

Selbstbalancierende Rakete 3.0

Projekt von Philipp Herrmann und Simon Schappacher

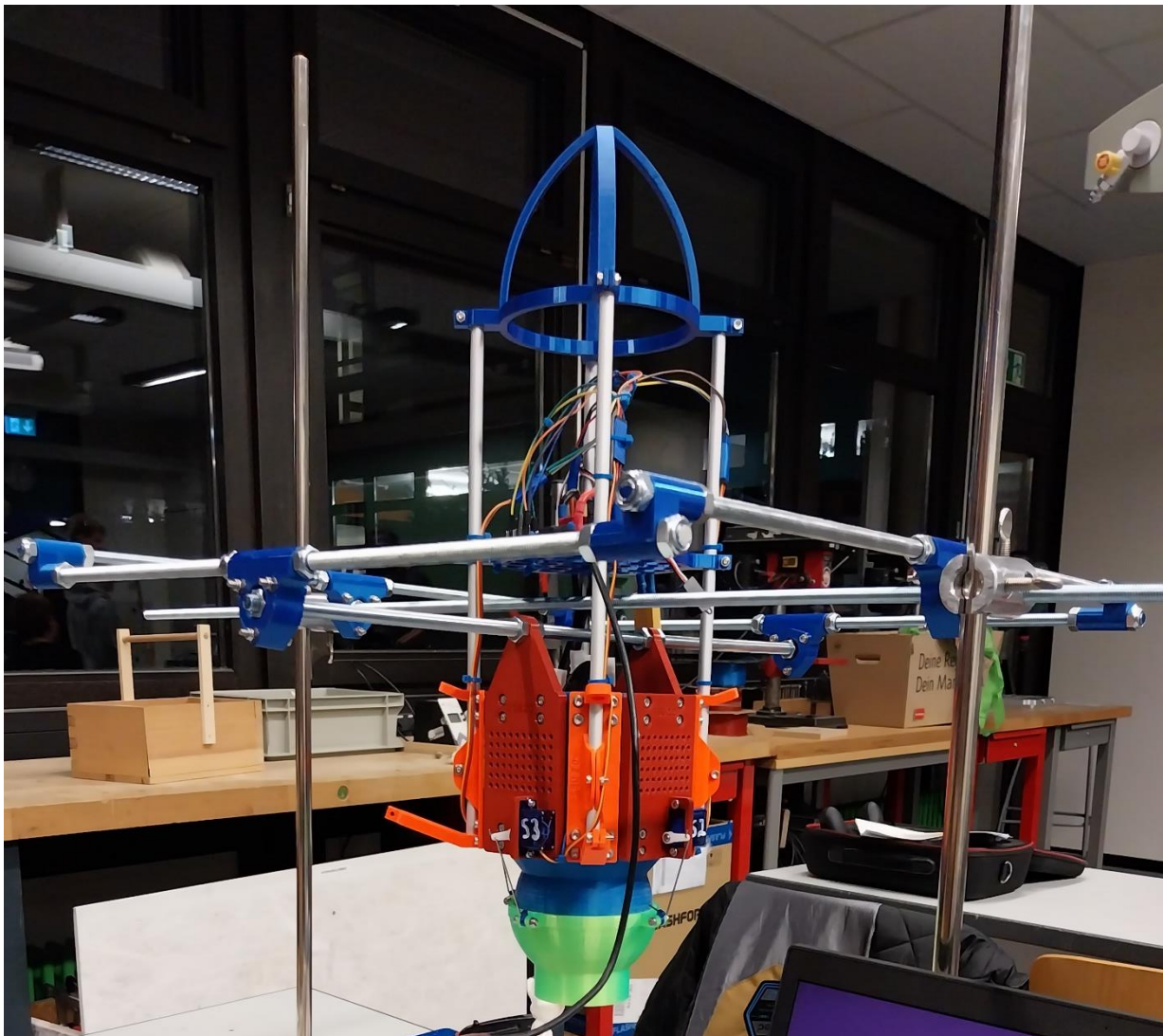
Kurzfassung:

Das Projekt baut auf zwei früheren Versionen auf und untersucht eine dritte Option um raketengetriebene Flugobjekte im Flug zu stabilisieren und zu steuern.

Die aktuelle Ausführung des Projekts basiert auf einer in der Ebene beweglichen Düse. Dadurch liegen keine Bauteile direkt im Abgasstrahl des Triebwerks (wie bei der ersten Rakete) und es werden keine zusätzlichen Triebwerke benötigt (siehe zweite Rakete).

