von Servos und ESCs	11
4.2	Rotationsbestimmung im Raum	12
4.3	Berechnungen zur Lageänderung	13
4.3.1	Abgleich von IST und SOLL Werten	13
4.3.2	Vektoren als Grundlage der Steuerung	13
4.3.3	Ermittlung von Zeiten zur Ansteuerung der Klappen	14
4.4	Start und Landesequenz	15
5	Fazit	16
6	Verzeichnisse	17
6.1	Quellenverzeichnis	17
6.2	Vielen Dank an:	17

Kapitel 1

Einleitung

Der Traum vom Fliegen beflügelt seit jeher die kühnsten Träume der Menschheit. Waghalsige Konstruktionen folgten auf halsbrecherische Ideen und unzählige Abstürze auf zahllose Flugversuche. Die ersten funktionierenden Fluggeräte, Flugzeuge wie Helikopter, ließen sich ausschließlich über Mechaniken steuern und fliegen, der Pilot bewegte Klappen und Ruder allein durch Muskelkraft. Undenkbar für heutige Fluggeräte - Autopiloten und Flugassistenzsysteme nehmen Piloten viel Arbeit ab, Sensoren überwachen jede kleinste Bewegung und Motoren fahren Klappen und Ruder aus und ein.

Microcontroller und Computer haben jedoch nicht nur in großen Verkehrsmaschinen und Helikoptern Einzug gehalten. Sie ermöglichten Kleinstfluggeräte mit Konzepten, die mit einfacher Mechanik nicht realisierbar wären. Multicopter dienen hierfür als besonderes Beispiel: Die kleinen Fluggeräte sind simpel aufgebaut, fliegen von sich aus jedoch nicht stabil in der Luft. Erst die Flug- und Lageregelung durch einen kleinen Computer, einen Microcontroller, erlaubt auch ungeübten Piloten das einfache Fliegen dieser Geräte. Der Pilot steuert nicht mehr selbst, er gibt lediglich seinen Steuerwunsch an den Microcontroller weiter. Dieser kann die Daten, die für die Steuerung relevant sind, viel schneller und genauer messen und so das Fluggerät, in Zusammenspiel mit der gegebenen Mechanik, letztendlich steuern.

Über die letzten Jahre sind die Leistungsdaten von Microcontrollern erheblich gestiegen und die Preise gefallen. Das lässt nicht nur die Anzahl der verfügbaren, günstigen Kleinstfluggeräte steigen, sondern ermöglicht es auch, innerhalb eines überschaubaren Preisrahmens, selbst ein Konzept für ein Kleinstfluggerät zu entwickeln und umzusetzen. Genau dies habe ich mit dem RoflCopter getan.

Der RoflCopter ist zuerst einmal ein **R**otations**f**lügler (daher der Name), er erzeugt den benötigten Auftrieb also durch rotierende Flügel bzw. Rotoren. Des weiteren ist er ein Fluggerät, dass von seiner Bauform her eine Art Monocopter ist. Ein Monocopter zeichnet sich dadurch aus, dass er nur ein Rotorblatt und keinen Drehmomentausgleich, wie beispielsweise ein Helikopter, hat. Das bedeutet, dass sich nicht nur der Rotor, sondern das gesamte Fluggerät in der Luft dreht. So ist es natürlich schwierig, in der Luft sowohl



Abbildung 1.1: Bladestar [1]



Abbildung 1.2: Samari UAV [2]

eine ruhige Fluglage, als auch eine Steuerung über alle Achsen zu erhalten.

Der RoflCopter basiert im Prinzip auf den Konzepten des “WowWee Bladestar” (Abb.: 1.1) und des “Lockheed Martin Samari UAV” (Abb.: 1.2). Diese Konzepte habe ich zusammengeführt, da sie sich gegenseitig in ihren Vor- und Nachteilen ergänzen. Der “WowWee Bladestar” soll sich zwar steuern lassen, der technische Aufbau des Produktes lässt dies allerdings nicht zu: die Rotorblätter sind auf einen Winkel fest angestellt und nur die Motoren lassen sich in ihrer Umdrehungsgeschwindigkeit steuern. So lässt sich die Lage des Copters natürlich nicht verändern, das Fluggerät kann seine Position im Raum lediglich auf der Z-Achse verändern, also steigen oder sinken.

Der “Lockheed Martin Samari UAV” hat, anders als der Bladestar, eine aktive Lageregelung. Hier ist auch der technische Aufbau ein anderer, es handelt sich um einen klassischen Monocopter bzw. Gyropter - die Entwickler haben sich von einem Ahornsamen inspirieren lassen. Doch durch den asymmetrischen Aufbau und die daraus entstandene Unwucht fliegt der Copter nicht ruhig.

Ich versuche nun die Vorteile beider Konzepte zu verbinden: Den symmetrischen Aufbau des “WowWee Bladestar” und die aktive Steuerung des “Lockheed Martin Samari UAV”. Mit diesem Projekt möchte ich untersuchen, wie aufwändig es ist, eine eigene, neuartige Flugsteuerung für ein selbst konzipiertes Kleinstfluggerät zu entwickeln und umzusetzen. An Monocoptern als elektrisch betriebene Kleinstfluggeräte wird relativ wenig geforscht, es gibt nur einige wenige Forschungsarbeiten von Universitäten wie der ETH Zürich [3], dem MIT [4], sowie der University of Maryland [5] und Firmen wie Lockheed Martin [6]. Auch das hat mich besonders an dem Bau eines solchen Fluggerätes gereizt, da ich nahezu alles selbst entwickeln muss. Eine weitere Herausforderung ist natürlich, die Hardware und die Software so zu konzipieren und aufeinander abzustimmen, dass sie gleichzeitig Steuerbefehle und Sensoren einlesen, die nötigen Berechnungen vornehmen und die jeweiligen Steueraktionen ausführen kann.

Kapitel 2

Projektablauf

Die Idee für den RoffCopter hatte ich bereits im September 2015. Ich habe die ersten Entwürfe gemacht und angefangen, Bauteile für die erste Version zu fertigen oder zu bestellen.

Im Oktober 2015 habe ich die ersten Rotorblätter hergestellt. Diese habe ich dann an ein Grundgerüst, bestehend aus dem Motorausleger mit Motoren, Motorreglern und Servos, angebracht. Gegen Ende Oktober habe ich mit der Steuerelektronik und der Software angefangen. Zu diesem Zeitpunkt hatte ich noch keine Steuerklappen vorgesehen, sondern wollte die Servos zur Steuerung benutzen. Es hat sich jedoch herausgestellt, dass diese zu träge sind. Deshalb habe ich nach einer anderen Lösung gesucht und bin so auf die linearen Magnetaktuatoren gekommen. Im November und Dezember 2015 habe ich mich mit der Lageerkennung beschäftigt. Ich habe auch einen Kompasssensor verbaut, musste jedoch feststellen, dass dieser Sensor zu langsam für eine genaue Messung während der Rotation war.

