

Autonomer Apportiercopter für Bälle

Ein Projekt von Marc Panse, Hannes Reichle und Marie Schurna

Kurzfassung

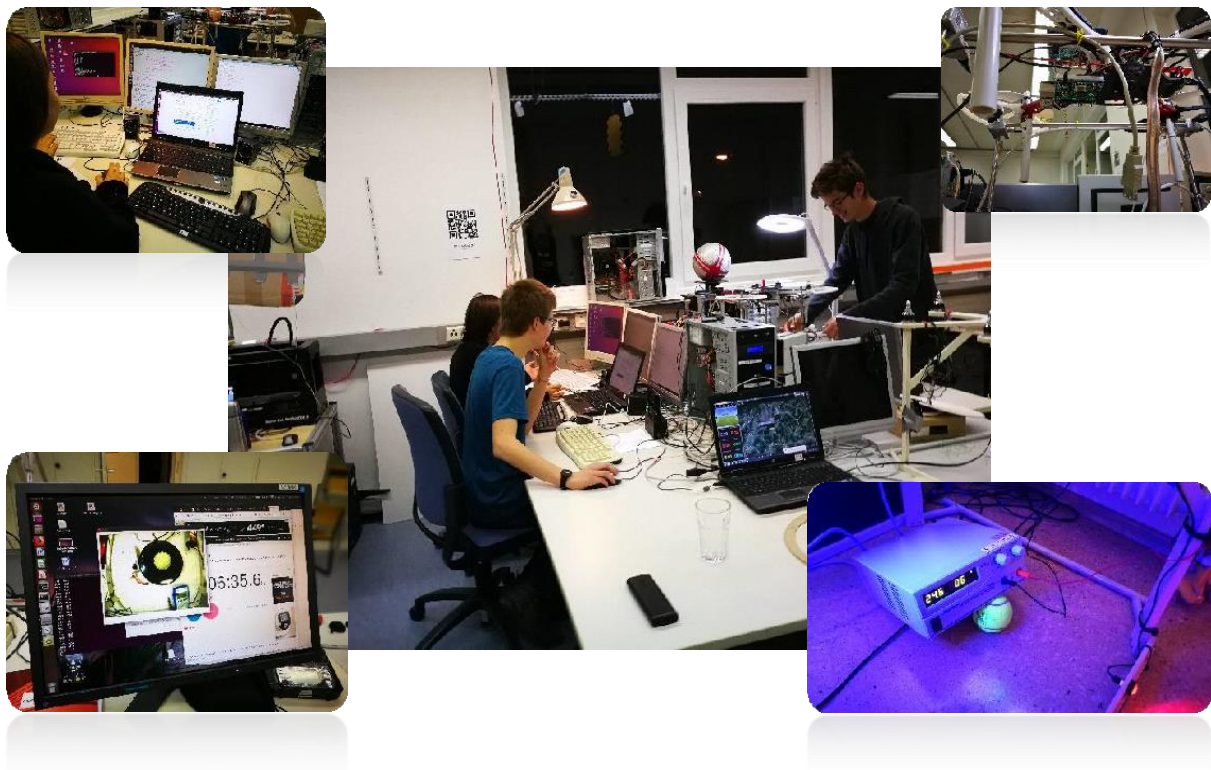
Das Ziel unseres Projekts ist, die Arbeit von Hausmeistern an Schulen und Sporthallen in naher Zukunft zu erleichtern.

Dafür wurden ein Quadro- und ein Hexacopter entwickelt, für die wir eine Bildverarbeitung programmiert haben, welche Tennisbälle automatisch erkennt und autonom aufammelt.

Zentrale Aspekte unseres Projektes sind hierbei die Entwicklung von:

- Bildverarbeitung
- Steuerungssoftware
- Copter
- Ballaufnehmer

Und letztlich die Verlinkung aller Teilgebiete.



Inhaltsverzeichnis

Kurzfassung	1
1. Einleitung	3
1.1 Ideenfindung	3
1.2 Wie kann so eine Lösung realisiert werden?	3
2. Vorgehensweise, Materialien und Methode	4
2.1 Entwicklung des Quadrocopters.....	4
2.2 Versuch zur Bestimmung der Stabilität von Landebeinen	6
2.3 Entwicklung der Software.....	6
2.4 Versuch zur Bestimmung der optimalen Farbwerte	7
2.5 Entwicklungen zur Bilderkennung mit niedriger Latenz	8
2.5.1 Raspberry als OnBoard PC	8
2.5.2 WLAN Verbindung.....	8
2.5.3 FPV-System	9
2.6 Live-Steuerung des Copters	9
2.7 Entwicklung der Steuerung des Copters	10
2.7.1 Steuerung bei gleichbleibender Höhe und Ausrichtung	11
2.7.2 Steuerung bei wechselnder Höhe und Ausrichtung.....	11
2.8 Zweite Copter-Plattform, Bau des Hexacopters	12
2.9 Entwicklung des Ballaufnehmers	12
3. Versuch zur Bestimmung der maximalen Steiggeschwindigkeit	14
4. Versuch zur Bestimmung der Effizienz	16
6. Zusammenfassung	17
7. Quellen- und Literaturverzeichnis	18
8. Unterstützungsleistung	18

1. Einleitung

1.1 Ideenfindung

Unser Projekt wurde in der Elektronik AG am Graf-Eberhard-Gymnasium in Bad Urach geboren. Auf der Suche nach einem Projekt schauten wir gedankenverloren aus dem Fenster unseres Physiksaales. Dabei sahen wir den Hausmeister auf dem Dach der Turnhalle (Ermstalhalle) Bälle einsammeln. Dies gestaltete sich aufgrund der unterschiedlichen Höhen der Dächer (Abbildung 1) als sehr mühsam. Wir haben uns als Ziel gesetzt, eine sichere und komfortable Lösung dafür zu finden.