Zentrale Anforderungen sind:

- Mechanisch funktionsfähige schwenkbare Düse
- Programmierung der Ansteuerung (Verspannungen sind möglich)
- Programmierung der Flugsteuerung



2. Inhaltsverzeichnis:

- 1. Titelseite und Kurzfassung**
- 2. Inhaltsverzeichnis**
- 3. Ursprung des Projekts / Bewertung vergangener Iterationen**
- 4. Theoretisches Prinzip**
- 5. Materielle Umsetzung**
- 6. Entwicklung**
 - 6.1. Struktur**
 - 6.2. Düse**
 - 6.3. Software**
 - 6.3.1. Korrektur einbaubedingter Ungenauigkeiten**
 - 6.3.2. Korrektur des nicht linearen Servoausschlags**
 - 6.3.3. IMU - MPU6050**
 - 6.3.4. PID – Controller**
 - 6.3.4.1. P-Proportional**
 - 6.3.4.2. I-Integral**
 - 6.3.4.3. D-Derivative**
- 7. Yaw – Problem**
- 8. Bewertung des Ergebnisses**
- 9. Erweiterungsmöglichkeiten und Ausblick**
- 10. Zusammenfassung**
- 11. Quellenverzeichnis**
- 12. Dank**

3. Ursprung des Projekts / Bewertung vergangener Iterationen

Mit dem Ziel die hohen Kosten in der Raumfahrt zu reduzieren hat sich das Prinzip der Wiederverwendung etabliert. Dieses wurde zunächst durch das Spaceshuttle Programm verfolgt, welches jedoch nicht zu einer signifikanten Verringerung der Startkosten führte. Durch das Unternehmen SpaceX haben sich erstmals wiederverwendbare Raketen durchgesetzt, dabei landen die Hauptstufe (Falcon 9 / Heavy) und ggf. die Booster (Falcon Heavy) nach der Abtrennung wieder auf der Erde und können erneut verwendet werden. Während der Landung wird die Fluglage vorrangig von einer beweglichen Düse gesteuert. Das Schubvektorprinzip wird in der Raumfahrt zwar schon seit langer Zeit verwendet und auch einige Flugzeuge nutzen dieses, jedoch erlangt es durch die Verwendung bei der punktgenauen Landung neue Bedeutung.

Unser Projekt lehnt sich an diese Entwicklung in der Raumfahrt an und untersucht verschiedene Möglichkeiten raketentriebene Flugobjekte zu stabilisieren und zu steuern. Das Projekt stellt dabei die dritte Ausführung einer Serie dar. In dieser Ausführung wird das Prinzip der schwenkbaren Düse angewendet um den Abgasstrahl zu neigen und so ein Drehmoment am Schwerpunkt der Rakete zu verursachen welches die Fluglage verändert.

Dieser Ansatz hat gegenüber den vergangenen Versionen jeweils Vorteile. Bei der ersten Rakete wurden aerodynamische Flächen im Abgasstrom der Rakete platziert welche den Strahl lenkten. Wenn ein echtes Raketentriebwerk verwendet wird sind die Belastungen (v.a. thermisch) auf diese Bauteile jedoch extrem, was schnell zum Verschleiß und Versagen der Bauteile führt.

Bei der zweiten Version der Rakete wurde dieser Ansatz verworfen und es wurden schwenkbare Düsen seitlich an der Rakete angebracht, durch das schwenken der Düsen wurde der Schub relativ zum Boden verändert, dieser Ansatz wurde verfolgt, da der Schub echter Triebwerke nur schwer variiert werden kann. Ein Nachteil war jedoch die zusätzliche Komplexität welche zu zusätzlichem Gewicht führte.

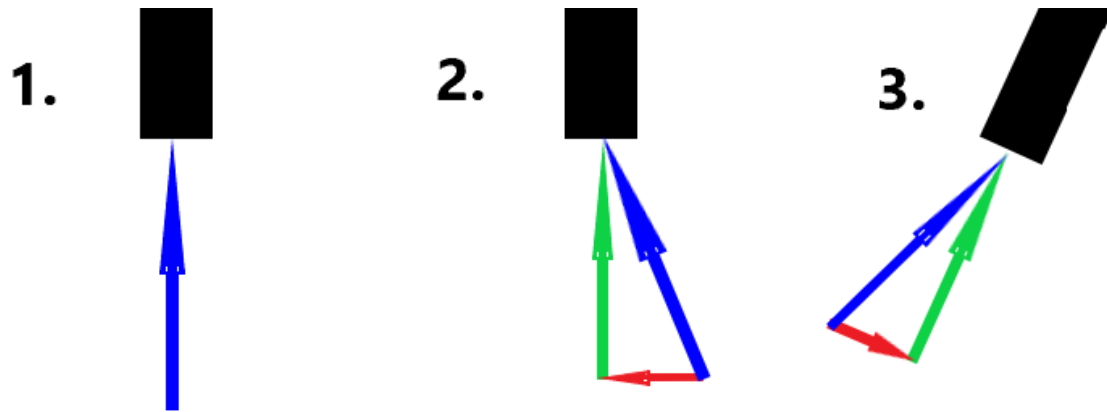
4. Theoretisches Prinzip

Durch die in der Ebene schwenkbare Düse wird es möglich die Fluglage um zwei Achsen zu manipulieren, dabei handelt es sich in unserem Projekt um die Nick- und die Rollachse (engl. pitch / roll). Eine Beeinflussung der Bewegung um die Gierachse ist mit dem gewählten Ansatz nicht möglich.

Wenn die Düse nicht geneigt wird kann die durch das Triebwerk verursachte Krafteinwirkung auf die Rakete als Vektor in Verlängerung der Rakete dargestellt werden (siehe unten 1.).