Im Januar 2016 habe ich die ersten Steuerklappen an den Rotorblättern installiert und die Schaltung zur Ansteuerung der Magneten entworfen. Diese Schaltung ist bis heute die selbe geblieben. Im März habe ich die grundlegenden mathematischen Algorithmen zu Stabilisierung und Lageänderung entworfen und mit dem Bau des Infrarotmoduls begonnen.

Den Infrarot Teil konnte ich im April fertigstellen und anfangen, alle einzelnen Module für die erste Version des Copters zu testen und zu verbessern. Ende April habe ich mit den Tests des gesamten Systems angefangen und konnte dabei Umdrehungsgeschwindigkeiten und Lagemessungen prüfen. Die erste Version des Roff-Copters war fertig und ich konnte im Mai meine Facharbeit über den Copter abgeben. Letztendlich hat sich jedoch herausgestellt, dass erste Version zwar genügend Auftrieb für einen Flug erzeugt, jedoch flugunfähig ist. Hauptsächlich war die Montierung der Rotorblätter zu instabil, außerdem waren die elektrischen Komponenten nicht geordnet aufgebaut und der gesamte technische Aufbau nicht aerodynamisch durchdacht.

Nach einer Pause im Sommer habe ich ab September eine zweite Version des Copters designt. Diesmal begann ich nicht mit dem Bau, sondern fing an, das System per 3D Modell zu entwerfen und wichtige Teile am Computer zu simulieren. Darüber hinaus habe ich geprüft, welche Komponenten in der ersten Version funktionierten und welche ich tauschen wollte. Unter anderem habe ich die Motoren ersetzt, außerdem habe ich einen Lagerblock designt und 3D gedruckt. An diesem sind alle Komponenten befestigt. Die Elektronik ist in der zweiten Version auch nicht mehr lose verkabelt, sondern zentral auf einer Platine angebracht. Mit dem Bau der zweiten Version beschäftige ich mich seit Oktober 2016 und freue mich, sie funktionsfähig bei der Projektpräsentation zu zeigen.

Kapitel 3

Aufbau und Funktion

3.1 Technischer Aufbau

Der RofiCopter ist im Wesentlichen aus drei flugrelevanten Elementen aufgebaut: Dem antreibenden Element, dem auftreibenden Element und dem lageverändernden Element. Das Zusammenspiel dieser drei Elemente ermöglicht den Flug des Copters.

Zwei bürstenlose Elektromotoren der Baugröße 1304 mit zwei 3030 3-Blatt Propellern stellen das antreibende Element (Abb. 3.1). Die Motoren sind in die jeweils entgegengesetzte Richtung zeigend an den Enden eines 32cm langen Motorauslegers, einem 6mm Carbon-Vierkantsab, befestigt. Dazu habe ich die entsprechenden Motorhalterungen selbst designt und 3D gedruckt (Abb. 3.1). Da die Motoren jeweils in die entgegengesetzte Richtung zueinander angebracht sind, können sie den Copter in die benötigte Rotation versetzen, ihn also antreiben. Die Drehfrequenz bleibt während des Fluges näherungsweise gleich.



Abbildung 3.1: Motor am Motorausleger montiert [11]

Welcher Teil des RofiCopters das auftreibende Element ist, zeigt sich schon an seinem Namen. Er ist ein Rotationsflügler, das auftreibende Element ist der rotierende Flügel bzw. Rotor. Dieser hat die Maße von $250mm * 55mm$ und ist im Winkel von 90° zum Motorausleger angebracht. Auf die aerodynamischen Eigenschaften werde ich später noch eingehen. Beide Rotorblätter sind jedoch nicht starr verbaut, sondern lassen sich über Modellbauservos kippen und in ihrem Anstellwinkel verändern. Hier zeigt sich auch, dass die Rotorblätter nicht nur aufgrund ihres Flügelprofils das auftreibende Element des Copters sind, sondern auch, weil die eigentliche Auftriebssteuerung nicht über ein Verändern der Drehfrequenz geregelt wird, sondern bei konstanter Drehfrequenz über ein Verändern des Anstellwinkels beider Rotorblätter. Dieses Prinzip nennt sich kollektiver Pitch und wird in jedem Helikopter angewendet, da das Steuern der Steig- und Sinkrate über das Verändern der Drehfrequenz des Hauptrotors, also des permanenten Beschleunigungs und Bremsens des Rotors, einerseits zu träge und andererseits zu energieaufwändig wäre.

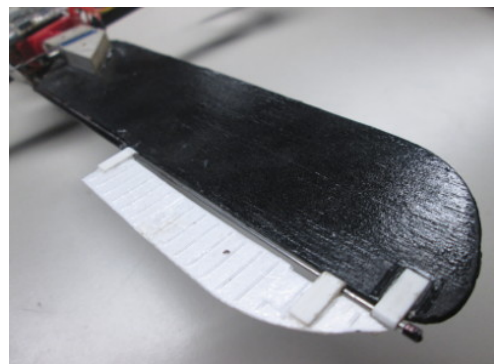


Abbildung 3.2: Steuerklappe am Rotorblatt; Prototyp V.1 [11]

Mit den beiden vorgestellten Elementen ließe sich der Copter zwar fliegen, eine Lage- bzw. Positionsänderung im Raum wäre jedoch nicht möglich. Dazu würde dem Copter die Möglichkeit fehlen, zu kippen und so in eine gewünschte Richtung zu fliegen. An dieser Stelle kommen die Klappen an den Rotorblattspitzen zum Einsatz (Abb. 3.1) - sie stellen das lageverändernde Element. Die Steuerklappen können über lineare Magnetaktuatoren (Abb. 3.6), kleine elektromagnetisch steuerbare Kolben, sehr schnell an- und abgestellt werden. Die lineare Bewegung wird über ein Ruderhorn und das daran angeschlossene Gestänge in eine Drehbewegung umgelenkt und auf die Steuerklappe übertragen. So kann einem Rotorblatt bzw. einer Seite des RoffCopters an einer bestimmten Stelle während der Umdrehung kurzzeitig mehr Auftrieb verschafft werden, als der anderen Seite. Dies bewirkt eine leichte Kippbewegung in die gewünschte Richtung. Auch dieses Prinzip ist im Helikopter umgesetzt; Es wird zyklischer Pitch genannt.