Einsatzgebiet unserer Erfindung, Höhenunterschiede farbig markiert

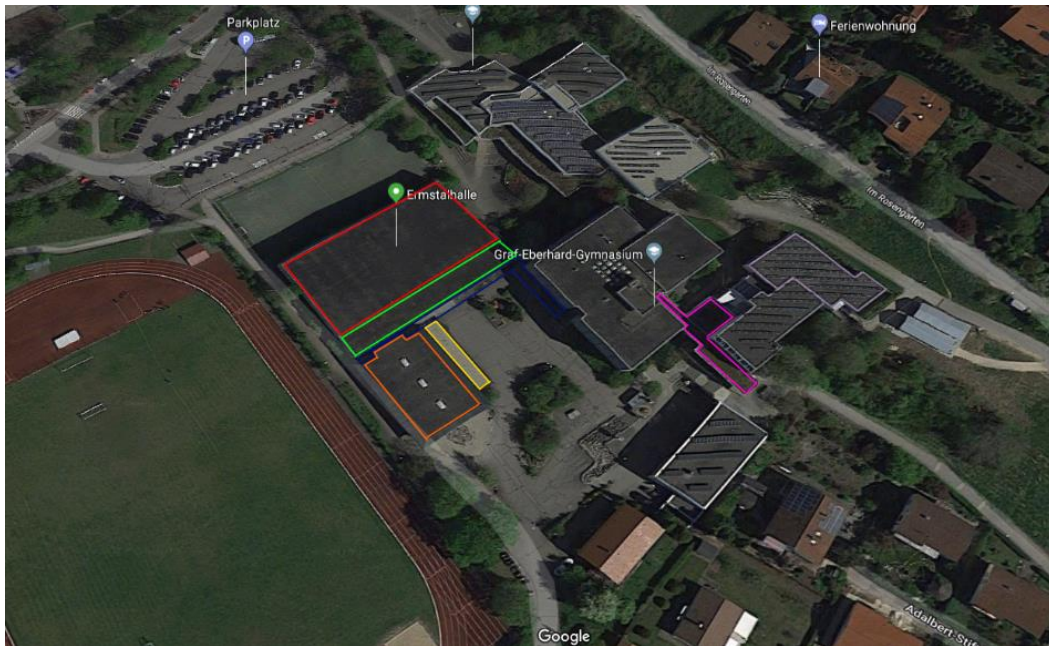


Abbildung 1, Graf-Eberhard-Gymnasium Bad Urach, Screenshot Google Earth, 20.12.2017

1.2 Wie kann so eine Lösung realisiert werden?

Schnell merkten wir, dass die Zeit, die uns in der Schule zur Verfügung steht, nicht ausreichen würde, um diese komplexe Forschungsarbeit umzusetzen. Zufällig war der Leiter der Elektronik AG auch der Leiter des Schülerforschungszentrums (SFZ) in Eningen unter Achalm. Das Projekt wurde deshalb ab dem 17.06.2015 im SFZ weitergeführt.

Sogenannte Multicopter, die zum Zeitpunkt der Projektgründung relativ neu auf den Markt gekommen waren, eignen sich hervorragend für dieses Vorhaben, da sie eine hohe Traglast besitzen und auf der Stelle schweben können.

Die Idee, Multicopter für Aufgaben zu benutzen, gibt es schon länger. Schon 2013 zeigte Raffaello D'Andrea mit seiner Rede bei TED "The astounding athletic power of quadcopters" (Q1), wie Copter Probleme lösen können und welche weitreichenden Einsatzgebiete möglich sind.

In unserer Forschungsarbeit wollen wir uns damit beschäftigen, ob und wie eine solche Nutzung realisierbar ist. Zum Zeitpunkt des Forschungsanfangs gab es schon Ballerkennungsalgorithmen, welche Bälle (und auch Objekte) erkennen können, sowie Copter, die mithilfe einer Infraroterkennung (mit sogenannten Beacons) eine zielsichere Landung schaffen. Jedoch gibt es noch keine Kombination, welche eine zielsichere Landung mithilfe einer Bilderkennung ermöglicht.

Dies liegt wohl daran, dass die "Beacon"-Systeme (Q2) zweiteilig sind, dass also immer ein Teil auf dem Boden sein muss, über welchem zentriert wird. Zudem ist Objekterkennung, welche ohne große Latenz

funktionieren soll, relativ rechenaufwendig. Dies stellt auf einem Flugobjekt, welches immer so leicht wie möglich sein sollte, ein Problem dar.