Wird die Düse im Bild nach rechts geneigt (2.) und die Kraft wieder durch den blauen Vektorpfeil dargestellt, lässt sich dieser in zwei Komponenten zerlegen. Der grüne Pfeil zeigt den Anteil der Kraft in Verlängerung der Rakete. Der rote Pfeil verläuft rechtwinklig zum grünen Pfeil und stellt die Kraft dar, welche die Rakete neigt, dabei übt der rot visualisierte Teil der Kraft ein Moment am Schwerpunkt der Rakete aus welches zu der genannten Neigung führt.

Rechts im Bild (3.) wird die fehlerhafte Fluglage durch das Neigen der Düse nach links (erkennbar an blauem Pfeil) korrigiert, der rote Pfeil zeigt wieder die lateral wirkende Kraft. Diese Bewegung führt zur Reduktion der Schräglage.



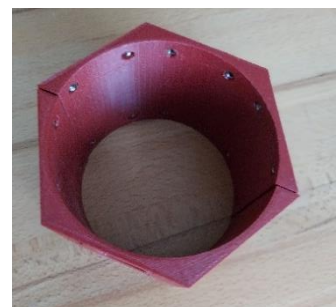
5. Materielle Umsetzung

- **Triebwerk:**
Das Triebwerk wird durch einen elektrischen Impeller dargestellt. Der Impeller wird durch einen 120A Regler gesteuert. Um die erforderliche Spannung zu erreichen werden zwei 3S Lipo Akkus in Reihe geschaltet.
- **Struktur:**
Die grundlegende Struktur wird durch Aluminiumrohre gewährleistet, alle anderen Teile der Struktur werden mit einem 3D-Drucker aus PLA oder PETG gefertigt. Die Verbindungen basieren auf M3-Maschinenschrauben.
- **Steuerung:**
Die Düse wird durch vier Servomotoren über Schubstangen aus Stahl bewegt. Die Servos werden von einem Arduino Mikrocontroller angesteuert. Als Sensor wurde ein MPU6050 Gyroskop und Beschleunigungssensor verwendet.
- **Aufhängung:**
Die vollkardanische Aufhängung der Rakete basiert auf gedruckten Teilen und M8-Gewindestangen.

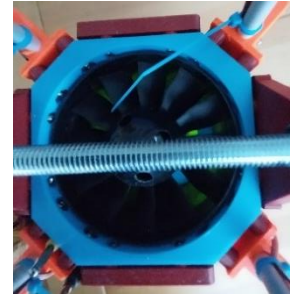
6. Entwicklung

6.1. Entwicklung – Struktur

Als Planungsbasis für die Struktur wurde der elektrische Impeller (kurz EDF) verwendet. Die erste Version der Struktur sah ein sechseckiges Prisma um den Impeller in der Mitte vor. Jeweils drei der Flächen sollten für die Verankerung der Struktur dienen und die anderen drei sollten für die Aufnahme der Akkus und der Servos verwendet werden. Diese Version der Struktur wurde gedruckt und montiert, wobei sich das modulare Design (sechs offene Flächen) als wartungsfreundlich und einfach erweiterbar erwies. Letztlich wurde diese Ausführung verworfen, da es nicht möglich war die Akkus symmetrisch zum Mittelpunkt zu befestigen, Grund dafür war die zwangsläufig gerade Anzahl an Akkus (Option für zwei oder vier Akkus).



Die folgende Version der Struktur führte das modulare Design weiter und basierte auf acht Flächen um den Impeller herum, wobei vier Flächen jeweils kleiner waren und für die Verankerung der Aluminiumstäbe genutzt wurden. Die verbleibenden Flächen dienten wie geplant der Aufnahme der Akkus und der Servos.



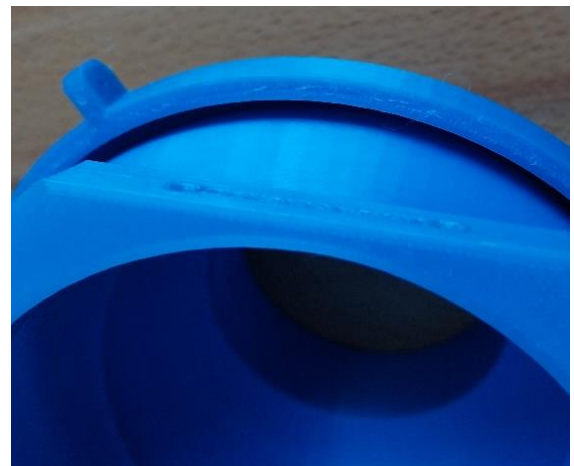
Die Befestigung der Elektronik wurde auf einer Platte mit Wabenmuster realisiert, welche über dem Lufteinlass des Impellers platziert wurde, so konnte der durch das Ansaugen der Luft entstandene Luftstrom gleichzeitig für die Kühlung des Reglers genutzt werden. Das Wabenmuster verringerte außerdem das Gewicht. Die meisten Bauteile wurden mit Kabelbindern auf der Platte befestigt.



6.2. Entwicklung – Düse

Die Beweglichkeit der Düse sollte durch zwei Kugelschalen gewährleistet werden, wobei sich die äußere Schale über die innere bewegen kann. An der äußeren Schale ist die Düse befestigt, gleichzeitig sind die Endpunkte der Schubstangen an der oberen Kante der äußeren Schale befestigt.