Eine weitere wichtige Voraussetzung für den Flug des RoffCopters ist ein Design, dass möglichst symmetrisch aufgebaut ist, um keine Unwucht zu erzeugen. Dazu gehört, möglichst alle schweren Bauteile, wie Elektronik, Servos oder den Akku zu zentralisieren und in der Mitte des Fluggerätes möglichst kompakt anzuordnen. Eine kompakte Anordnung in der Mitte ist auch insofern wünschenswert, als dass sie aerodynamisch gesehen die bremsende Angriffsfläche für die bei der Rotation entstehenden Luftströme minimiert. Zu diesem Zweck habe ich einen sechseckigen Lagerblock designt (Abb. 3.3), der im Zentrum des

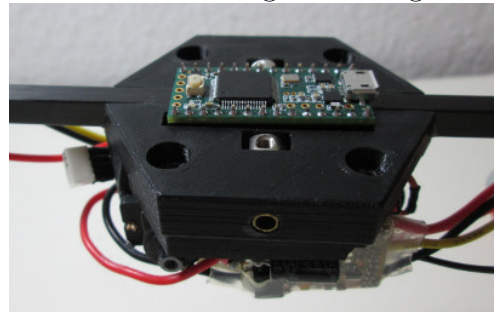


Abbildung 3.3: Lagerblock mit Teensy 3.2 [11]

beiden Rotorblätter und der Motorausleger befestigt; hier wird das antreibende Element mit dem auftreibenden Verbunden. Das bedeutet jedoch auch, dass in diesem Lagerblock alle Belastungen, die auf diese Bauteile wirken, zusammenlaufen und absorbiert werden müssen. Gerade die Rotorblätter müssen hier so befestigt sein, dass alle entstehenden Kräfte zwar gehalten werden können, sich die Rotorblätter aber dennoch kippen lassen. Ermöglicht wird dies durch die Messing-Gleitlager im Lagerblock und die Stellringe, mit denen die Rotorblätter fixiert werden (siehe Abb. 3.3). Da darüber hinaus alle Bauteile des Copters am Lagerblock befestigt sind, trägt er zwar die gesamte mechanische Belastung des Copters, muss aber trotzdem so leicht wie möglich sein. Deshalb habe ich auch dieses Bauteil 3D gedruckt.

Unter dem Lagerblock ist die Hauptplatine mit der Mess-, Steuer- und Funkelektronik angebracht. Die Schaltung auf der Platine und das Platinenlayout habe ich mithilfe der Software "KiCad" entworfen und die Platine in einem Ätzbild hergestellt. Da auf dieser Platine auch der Lagesensor sitzt, ist diese jedoch nicht direkt am Lagerblock angeschraubt. Sie ist vielmehr zwischen zwei Gummiringen eingeklemmt und damit schwingungsgedämpft aufgehängt. Außerdem ist der Spalt, der zwangsweise zwischen Platine und Lagerblock entsteht, durch eine Abschirmung vor den Luftströmen geschützt, die beim Rotieren entstehen. Beides dient dazu, um die Vibrationen, die während des Fluges entstehen und auf den Lagerblock übertragen werden, vor dem Lagesensor abzuschirmen und eine präzisere Lagemessung zu ermöglichen. Um den Aufbau noch kompakter halten zu können, habe ich den Microcontroller nicht direkt auf der Platine angebracht, dieser befindet sich oben auf dem Lagerblock (Abb. 3.3) und ist über Pinheader mit der Platine und den anderen Modulen verbunden.

Unter der Platine befinden sich die zwei Modellbauservos, die den Anstellwinkel der Rotorblätter verändern, sowie die Motortreiber für die Motoren und der Akku.

3.2 Flugrelevante Funktionen

Für den Flug des RoffCopters spielen die Funktionen, die Eigenschaften und die Auslegung der Komponenten eine wichtige Rolle. Alles muss so leicht wie möglich, so stabil wie nötig und so schnell wie gewünscht funktionieren, um im Zusammenspiel ein flugfähiges Gerät zu bilden. Aus diesem Grund ist es unumgänglich, Modell und Bauteile vor der Konstruktion am Computer als 3D Modell anzufertigen und über Simulationen zu testen.

Zunächst werde ich auf die aerodynamischen Eigenschaften der Rotorblätter eingehen und darauf auf weitere Mechanik zu sprechen kommen. Die Rotorblätter habe ich mit Hilfe des Programms "XFLR5" in ihrem Profil designt, simuliert und letztendlich als 3D Modell angefertigt. Über die Simulation konnte ich zum Beispiel feststellen, wie groß der Auftrieb des Rotors bei einem bestimmten Anstellwinkel ist und dass bei einem Anstellwinkel von 10° der Quotient zwischen Auftrieb und Luftwiderstand maximal ist; der Rotor demzufolge bei diesem Anstellwinkel am effizientesten arbeitet. Die Simulationsergebnisse des letztendlich entstandenen Rotorprofils sind in Abbildung 3.1 dargestellt. Ein weiterer wichtiger Anspruch an das Profildesign war, dass es sich mit einer CNC Fräse aus Balsaholz fräsen lässt.



Abbildung 3.4: Auftriebsverteilung am verwendeten Rotorprofil [12]

Wie bereits im vorherigen Abschnitt "Technischer Aufbau" beschrieben, können die Rotorblätter mithilfe von Modellbauservos in ihrem Anstellwinkel verändert werden und bilden so eine Auftriebssteuerung nach dem Prinzip des kollektiven Pitch. Der Aufbau der Mechanik ist in Abbildung 3.2 dargestellt. Um das Rotorblatt anzustellen, dreht sich der Servo (Abb. 3.2, rot eingefärbt) im Uhrzeigersinn. Über den Servoarm wird diese Bewegung auf das Gestänge übertragen, das den Servoarm und das Ruderhorn am Rotorblatt verbindet (Abb. 3.2, ebenfalls rot eingefärbt). Da das Ruderhorn (Abb. 3.2, grün eingefärbt) parallel zur Lageraufhängung angebracht ist, dreht sich das Rotorblatt ebenfalls im Uhrzeigersinn und wird so um einen gewünschten Winkel angestellt. Beide Servos werden vom Microcontroller über die selbe Signalleitung angesteuert und können so auf jeden beliebigen Winkel von 0° bis 180° eingestellt werden. Dabei haben sie ein Drehmoment von ca. $20N_{cm}$ [7]. Allerdings sind die Servos recht langsam, für eine Positionsänderung von 60° benötigen sie ca. 110ms. Für den RoffCopter werden so große Stellwinkel zwar nie benötigt, jedoch dauert es aufgrund des Übertragungsprotokolls für Modellbauservos (Details dazu im nächsten Abschnitt) mindestens 20ms, bis ein Signal eingelesen wurde. Erst darauf beginnt der Servo mit der Bewegung. In dieser Zeit hat sich der RoffCopter, bei einer angenommenen Drehfrequenz von ca. 13Hz, also ca. 800 U/min., schon $\frac{1sec=1000ms}{13Hz=13U/1000ms} = 76,92ms = \text{Dauer einer Umdrehung} \rightarrow \frac{76,92ms}{360^\circ} = 0,214ms = \text{Dauer einer Drehung von } 1^\circ \rightarrow \frac{20ms}{0,214ms/1^\circ} = 93,46^\circ$ weitergedreht. Daraus folgt, dass die Servos viel zu träge sind, um kurzzeitig ein Rotorblatt anders anzustellen um eine Lageänderung zu erzeugen und so den Copter zu steuern.