Unser Schwerpunkt liegt auf der Entwicklung der Software, des Copters und des Ballaufnehmers. Zudem muss eine Möglichkeit gefunden werden, die Bilderkennung mit niedriger Latenz aber mit Verbindung zur Flugsteuerung des Copters zu verwirklichen, da sonst die sichere Steuerung nicht mehr gewährleistet ist. Als Quadrocopter kam zunächst ein F450 von DJI zum Einsatz, in Kombination mit der Open Source Flugsteuerung APM. Als Kamera und Recheneinheit wurde ein Raspberry Pi mit Pi-Cam verbaut, auf dem ein gewöhnliches Raspbian installiert wurde. Später wurde auf einen selbst gebauten Hexacopter mit der Flugsteuerung Pixhawk 2 umgeschwenkt, welcher mit einem Small-Factor-PC ausgerüstet ist, auf dem die Bildverarbeitung läuft.

2. Vorgehensweise, Materialien und Methode

2.1 Entwicklung des Quadrocopters

Beim Planen des Projektes fingen wir mit der Grundplattform an, da an ihr alle Komponenten befestigt werden mussten. Wir stießen hierbei auf das F450-Set von DJI in Verbindung mit der Open-Source Flugsteuerung ArduPilot Mega (APM), welcher einen etwas umkonfigurierten Arduino darstellt. Dies stellte, da die Komponenten nur in zwei verschiedenen Sets gekauft werden mussten, eine enorme Vereinfachung für uns dar, da wir nach einiger Recherche immer noch wenig Ahnung davon hatten, wie ein Copter letztendlich im Detail funktioniert und welche Spezifikationen die benötigten Teile brauchen.

Beim Zusammenbauen der Komponenten berücksichtigten wir den Abstand, der zum Tennisball gehalten werden musste, da der F450-Frame von sich aus nur wenige Zentimeter Bodenfreiheit hat.

Dazu haben wir verschiedene Ideen umgesetzt:

1. Standfüße aus Fahrradspeichen

Dieses Material wurde eingesetzt, da es leicht und stabil schien. Jedoch stellte sich schnell heraus, dass diese nicht geeignet waren, da sie durch den kleinen Querschnitt in sich keine Stabilität aufwiesen. Der Copter fing schon bei kleinsten Berührungen an zu wackeln. Der Copter ist in Abbildung 2 dargestellt.



Abbildung 2, Der eingesetzte Copter, Foto: Marie Schurna, 28.11.2015

2. 3D-gedruckte Standfüße

Davon wurden zwei Versionen getestet:

- a) Die ersten Standfüße wurden nach einer Vorlage aus dem Internet (Q3) angefertigt. Diese sind etwas nach innen gerichtet, was zu vielen Ausfällen führte, da die 3D-gedruckten Formen aufgrund der hohen Belastung oft zerbrachen. Abbildung 3 zeigt diese:

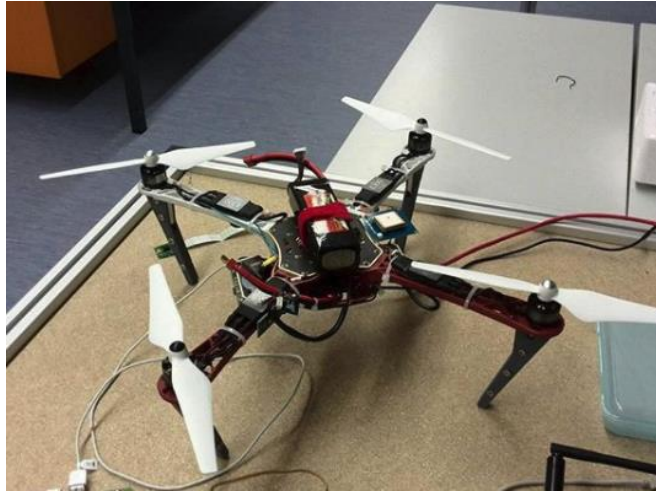


Abbildung 3, Standfüße aus dem 3D-Drucker, Foto: Marc Panse, 18.12.2015

- b) Die zweite Version der Standfüße wurde selbst konstruiert und mit PLA (Polylactide) 3D-gedruckt. Dabei wurde darauf geachtet, dass diese aus hohlen Zylindern bestehen und im rechten Winkel nach unten zeigen. Tests zeigten, dass diese Bauweise zwar eine hohe Stabilität aufwies, jedoch zu schwer und damit nicht geeignet war.

3. Standfüße aus Kabelführungsrohren

Diese wiesen bei sehr geringem Gewicht eine hohe Belastbarkeit auf, zudem sind sie bei Wärme verformbar, was der guten Befestigung diente. Deshalb entschieden wir uns dazu, diese für die finale Version zu verwenden.

Im Folgenden wird der Prozess, den Copter zum Fliegen zu bringen, beschrieben:

Nach der umfangreichen Konfiguration mithilfe der Mission Planner Software war der Copter für den ersten Flugversuch bereit. Der erste Testflug lief jedoch nicht erfolgreich ab. Der Copter hob nicht ab, sondern drehte sich stattdessen auf den Kopf und zerstörte dabei einige Propeller und 3D-gedruckte Teile. Die Suche nach dem Fehler erwies sich als außergewöhnlich schwierig. Viele mögliche Fehlerquellen wurden untersucht und daraufhin ausgeschlossen.