Die Anzahl der Schubstangen wurde zunächst auf drei festgelegt, da eine Ebene durch drei Punkte festgelegt werden kann und sich gleichzeitig keine Verspannungen bilden können. Verspannungen könnten durch einen vierten Punkt verursacht werden, der im Widerspruch zu den drei anderen Punkten steht. Als das Konzept von einer sechseckigen Struktur auf eine achteckige Struktur abgewandelt wurde, wurde auch der Ansatz mit drei Schubstangen verworfen. Alternativ kamen entweder zwei oder vier Servos mit Schubstangen in Frage, schließlich wurde die Option mit vier Servos gewählt, da so immer eine Schubstange jeder Achse auf Zug belastet wird. Dies ist vorteilhaft, da die Schubstangen irrelevant hohen Zugkräften standhalten können, sich jedoch bei Druckbelastungen schnell durchbiegen und der Belastung ausweichen.

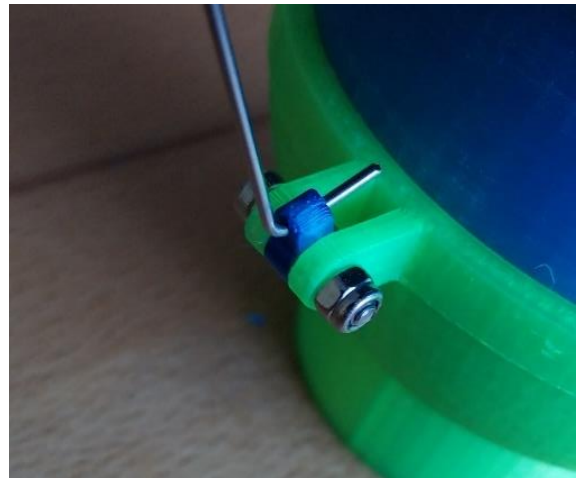


Die erste Ausführung der Düse erfüllte ihren Zweck nicht, da bei Bewegung der Düse ein Spalt zwischen der oberen und der unteren Schale entstand. Es stellte sich heraus, dass die Schale keine Kugel bildete, da bei der Modellierung im CAD-Programm die Mittelpunkte der Bögen nicht auf der Mittellinie zwischen den Bögen lagen.

Die zweite Iteration der Schale erfüllte schließlich die Anforderungen an die Beweglichkeit. Der Entwurf beinhaltete außerdem eine Verkleinerung der Kugelschalen. Als Problem erwies sich jedoch die Befestigung der Schubstangen an der äußeren Schale. Die Schubstangen sind jeweils nur um eine Achse beweglich (die Achse welche durch die Schubstange selbst beschrieben wird), die Düse musste jedoch um zwei Achsen geneigt werden können (pitch / roll). Daraus folgt, dass an der Schale ein zusätzliches Gelenk installiert werden muss, welches die Beweglichkeit um die zweite Achse gewährleistet.



Die dritte Version der Düse beinhaltet das genannte Gelenk und erfüllt so alle Anforderungen.



6.3. Entwicklung – Software

Sowohl die Servos, als auch der Regler werden von einem Mikrocontroller (Arduino Nano) gesteuert. Alle Signale werden digital über Pulsweitenmodulation (PWM) übertragen. Der IMU Sensor MPU6050 wird über die I2C Schnittstelle ausgelesen

Anforderungen an die Software sind:

- Korrektur von einbaubedingten Ungenauigkeiten an der Düse
- Korrektur der nicht linearen Auslenkung der Düse
- Temporär manuelle Steuerungsmöglichkeit über die serielle Schnittstelle
- Ermittlung der Lage des Flugobjekts im Raum aus den Rohdaten des Sensors
- Stabilisierung der Fluglage mit einem PID-Controller

6.3.1. Korrektur einbaubedingter Ungenauigkeiten

Beim Einbau der Schubstangen ergaben sich geringfügige Ungenauigkeiten, da es praktisch nicht möglich war die Schubstangen auf die genaue Länge abzumessen. So kam es zu einer minimalen Fehlstellung der Düse wenn sich alle Servos in der Mittelposition befanden, um diese zu korrigieren musste die Position der Servos geringfügig verändert werden. Diese Korrekturwerte wurden in der Software für jeden einzelnen Servo hinterlegt. Um die Korrekturwerte zu ermitteln war der rechts stehender Aufbau nötig:



Die Rakete wurde mit der Düse auf den Boden gestellt, während auf der Struktur Wasserwagenkerne platziert wurden um die Fehlstellung der Düse zu beurteilen, der im Bild auf der linken Seite zu sehende Wasserwagenkern war für die Pitch-Achse zuständig, der rechte für die Roll-Achse. Über die Serielle Schnittstelle wurden Befehle an die Rakete gegeben, die einzelnen Servos auf eine Bestimmte Position zu bewegen. Ein Befehl bestand aus fünf Zeichen nach dem folgenden Aufbau:

<Nummer des Servos (1 bis 4)>#<Position in Grad in dreistelliger Form>

Um z.B. den zweiten Servo auf die Position 70 Grad zu bringen musste somit folgender Befehl gegeben werden: 2#070.

Zu Beginn des Experiments wurden alle Servos auf die Position 30 gebracht und dann stückweise an die Mittelposition angenähert.