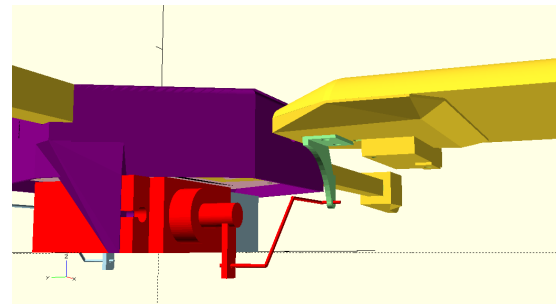


Abbildung 3.5: Servo und Gestänge zum Anstellen des Rotorblattes [13]

Zu diesem Zweck verwende ich lineare Magnetaktuatoren, auch solenoid actuator oder einfach solenoid genannt, die eine Steuerklappe an der Rotorblattspitze anstellen. Im Prinzip handelt es sich dabei um einen Elektromagneten, der einen Kolben anzieht und so eine lineare Bewegung ausführt. Der Elektromagnet kann allerdings sehr schnell geschaltet werden, ich habe in praktischen Tests herausgefunden, dass die Steuerklappe innerhalb von ca. 2ms - 3ms angestellt werden kann, wenn der solenoid auf die Akkuspannung von ca. 8V geschaltet wird. So kann das Funktionsprinzip des sogenannten zyklischen Pitch angewendet werden. Das bedeutet für den RoffCopter, dass durch Anstellen einer Steuerklappe innerhalb eines Teiles einer Umdrehung ein Rotorblatt mehr Auftrieb bekommt, als das andere. Durch diese Auftriebsdifferenz kippt der Copter leicht in die gewünschte Richtung und kann gesteuert werden. Die Solenoids sind über ein Gestänge mit den Steuerklappen verbunden; der Aufbau ist in Abbildung 3.3 dargestellt. Eine weitere flugrelevante Funktion stellt auch die Anordnung der Komponenten unter der Höhe der Rotorblätter dar. Dadurch, dass Servos, Motorregler und Akku als schwerste Komponenten unter dem Lagerblock sitzen und möglichst kompakt angeordnet sind, sitzt der Schwerpunkt des Copters sowohl in der Mitte des Fluggerätes, jedoch auch unter der Höhe der Rotorachse. So kann der Copter unwuchtsfrei drehen und wird durch die tief-lastige Anordnung der Komponenten automatisch in eine waagerechte Fluglage gezogen. Diese passive Stabilisierung allein reicht für einen stabilen waagerechten Flug nicht aus, da dafür das Trägheitsmoment verglichen mit der Bewegung in den drei Raumachsen zu groß ist. Die aktive Lagerregelung wird daher durch die Steuerklappen übernommen, die Stabilisierung in der waagerechten Fluglage jedoch erheblich unterstützt.

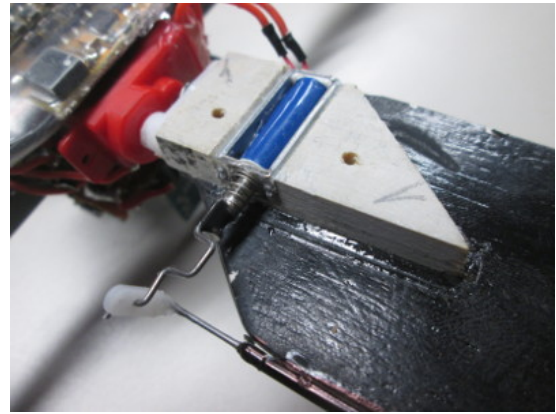


Abbildung 3.6: Solenoids und Gestänge für die Steuerklappen; Prototyp V.1 [11]

3.3 Hardware und Komponenten

In diesem Abschnitt möchte ich darauf eingehen, welche elektrotechnischen Komponenten für die Steuerung des RoffCopters verbaut wurden. Das "Herzstück" des Copters ist der Microcontroller. Dieser kleine Computer nimmt alle Steuerbefehle des Piloten entgegen, wertet die Sensoren aus, berechnet die nötigen Steueraktionen und führt diese aus. Als Chip kommt ein ARM Cortex M4 Microcontroller auf einem Teensy 3.2 Board zum Einsatz. Dieser Prozessor hat nicht nur die entsprechend benötigten Leistungsdaten (144MHz Taktgeschwindigkeit...), sondern auch mehrere Hardwaretimer. Diese Hardwaretimer ermöglichen, dass der Microcontroller die Steuerbefehle des Piloten entgegennehmen kann, während gleichzeitig und zum richtigen Zeitpunkt die Steuerklappen mit Signalen versorgt werden können. Anders gesagt ermöglichen die Hardwaretimer eine Form der Parallelisierung für die zu bewältigenden Aufgaben, ohne dass die Software durchs Warten auf Ereignisse gelähmt und das Ausführen anderer Aktionen behindert wird. Der ARM Cortex M4 Prozessor auf dem Teensy 3.2 kombiniert alle gewünschten Eigenschaften für den Copter: Er ist Klein, leicht und hat ausreichend Rechenleistung.

Der Grundstein, auf dem die Lageregelung basiert, ist der Lagesensor. Hier kommt ein MPU9250 von Ivensense zum Einsatz. Dieser Sensor kombiniert einen 3-Drehratensensor, einen 3-Achs Beschleunigungssensor und einen 3-Achs Kompass. Mithilfe dieser Sensoren kann der RoffCopter seine Lage im Raum bestimmen und entsprechende Steueraktionen

ausführen, um einen stabilen Flug zu gewährleisten.

Um die Steuerbefehle des Piloten entgegennahmen zu können, benötigt der RoflCopter einen passenden Empfänger für die Fernbedienung. Als Fernsteuersystem habe ich mich für das Taranis System von FrSky entschieden, weil es einerseits das sehr schnelle Übertragungsprotokoll ACCST nutzt, das Steuerbefehle mit einer Latenz von nur ca. 7ms überträgt. Ein weiterer Vorteil ist die große Verbreitung und damit auch die Verfügbarkeit sehr kleiner Empfängerplatinen, da diese in sehr vielen Wettkampquadrocoptern verbaut werden. Da es sich bei den Motoren des RoflCopters um bürstenlose Motoren, sogenannte BLDCs handelt, können diese nicht einfach über das Anlegen einer Gleichspannung in Drehung versetzt werden. Bürstenlose Motoren, im Fall des RoflCopters bürstenlose Außenläufer (Abb. 3.1), haben drei voneinander getrennte Motorphasen, jeweils alternierend zueinander auf dem Stator angebracht sind. Diese werden entsprechend der Ausrichtung des Rotors nacheinander angesteuert. So ergibt sich ein annähernd kreisförmiges Drehfeld zwischen Rotor und Stator, das den Rotor in Drehung versetzt. Die Motorregler, die ich im RoflCopter verbaut habe, stammen ebenfalls auf dem Modellbaubereich und übernehmen genau diese Aufgabe. Angesteuert werden sie genau so, wie die Modellbauservos, die den Anstellwinkel der Rotorblätter steuern. Mehr dazu in Abschnitt 4.1

Um seine Ausrichtung im Raum relativ zum Piloten bestimmen zu können, ist im RoflCopter ein Infrarot-Empfänger Typ TSOP38238 verbaut. Dieser hat ebenfalls nur eine Signalleitung, die entweder an oder aus ist, je nachdem, ob ein Signal empfangen wurde oder nicht. Um nicht von anderen Infrarot aussendenden Lichtquellen gestört zu werden, besonders der Sonne, nimmt der Sensor nur Signale an, die auf einer Wellenlänge von 940nm mit 38kHz gepulst werden. Wie genau der Microcontroller das Signal des IR-Empfängers auswertet ist in Abschnitt 4.2 beschrieben.