Als mögliche Fehlerquellen wurde erkannt:

- Copter falsch kalibriert: Wir kalibrierten den Copter mehrmals neu und trimmten ihn, jedoch ohne Besserung.
- ESCs (Regler der Motoren; Englisch: „Electronic Speed Controller“) nicht kalibriert, und deshalb ungleichmäßige Ansteuerung der einzelnen Motoren: Jeder Motor wurde einzeln für unsere Fernsteuerung eingelernt, das Problem blieb erhalten.
- Allgemein zu wenig Schub, um abzuheben: Es wurde eine Vorrichtung gebaut, bei der der Copter mit Seilen gehalten wurde, aber die Propeller sich trotzdem noch drehen konnten. Ergebnis: Der Copter durchriss aufgrund des großen Schubs die Seile, drehte sich aber wieder auf den Kopf und flog gegen eine Wand.
- Motoranordnung fehlerhaft: Da wir in deutschsprachigen Foren keine eindeutige Anordnung finden konnten, versuchten wir die richtige Motoranordnung experimentell zu ermitteln. Dies führte immer noch nicht zu dem gewünschten Ergebnis. Jedoch stießen wir durch Zufall auf die englischsprachige Originalanordnung, die uns auf eine Vertauschung zweier Kabel führte.

Nach einigen Parameteranpassungen flog nun der Copter stabil in der Luft.

2.2 Versuch zur Bestimmung der Stabilität von Landebeinen

Durch vertikale Krafteinwirkung sollte ein Wert für die Maximalbelastung der Standfüße ermittelt werden. Dazu bauten wir einen Versuch, siehe Abbildung 4, auf.



Abbildung 4, Messung der Stabilität der Standfüße, Foto: Marie Schurna, 02.01.2018

Es stellte sich heraus, dass dieser Versuchsaufbau (Abbildung 4) ungeeignet ist. In unserem Test kam es zu keiner Zerstörung der Landebeine.

Bei einer unsanften Landung, die beim autonomen Fliegen öfters vorkam, wirken zudem seitliche Kräfte auf die Landebeine, die ein Zerbrechen der Füße zur Folge hatten.

2.3 Entwicklung der Software

Für die Bildverarbeitung bietet sich die Library OpenCV an, da diese viele Funktionen zur Bildverarbeitung beinhaltet. In unserem Fall soll der Copter hellgelbe Tennisbälle apportieren. Aufgrund der hohen Farbabweichung zur Umwelt bietet sich ein Farbfilter an.

Da sowohl OpenCV und DroneKit (Steuerung des Copters) beide mit der Programmiersprache Python funktionieren, haben wir uns für diese entschieden.

Nach der Installation von OpenCV, die sich komplizierter als gedacht herausstellte, konnte ein Farbfilter über das Live-Bild gelegt werden. Dabei wurde ein passender Bereich des Farbfilters experimentell festgelegt. Weitere Befehle, die vorgeben, dass eine bestimmte Pixelanzahl auf einer Fläche im Bereich des Farbfilters liegen muss, sorgen für eine geringere Fehlerquote. Die Mitte dieser Anhäufung wird als Bildpunkt für die weitere Verarbeitung ausgegeben.

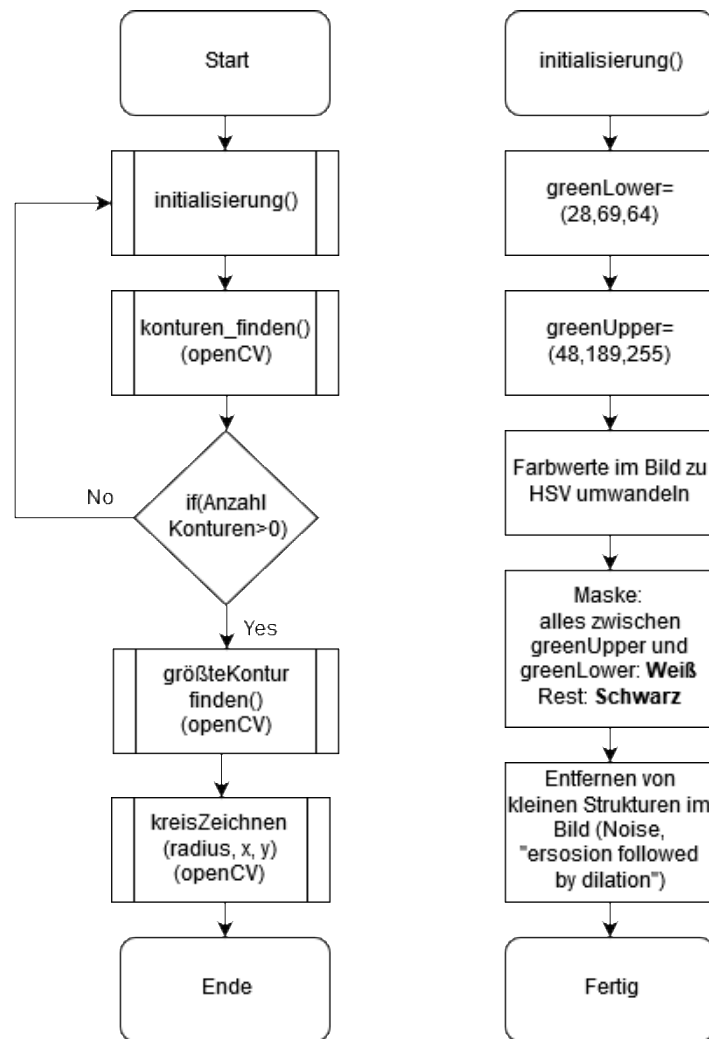


Abbildung 5, Bilderkennung, Programmschema

2.4 Versuch zur Bestimmung der optimalen Farbwerte

Für unsere Bilderkennung nutzen wir einen Farbfilter. Dieser hat eine gewisse Toleranzzone, wie „stark“ die Farbe vom vorgegebenen optimalen Wert abweichen darf. Ist dieser Bereich zu groß, werden falsche Objekte als Ball identifiziert und eine Zentrierung des Balles ist nicht möglich. Wird der Wert hingegen zu gering gewählt, führen kleine Änderungen der Beleuchtung direkt zu einem Nichterkennen des Balles.