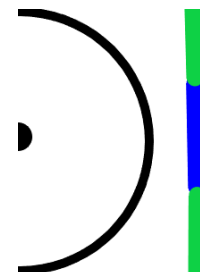
Die Korrekturwerte für die einzelnen Servos ergaben sich aus:

Korrekturwert = 90 – Experimentell ermittelte Mittelposition

6.3.2. Korrektur des nicht linearen Servoausschlags

Das zweite Problem welches durch die Software zu korrigieren war ist die nicht lineare Auslenkung der Düse.

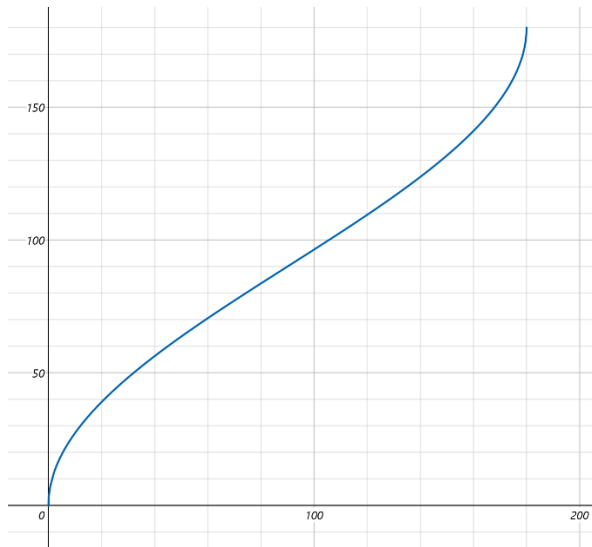
Wenn der Servo im blau markierten Bereich bewegt wird, führt eine Winkeländerung zu einer großen Bewegung relativ zur vertikalen Achse, im grün markierten Bereich führt die gleiche Winkeländerung des Servos zu einer wesentlich kleineren Bewegung relativ zur vertikalen Achse. Die Veränderung relativ zur vertikalen Achse soll für alle Bereiche auf der Kreisbahn gleich sein, daher muss die Software in den Randbereichen der Grafik eine größere Winkeländerung des Servos veranlassen wie in der Mitte.



Die Beziehung zwischen der erwünschten Auslenkung (x) und der erforderlichen Stellung des Servos (f(x)) wird durch die Arcsin-Funktion beschrieben.

Um die Definitionsmenge auf dem Bereich [0; 180] und die Wertemenge ebenfalls auf dem Bereich [0; 180] abzubilden wird folgende Funktion verwendet:

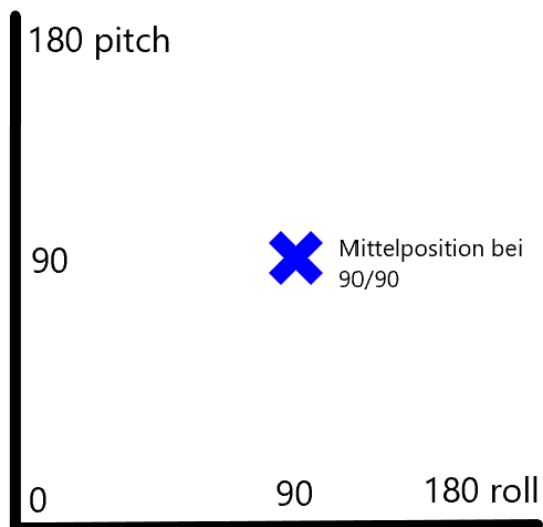
$$f(x) = \frac{180}{\pi} * \arcsin\left(\frac{x}{90} - 1\right) + 90$$



Anhand des Graphen sieht man, dass der Servoausschlag in den Randbereichen höher sein muss um die selbe Änderung des relativen Ausschlags zu erreichen wie in den Mittelbereichen. Die genannte Funktion wurde in der Software hinterlegt.

Die Befehle sollten schließlich über die serielle Schnittstelle an den Arduino gesendet werden.

Das Koordinatensystem für Pitch und Roll lautet wie folgt:



6.3.3. IMU – MPU6050

Als Sensor für die Feststellung der Fluglage (inertial **m**esaurement **u**nit) wurde das Sensormodul MPU6050 verwendet, dabei handelt es sich um einen Beschleunigungssensor (= Accelerometer) und ein Gyroskop, welche beide in einem MPU6050 Chip verbaut sind. Beim Chip handelt es sich um ein MEMS (Micro electro-mechanical system).

Der Beschleunigungssensor misst die Beschleunigung auf den drei Raumachsen, während das Gyroskop die Winkelgeschwindigkeit der Drehung des Sensors um die drei Raumachsen misst.

In unserer Rakete wurden die Daten über die I2C-Schnittstelle (Pins SDA und SCL) an den Arduino übertragen, die Raumlage wurde sowohl mithilfe der Beschleunigungswerte als auch der Werte des Gyroskops ermittelt, dabei sollte das Gyroskop bevorzugt werden, da es die Aufgabe präziser erfüllt. Dabei kommt es jedoch zu dem Problem, dass sich der Messfehler des Gyroskops über erweiterte Zeiträume aufaddiert und nach einer gewissen Zeitspanne der Wert nur noch bedingt brauchbar ist. Um dies auszugleichen wurde die Raumlage zu 98% durch die Werte des Gyroskops und zu 2% durch die Werte des Accelerometers festgelegt. Dieses Verhältnis wurde der Website der „Electronoobs“ entnommen (siehe Quellenangabe).