Neben der Kommunikation mit dem Piloten in Form des Einlesens der Steuerbefehle muss der Copter ebenfalls mit einem Computer kommunizieren können, um beispielsweise Debug Informationen bzw. Telemetriedaten, wie den Ladestand des Akkus, zu übermitteln. Dies gewährleistet den sicheren Betrieb und vermeidet Unfälle sowie unkontrollierte Abstürze. Zu diesem Zweck habe ich ein kleines Bluetooth Modul eingebaut. Dieses sendet einfache Zeichenketten, die über eine serielle RX/TX Schnittstelle eingegeben werden.

3.4 Kein Flug ohne Software

Auf Grund der hohen Drehgeschwindigkeiten und der dadurch erforderlichen, kurzen Reaktionszeiten ist eine lageregulierende Software für einen stabilen Flug unabdingbar. Dabei wird die Flugweise vor allem durch zwei Faktoren erschwert. Einerseits fliegt der Copter nicht eigenstabil, das bedeutet, dass er nicht ohne eine aktive Steuerung geradeaus fliegt. Man kann das gut mit dem Balancieren eines Stiftes auf der Hand vergleichen. Es müssen andauernd kleine Korrekturen der Lage vorgenommen werden, um den Copter nicht aus der gewünschten Flugbahn ausbrechen zu lassen. Andererseits müssen alle Berechnungen und Steueraktionen bei einer Umdrehungszeit von 50ms - 100ms schnell erfolgen. Der menschliche Pilot ist für die notwendigen Korrekturen zu reaktionsträge, wobei diese Korrekturen teilweise auch noch gleichzeitig erfolgen müssen. Somit ist die Verwendung einer geeigneten Software, die die Steuerbefehle an die Regelstrecke weitergibt, die letztendlich die Steueraktionen bestimmt, notwendig. Zusätzlich zur Latenzzeit kommt der Umstand, dass der Pilot sich nicht im Fluggerät befindet und somit nötige Lageausrichtungen nicht erkennen kann. Auch dies muss also in der Boardsoftware automatisiert hinterlegt sein.

Kapitel 4

Algorithmik

Im folgenden Teil der Arbeit werde ich nur auf die verwendeten Prinzipien der Software und Algorithmik eingehen. Programmcode werde ich nicht einbinden, da sich diese noch laufend ändern; sie werden aber alle in meinen github repository veröffentlicht: <https://github.com/fr3dn3t/RoflCopter>

4.1 Ansteuerung von Servos und ESCs

Wie schon mehrfach erwähnt, stammen die Servos und Motorregler aus dem Modellbaubereich. Da es dort üblich ist, dass diese Komponenten direkt an den Modellbauempfänger oder einen Flugregler angeschlossen werden und sofort funktionieren (Plug and Play), beschränkt sich die einseitige Kommunikation zwischen Bauteilen einerseits auf das Wesentliche (die Übertragung der Steuerinformationen von Empfänger zum Bauteil) andererseits wird ein sehr einfaches Protokoll bzw. eine sehr einfache Modulation für die Datenübertragung genutzt, die sogenannte Pulsweitenmodulation, kurz PWM. Diese benötigt nur eine Signalleitung, die zwei Zustände annehmen kann: "An" oder "Aus". "An" bedeutet in diesem Fall, dass die Spannung auf der Signalleitung gleich der Versorgungsspannung des Bauteils ist (in der Modellbautechnik üblicherweise 5V) und "Aus", dass die Signalleitung aus Masse geschaltet ist. Pulsweitenmodulation bedeutet nun, dass es innerhalb eines festgelegten Zeitraumes, einer Periode, einen Puls gibt, dessen Länge variieren kann, um einen bestimmten Wert zu übertragen. Im Modellbaubereich beträgt die Länge des festgelegten Zeitraumes, die Periodendauer, 20ms. Innerhalb dieser 20ms gibt es nun einen Puls, dessen Länge zwischen 1ms und 2ms variieren kann (siehe Abb. 4.1), wobei 1ms das mögliche Minimum und 2ms das Mögliche Maximum der Aussteuerung darstellt. Minimum bedeutet bei einem Servo 0° und bei einem Motorregler keine Motordrehung, Maximum bedeutet bei einem Servo 180° und bei einem Motorregler die maximal mögliche Geschwindigkeit des Motors. Der Microcontroller muss nun diese Modulation anwenden um Servos und Motorregler anzusteuern. Normalerweise werden alle Servos und Motorregler mit einer eigenen Signalleitung angesteuert, im Falle des RoflCopters sind die Signalleitungen jedoch zusammengefasst. Dies ist möglich, da die Servos ausschließlich den kollektiven Pitch über das Verstellen des Anstellwinkels der Rotorblätter steuern und deshalb sowieso immer auf den selben Winkel eingestellt werden. Auch die Motoren sollen beide immer gleich schnell drehen, nutzen auch sie die selbe Signalleitung.

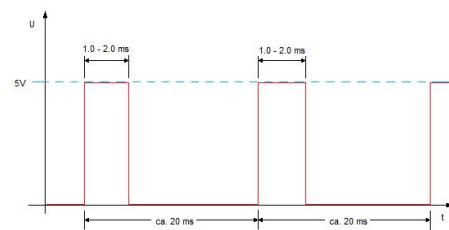


Abbildung 4.1: Servo Impulsdiagramm [8]

4.2 Rotationsbestimmung im Raum

Für die Steuerung des ROFL-Copters ist es unerlässlich, dass seine Ausrichtung zum Piloten bekannt ist. Gibt dieser den Steuerbefehl "nach vorne fliegen", so muss sich der Copter vom Piloten wegbewegen. Außerdem muss die Umdrehungszeit und -geschwindigkeit bekannt sein, um mit diesen Werten die aktuelle Lageabweichung und letztendlich die Ansteuerzeiten für die Klappen berechnen zu können.

Ich habe diese Probleme mit einem Infrarotempfänger gelöst, der nach einem Infrarotsender sucht. Der andauernd sendende Sender, eine Infrarotbarke, ist in Form eines JR-Moduls unter der Fernsteuerung angebracht (Abb. 4.3). Außerdem wird das Licht der 5W Infrarot LED über eine Sammellinse mit einem Öffnungswinkel von 10° ausgesendet. So ist gewährleistet, dass die Barke immer nur in die Richtung sendet, in die der Pilot schaut. Der IR Empfänger befindet sich im Lagerblock des RoFLCopters. So ist sein Öffnungswinkel über eine 0,6mm Blende ebenfalls eingeschränkt. Die Werte für die Sammellinse und die Blende habe ich experimentell ermittelt.