Um die optimale Toleranzzone zu finden, wurden Bilder von Tennisbällen in verschiedenen Umgebungen/ Beleuchtungen fotografiert.

Die Farbe eines Balles wird in Open CV in einem HSV Spektrum angegeben.

- **H**: Bezeichnung: „hue“ (Farbwert).
- **S**: Bezeichnung: „saturation“ (Sättigung)
- **V**: Bezeichnung: „value“ (Hellwert)

Für jeden dieser Werte wurde ein Diagramm angefertigt, um den benötigten Bereich zu identifizieren.

Es ergibt sich für den H Wert folgender Graph:

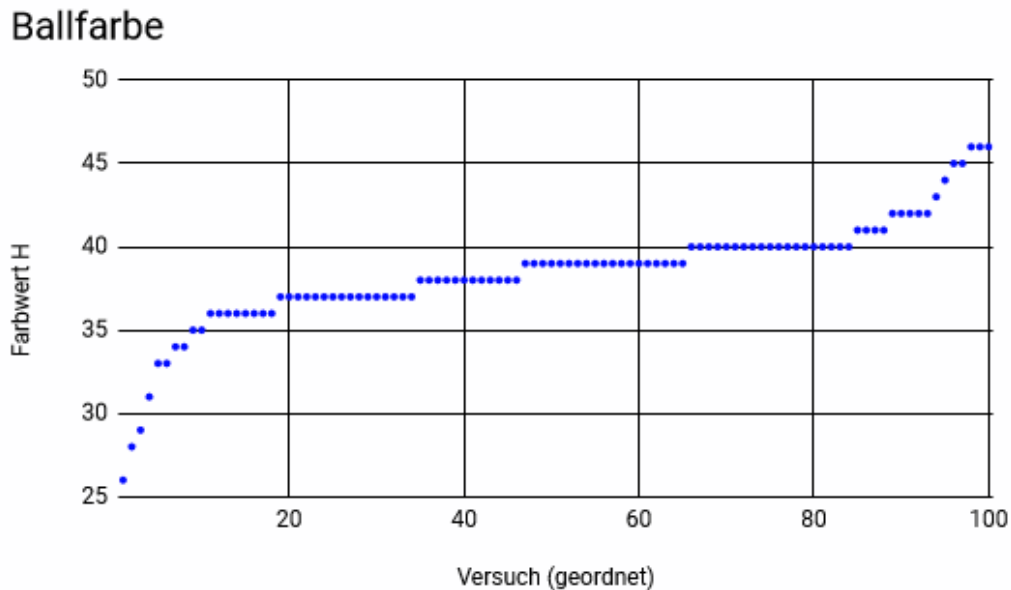


Abbildung 6, Die ermittelte Ballfarbe bei 100 Versuchen

Hierbei ist gut zu sehen, dass der H-Wert, welcher einen Messbereich von 0-180 hat, mit einer Standardabweichung von 3,26 sehr genau auch bei extrem unterschiedlichen Beleuchtungsbedingungen bestimmt werden kann. Die Werte S und V liegen im Gegensatz dazu zufällig über den Messbereich verteilt.

2.5 Entwicklungen zur Bilderkennung mit niedriger Latenz

2.5.1 Raspberry als OnBoard PC

Die ursprüngliche Überlegung war, die Bildverarbeitung auf dem Quadrocopter laufen zu lassen, um Verzögerungen, die durch die Übertragungen entstehen können, zu vermeiden.

Schnell stellte sich jedoch heraus, dass der Raspberry, der auf dem Quadrocopter montiert wurde, zu langsam für dieses Vorhaben ist, da die Hardware nicht für rechenintensive Anwendungen konzipiert wurde. Verzögerungen von mehr als sechs Sekunden waren die Folge.

Computer mit mehr Rechenleistung mussten wir aus Gewichtsgründen ausschließen.

Lösungsansatz: Bilderkennung auf einen externen Desktop PC mittels einer WLAN-Verbindung.

2.5.2 WLAN Verbindung

Mithilfe eines WLAN-Routers wurde ein drahtloses Netzwerk aufgebaut. Wir verwendeten ein Python Skript auf dem Raspberry, das dafür sorgte, dass der Quadrocopter sein Bild über WLAN versenden konnte. Als Ground Station diente ein Desktop PC, der sich per Cat Kabel ins Netzwerk einwählen konnte. Der Desktop PC in Verbindung mit dem Netzwerk konnte aufgrund seiner deutlich größeren Rechenleistung eine Minimierung der Verzögerung auf zwei Sekunden erzielen.

Abbildung 7 zeigt den Versuchsaufbau mit einem gewöhnlichen Heimrouter.

Probeflug mit einer WLAN-Verbindung



Abbildung 7, Aufbau zur Kommunikation mit dem Copter, Foto: Hannes Reichle, 16.03.2016

Beim Testflug kam die Ernüchterung. Die Übertragung findet nur bei hervorragender WLAN-Verbindung statt. Eine Reichweite von durchschnittlich acht Metern konnte nicht überschritten werden.