Die Rohdaten des Accelerometers setzen sich wie folgt zusammen: für jede Raumachse (x, y, z) wurde ein Wert zwischen -32768 und 32768 bereitgestellt. Der Maximalwert stellte dabei eine Beschleunigung von minus, bzw. plus 2g dar. Die Werte wurden durch 16384 geteilt um die Beschleunigung als Vielfaches der Erdbeschleunigung (1g) angeben zu können. Die Raumlage sollte als Winkel auf der Pitch-, bzw. Rollachse angegeben werden, diese Winkel wurden durch die trigonometrischen Funktionen (hier atan) berechnet.

In den unten aufgeführten Formeln wurden $accX$, $accY$ und $accZ$ bereits durch 16384 geteilt und so auf 1g skaliert.

$$\text{Pitch: } \varphi = \text{atan} \left(\frac{accY}{\sqrt{accX^2 + accZ^2}} \right)$$

$$\text{Roll: } \varphi = \text{atan} \left(\frac{accX}{\sqrt{accY^2 + accZ^2}} \right)$$

Die Rohdaten des Gyroskops mussten ebenfalls erst noch bearbeitet werden, damit sich ein Winkel für die Pitch-, bzw. Rollachse ergab. Gemäß dem Datenblatt des Sensors mussten die Werte des Gyroskops erst durch 131 geteilt werden um die Winkelgeschwindigkeit in der Einheit [deg / s] zu erhalten. Danach musste der entstandene Wert mit der seit der letzten Messung vergangenen Zeitspanne multipliziert werden um aus der Winkelgeschwindigkeit die Winkelabweichung zu erhalten ($\Delta\varphi = \omega * \Delta t$).

6.3.4. PID-Controller

Um die Fluglage zu stabilisieren sind technisch einfachere Controller wie On-Off Controller (= bang-bang Controller) nicht verwendbar, die Oszillation um den Zielwert ist bei dieser Anwendung nicht hinnehmbar, daher wurde der auch in vielen Industrieprozessen übliche PID-Controller verwendet. PID-Controller eignen sich um empfindliche Systeme präzise zu steuern, jedoch ist ihre Implementierung, sowie v.a. das Tuning sehr anspruchsvoll. PID setzt sich aus den Begriffen Proportional, Integral und Derivative zusammen, welche Subkomponenten darstellen, die in einem PID-Controller verbaut sind.

6.3.4.1. P-Proportional

Diese Komponente erfasst den aktuellen Fehler und berechnet auf dessen Basis einen Korrekturwert, welcher an die korrigierenden Einheiten geschickt wird (hier die Servos). Der Korrekturwert wird errechnet, indem der aktuelle Fehler mit einem anwendungsspezifischen Faktor multipliziert wird.

Die P-Komponente hat die Schwäche, dass sie bei sehr kleinen Fehlern nur sehr langsam korrigiert, außerdem hat sie keine Information über die Änderungsrate in der Fehlstellung.

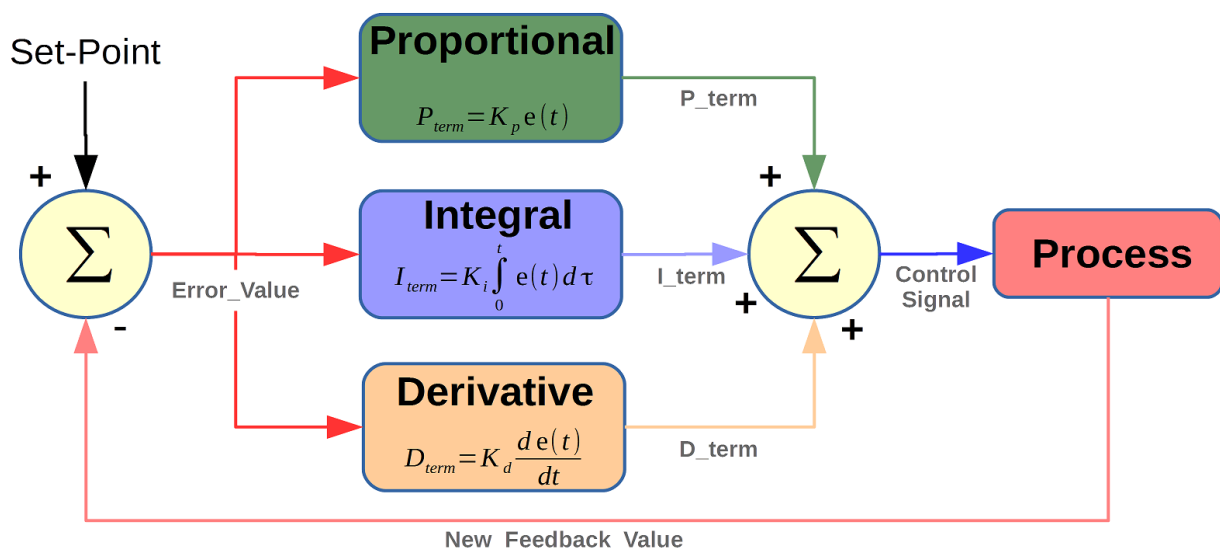
6.3.4.2. I-Integral

Die integrale Einheit beobachtet die Entwicklung des Fehlers in der Vergangenheit und erhält ihren Wert aus der Dauer und der Intensität des Fehlers, sie dient dazu kleine Fehler zu korrigieren, indem sie stärker eingreift, falls der Fehler bereits länger besteht.