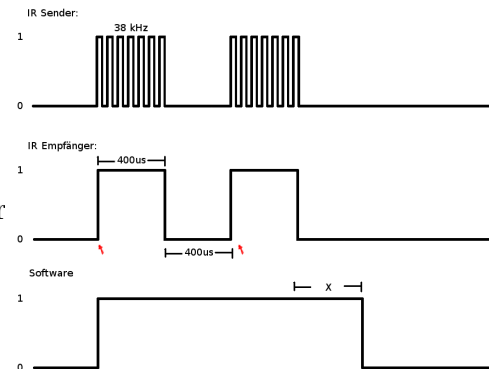


Abbildung 4.2: Infrarotsignal Impulsdia-
gramm [14]

Wie bereits in Abschnitt 3.3 beschrieben ist nimmt der Empfänger generell nur Infrarotsignale an, die mit 38kHz gepulst wurden. Wenn er jetzt also ein solches Infrarotsignal empfängt, gibt der Empfänger an den Microcontroller eine logische 1 zurück, ansonsten eine logische 0.

Da der Empfänger, der für Datenübertragung über IR konzipiert ist, keine anhaltenden Signale bzw. Zustände weitergeben kann, sendet der Sender nicht einfach den Dauerzustand 1, sondern wechselt alle 400us zwischen 0 und 1.

Die Software muss die eingehenden Signale nun so interpretieren, dass der andauernde Wechsel zwischen 0 und 1 innerhalb von 400us einer logischen 1 und das Anliegen einer 0 für mindestens 800us einer logischen 0 entspricht (siehe Abb. 4.2). Zusätzlich kann eine Pufferzeit hinzugefügt werden (x in Abb. 4.2), die verhindert, dass zu schnell angenommen wird, dass sich der Copter bereits aus dem Infrarotstrahl des Senders herausgedreht hat und so bei einer erneuten Verbindung angenommen würde, dass sich der Copter in der Zwischenzeit einmal gedreht hätte. Jedes Mal, wenn das Eingangssignal von 0 auf 1 wechselt (rote Pfeile in Abb. 4.2), wird ein Interrupt innerhalb des Microcontrollers aufgerufen, der den Regelkreis für die Lageregelung anstößt. Dies passiert, da der Microcontroller nun feststellen kann, dass der Copter mit dem IR Empfänger in Richtung des Piloten ausgerichtet ist und so seine Steuerbefehle passen interpretieren kann. Konkret bedeutet dies, dass wenn der Pilot den Befehl "Nach vorne fliegen" gibt, der Copter weiß, dass er bei seiner aktuellen Umdrehungsgeschwindigkeit noch x Millisekunden warten muss, bis er eine Steuerklappe anstellt, um die gewünschte Lageänderung vorzunehmen.

Neben der Bestimmung der Ausrichtung des Copter relativ zum Piloten kann über den IR-Empfänger ebenfalls die aktuelle Umdrehungsgeschwindigkeit sehr genau gemessen werden. Dazu misst der Microcontroller einfach die Zeitdifferenz zwischen



Abbildung 4.3: Infrarotsender mit Sam-
mellinse an der Fernsteuerung [11]

den Pulsen, die vom IR Empfänger weitergegeben werden. Beträgt die Zeit zwischen den Pulsen beispielsweise 100ms kann errechnet werden, dass der RoFLCopter mit $\frac{1}{60} \text{min} = 60 \text{sec} / 100 \text{ms} = 0.1 \text{sec} / U = 600 U / \text{min}$ dreht.

4.3 Berrechnungen zur Lageänderung

4.3.1 Abgleich von IST und SOLL Werten

Durch das Einlesen des Beschleunigungs- und Drehratensensors und anschließende Fusionieren der einzelnen Sensorwerte mithilfe des Kalman Filters ist bekannt, wie der Copter im Raum liegt. Da jedoch nur an einer Stelle jeder Umdrehung bekannt ist, wie der Copter ausgerichtet ist, nämlich wenn der IR Empfänger eine Sichtverbindung zu dem IR Sender hat, kann auch nur an dieser Stelle die Lage des Copters mit dem Lagesensor eingelesen werden, da sich das feste Koordinatensystem innerhalb des Lagesensors während der Umdrehung des Copters ebenfalls mitdreht. Wird an dieser Stelle gemessen und die Regelschleife angestoßen, gibt es einen aktuellen IST Wert. Der Pilot des ROFL-Copters sendet über die Fernsteuerung die Steuerbefehle. Sie definieren den SOLL Wert. Die Aufgaben der Flugsteuerung ist nun, einen Weg von IST zu SOLL zu finden und den Copter dementsprechend zu steuern.

Im Fall des ROFL-Copters beinhaltet sowohl der IST, als auch der SOLL Wert die Information über die Drehung bzw. Rotation auf der X als auch auf der Y Achse. Die Werte kann man als Punkte in ein Koordinatensystem eintragen. Einen IST-Punkt und einen SOLL-Punkt mit den dazugehörigen X|Y Koordinaten. Im Beispiel sind die Werte IST = (2 | 1) & SOLL = (-1 | 2) (Abb.: 4.4).

Jetzt zieht man die SOLL Werte von den IST Werten ab, man verschiebt die Punkte auf dem Koordinatensystem also so, dass der SOLL-Punkt im Ursprung liegt.

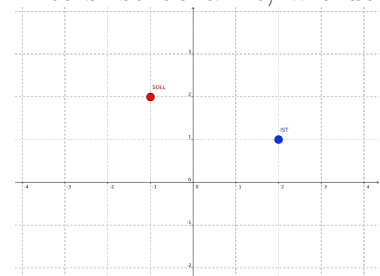


Abbildung 4.4: IST und SOLL Punkt [15]

4.3.2 Vektoren als Grundlage der Steuerung

Der Algorithmus zu Lageänderung basiert auf einem fundamentalen mathematischen Prinzip: Vektoren. Aber warum muss der SOLL Punkt in den Ursprung verschoben werden? Wenn man den SOLL-Punkt als Kreismittelpunkt ansieht, der Kreis außen durch den IST-Punkt verläuft und man diesen Kreis "Rotorkreisfläche" nennt, wird alles schon etwas klarer. Die Rotorkreisfläche ist der Bereich, in dem sich der Rotor des ROFL-Copters dreht.

Zeichnet man nun einen Vektor vom IST-Punkt zum SOLL-Punkt, erkennt man bereits einen Teil der Steuerung. Der IST-Punkt ist der Punkt, an dem eine Steuerklappe angestellt sein muss, damit der Copter in Richtung des Vektors kippt bzw. fliegt.

Um Herauszufinden, wo genau die Klappe innerhalb der Rotorkreisfläche angestellt werden muss, braucht man den Winkel zwischen der aktuellen Ausrichtung des Copters (mit dem IR Empfänger in Richtung des Piloten) und dem Richtungsvektor. Die aktuelle Ausrichtung des Copters ist ein weiterer Vektor. Er liegt auf der X-Achse, genauso wie die Rotorblätter, wenn der IR Empfänger eine Sichtverbindung zum IR Sender hat.