Lösungsansatz: Beim Recherchieren entdeckten wir sogenannte FPV Systeme (First Person View), die auf verzögerungsfreie Übertragung optimiert sind.

2.5.3 FPV-System

FPV Systeme (First Person View) kommen vor allem im Hobbybereich vor. Mit ihnen wird das Bild einer Kamera vom Copter über ein analoges Funk-Signal zum Empfänger auf dem Boden geschickt. Diese Systeme sind für die geringe Verzögerung der Bildübertragung optimiert und können dadurch zum Steuern von Flugobjekten jeglicher Art über sogenannte FPV Brillen benutzt werden.

Trotz längerem Anpassen der Farbwerte konnte kein zuverlässiger Farbwert für die Bilderkennung ermittelt werden. Dies liegt daran, dass diese Systeme für eine möglichst kleine Latenz (Verzögerung) entwickelt wurden. Dadurch gehen jedoch viele Informationen hinsichtlich der Farbe verloren.

Lösungsansatz: Die Bilderkennung wieder auf den Copter verlagern, um Verluste durch Übertragung zu vermeiden.

2.6 Live-Steuerung des Copters

Die ausgewerteten Daten müssen natürlich auch wieder möglichst verlustfrei an die Flugsteuerung weitergegeben werden. Dies bewerkstelligen im Verlauf des Projektes verschiedene Programmansätze:

Eine direkte Steuerung per emuliertem PWM (Pulsweitenmodulation)-Signal schlossen wir aus, da es bei einer Fehlfunktion dieser deutlich schneller zu einem Unfall kommen kann.

Die erste Grundkonzeption ist die Steuerung per Telemetrie-Modul, mit welchem Daten zwischen Copter und der Bodenstation während eines Fluges übertragen werden können.

Mit der vorhandenen Software unter Windows Mission-Planner und unter Linux der APM-Planner können damit sogar in Echtzeit (unter 0,5s) mit der Maus sogenannte Wegpunkte (GPS Koordinaten zu denen der Copter fliegt) gesetzt werden. (Abbildung 8)

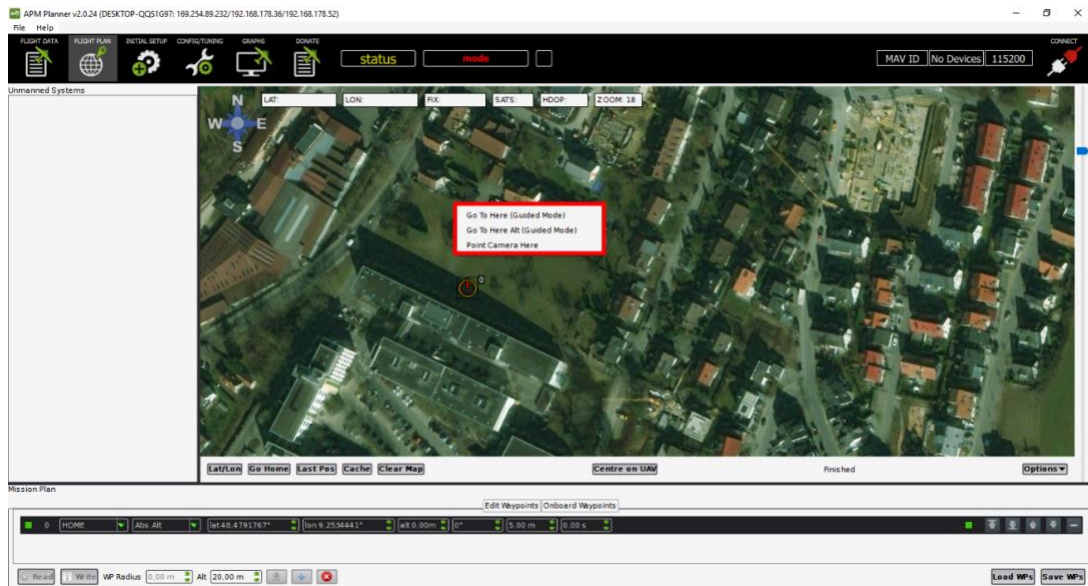


Abbildung 8, Mit der Maus können Wegpunkte gesetzt werden, 01.01.2018, Screenshot APM_Planner

Der Test des Telemetrie-Moduls im Gebäude erwies sich als problemlos, jedoch war während des Fluges keine Verbindung möglich. Deshalb wurde das Modul vibrationsgedämpft installiert, wodurch jedoch keine Verbesserung erzielt werden konnte.

Nun wurden die mitgelieferten Stab-Antennen geöffnet und die Antennenlänge bestimmt. Es stellte sich heraus, dass die Antennen für die Frequenz von 433 MHz ungeeignet waren. Daraufhin wurden nach

$$\lambda = \frac{c}{f}$$

$$\text{Antennenlänge} = \frac{\lambda}{x}$$

wobei x eine ganze Zahl (4) ist

selber Antennen in geeigneter Länge (17,32cm) hergestellt, die die gewünschte Verbesserung erbrachten.

Da die Daten wieder autonom zu dem Copter gesendet werden sollten, wurden verschiedene Lösungsansätze dafür entwickelt.

Erste Ideen, Mission Planner, eine Open-Source Software umzuschreiben, sodass die Befehle im Programm gesendet werden, wurden nach einiger Recherche verworfen, da der Source-Code für uns als Laien zu komplex ist.