6.3.4.3. D-Derivative

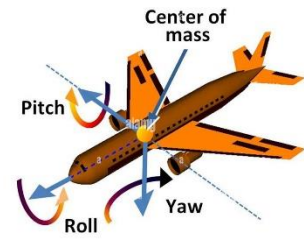
Dieser Teil erstellt aus der momentanen Änderungsrate des Fehlers eine Prognose für die Zukunft und hilft z.B. dabei den Korrekturwert bei Annäherung an den Soll-Wert zu reduzieren.

Alle drei Komponenten nehmen Einfluss auf den Output des Reglers, die von den einzelnen Einheiten ermittelten Korrekturwerte werden addiert, wobei die Einflussgröße der einzelnen Einheiten jeweils anwendungsspezifisch gewichtet wird. In unserer Rakete ist aufgrund der hohen Komplexität zunächst der Einsatz eines P-Controllers, mit Option auf eine Erweiterung mit I- und D-Komponenten geplant. Der P-Controller befindet sich zum aktuellen Zeitpunkt in der Entwicklung. Der Controller wird aufgrund des großen Unterschieds der Rakete zu einem Quadcopter nicht auf Basis von bestehender Software entwickelt, sondern von Grund auf selbst implementiert.



7. Yaw – Problem

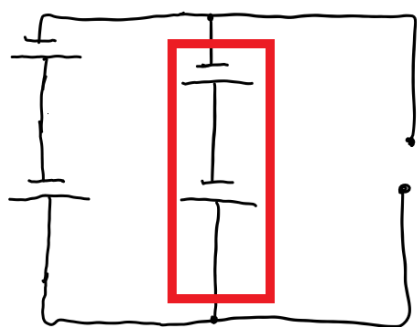
Während der Entwicklung der Rakete erwies sich die Eigendrehung der Rakete als größtes Problem. Da bei unserem Projekt kein echtes Raketentriebwerk verwendet werden kann und ein elektrischer Impeller verwendet wird kommt es zur Eigendrehung der Rakete. Der Impeller erzeugt sowohl im Betrieb bei gleich bleibenden Schub als auch besonders in der Start- und Landephase wenn der Schub erhöht bzw. verringert wird ein signifikant hohes Drehmoment um die Yaw-Achse (= Gierachse). Die Rakete hat keine Möglichkeit mit ihrer Schubvektordüse diese Yaw-Störung auszugleichen.



Die Yaw-Störung hat zwei Quellen. Die erste Quelle ist der aus der Schifffahrt bekannte Radeffekt, dieser tritt aufgrund der niedrigeren Dichte und Viskosität von Luft zwar weniger stark auf, reicht aber aus um die Rakete außer Kontrolle geraten zu lassen. Die zweite Quelle wird während der Veränderung der Schubkraft der Rakete wirksam und ist wesentlich stärker wie die erste, sie geht aus der Impulserhaltung hervor und ist ebenfalls unumgänglich.

In der Luftfahrt wird dieses Problem von Helikoptern gelöst, indem ein Heckrotor eingebaut wird, der die Störung ausgleicht, dies ist in einer Rakete unpraktikabel. Deshalb wurde bei dieser Version der Rakete auf Freiflüge verzichtet und eine vollkardanische Aufhängung entwickelt, welche eine Bewegung der Rakete um die Yaw-Achse unterbindet.

Unsere Aufhängung basiert auf M8-Gewindestangen und 3D-gedruckten Teilen. Die Rakete wird fest mit einer M8-Gewindestange verschraubt, wofür zwei gegenüberliegende Akku-Aufnahmen geringfügig modifiziert wurden. Die veränderten Aufnahmen verloren dabei ihre Fähigkeit Akkus aufzunehmen, dienten jedoch weiterhin als Aufnahmen für die Servos. Die Rakete konnte auch mit einer Versorgung durch zwei Akkus auskommen, jedoch halbierte sich so die Akkukapazität, schwerwiegender war es, dass sich die maximale Entladerate ebenfalls halbierte, jedoch ist für diese Anwendung der maximale Schub des Impellers nicht erforderlich, weshalb auch dies hinnehmbar war. Die Verschaltung der Akkus änderte sich dabei wie aus der Abbildung zu entnehmen ist, der rot markierte Teil wurde entfernt:



6.000mAh → 3.000mAh



8. Bewertung des Ergebnisses

Eine Steuerung von Flugkörpern auf Basis einer neigbaren Düse kann anders als Steuerflächen im Abgasstrom als langlebig angesehen werden und weist eine mechanisch geringere Komplexität auf wie eine Lösung mit mehreren Düsen. Der große Nachteil im Vergleich zu anderen Optionen ist, dass eine Beeinflussung der Bewegung um die Yaw-Achse nicht möglich ist und diese wenn erforderlich durch sekundäre Lagekontrollsysteme realisiert werden muss.