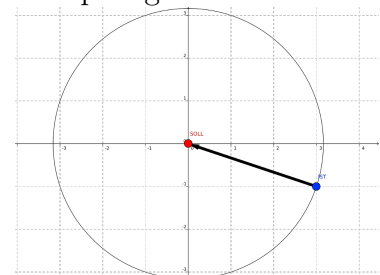


Abbildung 4.5: Rotorkreisfläche und Richtungsvektor [15]

Bevor man den Winkel zwischen den Beiden Vektoren errechnen kann, muss man feststellen, in welchem Quadranten des Koordinatensystems der IST-Punkt liegt.

So kann man beurteilen, ob es sich um einen stumpfen oder spitzen Winkel in Bezug auf die nächste Steuerklappe handelt und dementsprechend rechnen. In Abb. 4.6 liegt der Punkt im 4. Quadranten und damit in einem spitzen Winkel zum rechten Rotorblatt. Jetzt kann man den Betrag des Skalarproduktes beider Vektoren durch die Länge des Richtungsvektors teilen. Das Ergebnis setzt man in den \cos^{-1} ein und bekommt als Ergebnis den Winkel α von der aktuellen Ausrichtung bis zum Richtungsvektor.

Als nächstes berechnet man den Winkel β , in dem die Klappe angestellt sein soll. Dieser Winkel definiert, wie stark der Copter kippt (größerer Anstellwinkel \rightarrow längere Anstellzeit der Klappe \rightarrow mehr Auftrieb für das jew. Rotorblatt \rightarrow stärkeres Kippen).

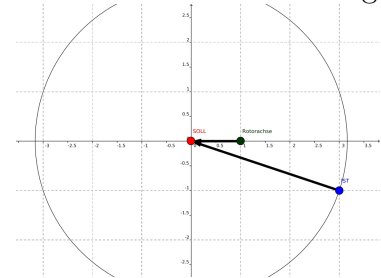


Abbildung 4.6: Vektor der Rotorachse [15]

Da der Copter um so stärker aussteuern soll, wenn die Differenz zwischen IST und SOLL Wert größer ist, kann man den Winkel einfach proportional zu dieser Differenz berechnen. Die Differenz und damit der Winkel β ist die Länge des Richtungsvektors multipliziert mit x . x ist der Proportionalitätsfaktor und damit eine Einstellbare Größe der Regelung.

Außerdem braucht man noch den Winkel, der zwischen der aktuellen Position und dem Anstellen der Klappe liegt. Dazu zieht man die Hälfte des Anstellwinkels β von dem Winkel zum Richtungsvektor α ab. Man erhält den Winkel γ . Letztendlich braucht man nur den Winkel γ zwischen der aktuellen Position und dem Anstellen der Klappen und den Anstellwinkel der Klappe β .

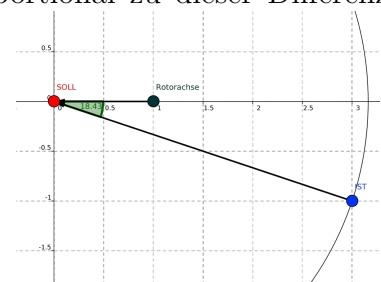


Abbildung 4.7: Schnittwinkel α [15]

4.3.3 Ermittlung von Zeiten zur Ansteuerung der Klappen

Mit den errechneten Winkeln β & γ aus Abschnitt 4.3.2 und der mithilfe des IR Empfängers ermittelten Umdrehungsgeschwindigkeit kann man jetzt sehr schnell die Ansteuer- und Wartezeiten für die Klappensteuerung herausfinden. Die erste Zeit, die für die Steuerung benötigt wird, ist die Zeit, die der Microcontroller warten muss, bis er eine Steuerklappe anstellen soll. Diese Zeit setzt sich aus der letzten Umdrehungszeit geteilt durch den Anteil des Winkels γ am Vollkreis, zusammen. Ein Beispiel: Die letzte Umdrehung hat $40\text{ms} = 40000\text{us}$ gedauert. Der errechnete Winkel γ beträgt 60° . Dann beträgt die Zeit, die der Microcontroller warten muss, bis er die Steuerklappe anstellt $\frac{40000}{360/60} \approx 6666\text{us}$.

Genauso kann man mit dem Anstellwinkel β verfahren, um zu berechnen, wie lange die jeweilige Steuerklappe angestellt bleiben soll. An dieser Stelle werden die Hardwaretimer im Microcontroller eingestellt und so die Steuerklappen präzise und im richtigen Moment angesteuert, um den gewünschten Kippeffekt zu erhalten. Nachdem die eine Steuerklappe wieder abgestellt wurde, werden die Hardwaretimer so gesetzt, dass die Steuerklappe am anderen Rotorblatt die selbe Bewegung zur passenden Zeit ausführt und den Copter so ein kleines Stück weiter kippt. Nachdem dies ausgeführt wurde, wartet der Microcontroller auf die Sichtverbindung des IR Empfängers zum IR Sender und die Regelschleife beginnt von vorne.

4.4 Start und Landesequenz

Die beschriebene Regelschleife läuft während des gesamten Fluges - außer an zwei Punkten: Start und Landung. An diesen Stellen kann die Lageregelung nicht funktionieren, da die Berechnung der Warte- und Ansteuerzeiten darauf basiert, dass die folgende Umdrehung ungefähr so lange dauern wird, wie die vorherige. Da während des Starts sich die Umdrehungsgeschwindigkeit jedoch andauernd ändert und jede folgende Umdrehung schneller ist, als die vorherige, funktioniert das Prinzip der Lageregelung hier nicht. Auch bei der Landung ändert sich die Umdrehungsgeschwindigkeit andauernd; der RoflCopter dreht zunehmend langsamer. Aus diesem Grund müssen für Start und Landung andere Abläufe implementiert werden, als für den normalen Flug.

Ein Vorteil für die Konzipierung dieser Abläufe ist jedoch, dass sowohl die Start, als auch die Landesequenz im Vergleich mit dem restlichen Flug nur sehr kurz ist - sie dauert nur einige Sekunden. An dieser Stelle ist die Tatsache wichtig, dass der RoflCopter über eine, wenn auch recht träge, passive und dauerhaft aktive Lagestabilisierung verfügt: Der tieflastige Aufbau der Komponenten wie Elektronik, Servos und Akku unter dem Lagerblock. Das bereits in Abschnitt 3.2 beschriebene Prinzip der passiven Stabilisierung, die die Rotorachse in eine waagerechte Position zieht, hilft bei Start und Landung erheblich, weil es eine aufwändige aktive Lageregelung an diesen Punkten ersetzt. Kurz: Während des Starts und der Landung ist die aktive Lageregelung des RoflCopters abgeschaltet.