Die zweite Idee war, die Tastatur- und Mauseaktionen zu automatisieren. Dafür eignet sich die Library xdotool (Q7), welche für solche Aufgaben konzipiert ist. Jedoch stellte sich nach einigen Tests heraus, dass Pop-ups und die Unzuverlässigkeit von diversen Programmen eine solche Benutzung nicht erlauben.

Die nächste Idee war es, die Steuerung mit DroneKit (Q8), einer Python Library laufen zu lassen. Diese wandelt Python-Befehle in MAVLink (Micro Air Vehicle Link) um, ein Protokoll für die Kommunikation mit kleinen unbemannten Fahrzeugen. Diese werden per Telemetrie-Modul übertragen.

2.7 Entwicklung der Steuerung des Copters

Um den Copter über dem Ball zu zentrieren, wurden von uns zwei Ideen aufgenommen:

2.7.1 Steuerung bei gleichbleibender Höhe und Ausrichtung

Das Kamera Bild verfügt über ein gleichbleibendes Sichtfeld. Mithilfe dieser Maße und dem Bildpunkt kann nun die Flugbahn vom Copter vorherbestimmt werden.

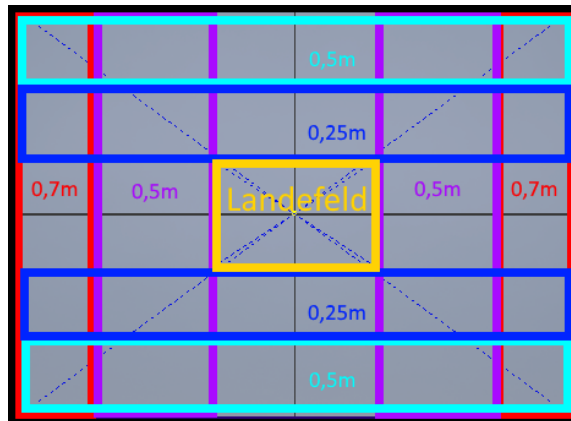


Abbildung 9, Skizze Sichtfeld

Befindet sich der Bildpunkt im sogenannten Landefeld (Abbildung 9) werden keine weiteren Änderungen vorgenommen und der Copter beginnt auf dem Ball zu landen.

2.7.2 Steuerung bei wechselnder Höhe und Ausrichtung

Um Bewegungen und Ungenauigkeiten des Copters in Bezug auf Ausrichtung und Neigung auszugleichen, wurde ein Regelkreis entwickelt, welcher Bewegungen in Echtzeit zur Berechnung der Fluggeschwindigkeit mit einbezieht. Das folgende Diagramm (Abbildung 10) stellt die Berechnung der Fluggeschwindigkeit in Nord-Süd und West-Ost Richtung dar. Die übergebenen Variablen ballX und ballY sind die Pixelkoordinaten des Ballmittelpunkts im Bild.

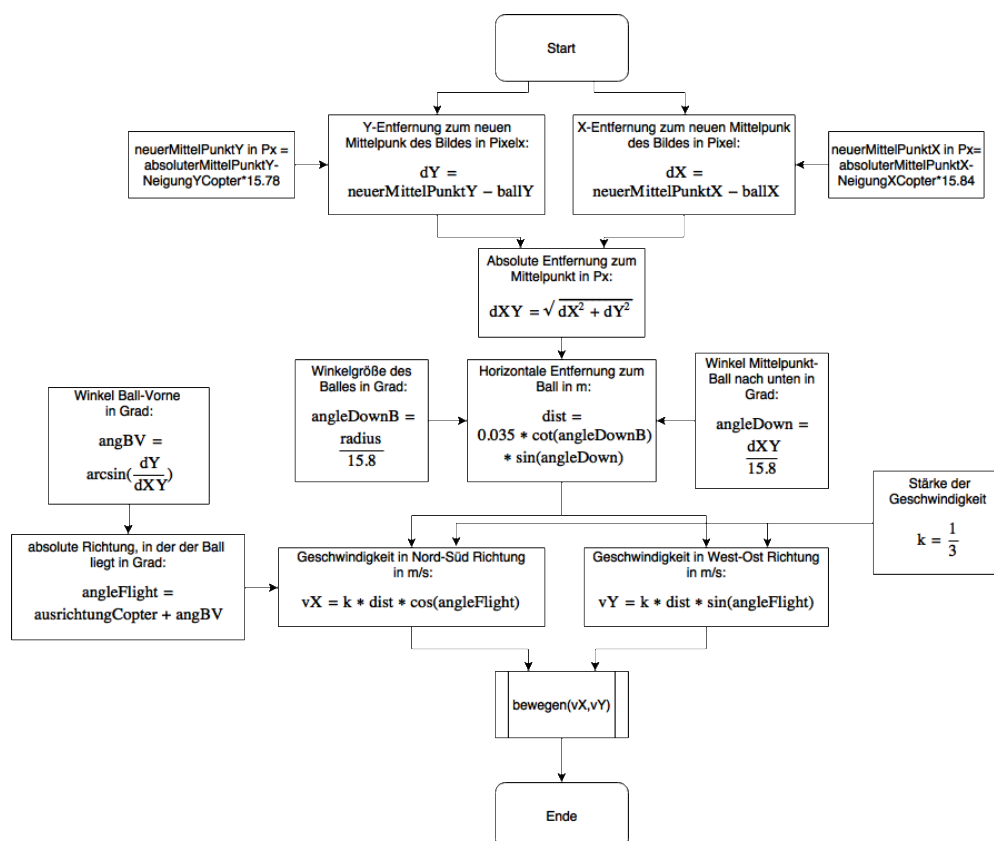


Abbildung 10, Berechnung der Fluggeschwindigkeit, schematisch

Letztendlich wird wohl eine Kombination aus beiden Möglichkeiten genutzt, da die Erste keine Ausrichtung berücksichtigt und die Zweite von einer unproblematischen Bilderkennung (Größe des Balles) abhängt, welche sich einige Male als nicht ganz zuverlässig herausgestellt hat.