Das untersuchte Prinzip ist gut zu gebrauchen, wenn ein Raumflugkörper in der Atmosphäre eines Himmelskörpers gesteuert werden muss, da es im Vergleich zu anderen Methoden hohe Kräfte aufwenden kann und starke Lagestörungen in der Atmosphäre somit besser ausgleichen kann. Außerdem ist es außerhalb der Atmosphäre effektiv, da keine Flächen verwendet werden, welche im Luftstrom liegen müssen.

Auch für die Steuerung von Raumsonden ist dieses Prinzip interessant, da das Prinzip auch auf andere Triebwerkstypen welche keine Drehung um die Yaw-Achse verursachen angewendet werden kann. Besonders kommen hier Ionentriebwerke in Frage, die Lagesteuerung würde dann von der Treibstoffeffizienz des Ionentriebwerks profitieren, da keine Gase ausgestoßen werden, die einen niedrigeren spezifischen Impuls hätten.

Es dürfte sich jedoch als schwierig erweisen Software zu entwickeln, welche den Flugkörper automatisch steuert, da das Prinzip von dem von Quadcoptern so weit entfernt ist, dass es nur bedingt möglich ist existierende Software entsprechend anzupassen.

9. Erweiterungsmöglichkeiten und Ausblick

Um Freiflüge der Rakete zu ermöglichen würde es sich anbieten zwei Drohnenmotoren oder kleinere elektrische Impeller an kurzen Auslegern an der Rakete anzubringen um die Yaw-Störung auszugleichen. Es müsste dann eine weitere Achse in den PID-Controller eingebunden werden, sowie die Stromquelle modifiziert werden um geringere Spannungen abgreifen zu können.

Für weitere Iterationen der Rakete würde sich ein System auf Basis von Steuerdüsen anbieten, welche die Fluglage durch das seitliche Ausstoßen von Gasen verändern. Dabei könnte in der Praxis Druckluft genutzt werden. Um den zentralen Schub zu erzeugen würde wieder der elektrische Impeller verwendet werden.

Ein anderes Prinzip, welches jedoch ein chemisches Triebwerk voraussetzt und so durch das hohe Gewicht und die hohe Gefährdungslage von Flugversuchen ausgeschlossen ist wird verwendet um den Schub von Feststofftriebwerken zu lenken, dabei wird ein reaktives Fluid seitlich in den Abgasstrahl eingeleitet, dies verändert die Richtung des Abgasstrahls.

10. Zusammenfassung

Die mechanische Umsetzung des Projekts erwies sich als gut machbar, wobei einige Herausforderungen beim Bau der Düse bestanden. Die Software welche zur Steuerung der Düse benötigt wird war wesentlich schwieriger zu implementieren, da zum einen bauliche Ungenauigkeiten ausgeglichen und zum anderen die Auslenkung der Servos durch die Software korrigiert werden mussten. Besonders die Ermittlung der Fluglage und der PID-Controller stellten den Schwerpunkt der Entwicklung der Software dar.

Die Entwicklung eines freiflugfähigen Demonstrators kam mit den gegebenen Mitteln nicht in Frage, da der Triebwerkstyp für das Konzept unpassend war. Die Flugsteuerung musste auf von Grund auf selbst entwickelt werden, was bessere Einblicke in die Regeltechnik ermöglicht hat, jedoch auch wesentlich aufwendiger war, wie eine vorgefertigte Lösung. Ein Freiflug hätte das Lösen des Yaw-Problems vorausgesetzt, was eine Entfernung vom Ursprünglichen Konzept in Richtung Helikopter zur Folge gehabt hätte.

Die Rakete in ihrer Aufhängung zeigt dennoch, dass es prinzipiell möglich ist den Flugkörper um zwei Achsen zu steuern und so die Flugbahn zu stabilisieren indem man die Düse neigt.

11. Quellenverzeichnis

https://en.wikipedia.org/wiki/Thrust_vectoring Verschiedene Informationen bezüglich des Prinzips

<https://duckduckgo.com/?q=pid+controller+schematic&va=b&t=hr&iar=images&iar=images&iar=images&iai=https%3A%2F%2Fwww.theengineeringconcepts.com%2Fwp-content%2Fuploads%2F2018%2F11%2FPID-CONTROLLER-BLOCK-DIAGRAM.png> PID-Controller Grafik

<https://www.youtube.com/watch?v=4cw9K9yulyU> PID / Flugsteuerung Basiswissen

<https://www.alamyimages.fr/aggregator-api/download?url=https://c8.alamy.com/compfr/krf3ty/pitch-roll-yaw-krf3ty.jpg> Yaw, pitch, roll Grafik

http://electronoobs.com/eng_robotica_tut6_1.php Electronoobs - Erklärung

http://electronoobs.com/eng_robotica_tut6_2.php Electronoobs - Code

12. Dank

Herzlichen Dank an Herrn Wolff, Lehrer am Marie-Curie Gymnasium Kirchzarten, der die AG Mikrocontroller und Automatisierung des Freiburg Seminars leitet. Das Projekt entstand vorrangig in dieser AG