Startsequenz: Die Grundbedingung für einen Start ist, dass alle Steuerknüppel der Fernsteuerung in Mittelstellung sind. Dies prüft der Microcontroller direkt zu Anfang der Startsequenz. Anschließend werden die Motoren langsam angedreht, sie laufen etwas mit 1/10 ihrer Leistung. Der Copter wird nun vom Piloten mit der Hand angedreht und losgelassen. Die Bewegung des Andrehens wird vom Microcontroller über den Lagesensor als Spitze in den Werten der Z-Achse des Drehratensensors festgestellt. Dieser weiß also nun, dass der RoflCopter vom Piloten losgelassen wurde und frei in der Luft steht. Daraufhin wird die Motorleistung und der Anstellwinkel der Rotorblätter maximiert, um den Copter schnellstmöglich in eine konstante Rotation zu versetzen, die für den Auftrieb nötig ist. Bei der Zielumdrehungsgeschwindigkeit von 600U/min - 800 U/min werden die Motoren auf ca. 70% ihrer Leistung heruntergeregelt, der Anstellwinkel der Rotorblätter auf ca. 5° - 8° reduziert und die Regelschleife aktiviert. Der RoflCopter befindet sich nun im finalen steuerbaren Zustand.

Landesequenz: Zuerst steuert der Pilot den RoflCopter auf eine Höhe von ca. 1m über dem Boden, darauf aktiviert der die Landesequenz mit einem Schalter an der Fernbedienung. Der Microcontroller deaktiviert die Regelschleife, bremst die Motoren auf ca. 20% ihrer Leistung und stellt die Rotorblätter auf einen Winkel von ca. 5° an. Der RoflCopter sinkt nun langsam. Das Aufsetzen auf dem Boden wird vom Microcontroller wieder als Spitze in den Werten des Lagesensors festgestellt. Daraufhin werden die Motoren ausgestellt und die Rotorblätter auf 0° angestellt. Der RoflCopter ist gelandet.

Kapitel 5

Fazit

Was lässt sich nun rückblickend über dieses Projekt sagen? Einerseits gab es Abschnitte bzw. Teile des Projektes deren Umsetzung ich anfangs schwieriger eingeschätzt habe, als sie waren. Andererseits gab es auch Abschnitte, die deutlich komplizierter waren, als ich ursprünglich gedacht habe.

Erstaunt hat mich, wie einfach der Algorithmus zur Berechnung der Steuerzeiten für die Klappen letztendlich war. Diesen Algorithmus jedoch so in das Programm einzubetten, dass er schnell genug lief, war eine Herausforderung. Da neben diesem Kernalgorithmus noch die Steuerungen für die Servos und die Motoren laufen und andauernd die Steuerbefehle der Fernsteuerung eingelesen werden mussten, ist es wichtig, dass die Flugsteuerung Modular, also in vielen kleinen, einzeln aufrufbaren Programmfunktionen, implementiert ist. Auch diesen Aspekt habe ich zunächst etwas unterschätzt, konnte ihn letztendlich aber gut umsetzen.

Gerade die Verfügbarkeit von open source Software wie openScad [9] oder XFLR5 [10] hat dieses Projekt ermöglicht. Auch die heutzutage nicht mehr selten anzutreffenden Techniken, wie 3D Druck oder CNC Fräsen, die eine günstige Fertigung der Bauteile möglich machen, unterstützen den Forschungsprozess erheblich. Darüber hinaus gibt es gut dokumentierte, leistungsstarke Microcontroller, deren Finanzierung niemanden vor eine unüberwindbare Hürde stellen (das im RoflCopter verwendete Teensy 3.2 Board kostet ca. 20€). So können auch Schüler, wie ich, eigene Projektkonzepte in dieser Größe entwerfen und innerhalb eines überschaubaren Preisrahmens (für den RoflCopter Version 2 ca. 200€) verwirklichen.

Einen Rotationsflügler habe ich anfangs für dieses Projekt gewählt, da dieser Bereich ziemlich unerforscht ist. Während des Projektes ist mir klar geworden, warum das so ist: Die Idee, mit einem Propeller einen Rotor anzutreiben, klingt zwar einleuchtend, ist aber alles andere als effizient. Schon ein einzelner Motor des ROFL-Copters könnte diesen, bei einer Zugkraft von ca. 3N, problemlos hochheben und in der Luft halten. Da mir das Entwerfen und Bauen des Modells aber viel Spaß gemacht hat, ist mir das nicht weiter wichtig.

Zusammenfassend möchte ich festhalten, dass die Entwicklung einer eigenen, neuartigen Flugsteuerung zwar sehr aufwändig, aber nicht unbedingt kompliziert ist. Es müssen viele verschiedene Aspekte berücksichtigt und die Software genau auf die Hardware abgestimmt werden, doch letztendlich ist auch die Flugsteuerung nur der Übersetzer zwischen Pilot und Fluggerät.

Kapitel 6

Verzeichnisse

6.1 Quellenverzeichnis

- [1] Bildquelle: <http://www.engadget.com/2008/09/15/wowwees-flytech-bladestar-can-govern-your-home-autonomously/> (eingesehen am 07.05.2016)
- [2] Bild extrahiert aus: <https://www.youtube.com/watch?v=jtU1ZsjAqsE> (eingesehen am 07.05.2016)
- [3] <http://robohub.org/the-monospinner-worlds-mechanically-simplest-controllable-flying-machine/> (eingesehen am 06.05.2016)
- [4] <https://www.youtube.com/watch?v=ozp9e98BMhw> (eingesehen am 15.01.2017)
- [5] <https://www.youtube.com/watch?v=sbuGCgc-JCM> (eingesehen am 15.01.2017)
- [6] <https://www.youtube.com/watch?v=jtU1ZsjAqsE> (eingesehen am 07.05.2016)
- [7] https://hobbyking.com/de_de/turnigytm-tgy-180d-180-degree-digital-servo-2-2kg-0-10sec-12g.html (eingesehen am 15.07.2017)
- [8] <http://www.mikrocontroller.net/articles/Datei:Servo.gif> (eingesehen am 15.01.2017)
- [9] <http://www.openscad.org> (eingesehen am 15.01.2017)
- [10] <http://www.xflr5.com/xflr5.htm> (eingesehen am 15.01.2017)
- [11] Eigene Aufnahme
- [12] Bildschirmfoto der Software XFLR5
- [13] Ausschnitt aus dem 3D Modell des RoflCopters Version 2
- [14] Eigene Grafik
- [15] Bildschirmfoto der Software Geogebra

6.2 Vielen Dank an:

- Thommas Konnemann für die Betreuung des Projektes
- Christian Ambros für die Unterstützung bei jeglichen physikalischen, mechanischen und softwaretechnischen Fragen und die stetigen Rückmeldungen während des Projektes.
- Jan Henrik Hemsing für die Unterstützung beim Ätzen der Platinen und Beantworter sämtlicher Fragen zur Elektrotechnik.
- Jörg Frede für die Bereitstellung seines 3D Druckers und Hilfe bei allen Modellbautechnischen Fragen.
- Phillip Claves für die Hilfe bei Fragen zur Firmwareentwicklung.
- Den Verein Warpzone e.V. für die Bereitstellung der Werkstätten, Werkzeuge und Messgeräte.
- Die Friedensschule Münster für die Finanzierung des Projektes.