2.8 Zweite Copter-Plattform, Bau des Hexacopters

Aufgrund der entstehenden Verluste bei der Übertragung des Bildes vom Copter auf den Desktop-PC am Boden musste das System zur Bilderkennung auf das Fluggerät verlagert werden. Eine Installation eines Computers mit ausreichender Rechenleistung auf dem Copter ist nicht möglich gewesen, da das Gewicht der Elektronik außerhalb der Zuladungsgrenze liegt.

Mögliche Lösungen:

- Anpassung der Software
- Bau eines größeren Copters

Da die Entwicklung des Tennisballerkennungsprogramms schon fast abgeschlossen war, entschieden wir uns für den Bau eines größeren Copters, der mit einer größeren Trageleistung auch Fußbälle aufnehmen soll.

Dazu verwendeten wir als leistungsstarken Computer für die Bilderkennung auf dem Copter den XPC Nano Barebone NC01U7, welcher nur 0.36 kg wiegt (Q4). Zudem wurde uns freundlicherweise eine USB Industriekamera von dem Hersteller IDS zur Verfügung gestellt. Aufgrund der USB 3 Schnittstelle besteht nur eine sehr geringe Verzögerung bei der Bildübertragung. Als Flugsteuerung dient ein Pixhawk 2, welcher einer Weiterentwicklung des älteren APMs entspricht.

2.9 Entwicklung des Ballaufnehmers

Im Folgenden wird die Entwicklung des Ballaufnehmers beschrieben. Dies ist ein bedeutender Aspekt dieser Arbeit, da einige Faktoren berücksichtigt werden mussten:

- Gewicht
- Größe der Aufnahmefläche
- Zuverlässigkeit

Der erste Prototyp wurde aus leichtem Kunststoffmaterial mithilfe eines 3D Druckers hergestellt und besteht aus einem Kreis der nur minimal größer als ein Tennisball ist. Kleine Widerhaken, ebenfalls aus dem 3D-Drucker, halten den Ball fest an seiner Position (siehe Abbildung 11).

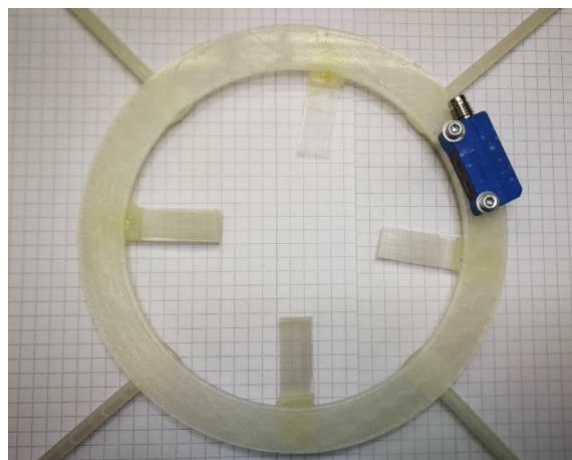


Abbildung 11, Ballaufnehmer, 01.01.2018, Foto: Marc Panse

Beim Probeflug stellte sich heraus, dass der Durchmesser des Kreises einen entscheidenden Nachteil mit sich bringt. Dieser ist zu klein, um den Copter präzise auf dem Ball zu landen. Zudem erschwert der Bodeneffekt eine genaue Landung.

Der Lösungsansatz für die Weiterentwicklung war der Bau eines größeren Ballaufnehmers. Bei der Entwicklung des zweiten Ballgreifers wurde der Durchmesser auf ein Maximum vergrößert. Die Zuverlässigkeit verschlechterte sich dadurch nur unwesentlich. Vor dem finalen 3D Druck wurde dieser aus kostengünstigem Karton erstellt. Abbildung 12 zeigt einen Prototyp für Fußbälle, gleiche Funktionsweise.



Abbildung 12, Prototyp eines größeren Ballaufnehmers, Foto: Marc Panse, 12.01.2018

Am Tag der offenen Tür im SFZ Eningen am 7. Oktober 2017 stellten wir unser Projekt vor und konnten wertvolle Kontakte knüpfen. Dabei erfuhren wir, wie im Tennisverein mithilfe eines mit Schnüren bespannten Abflussrohres, Tennisbälle aufgesammelt werden. Dieser Idee nahmen wir als Impuls für den Ballaufnehmer auf.

Dazu spannten wir Nylonschnüre parallel in einen Holzrahmen. Es wurde darauf geachtet, dass die Spannkraft auf die Schnüre während aller Versuche konstant bleibt, da eine zu große Spannkraft zur Zerstörung des Holzrahmens beziehungsweise eine zu kleine Spannkraft zum Verrutschen der Schnüre führt. Die Schnüre sind so flexibel, dass sie den Ball bei der Landung des Copters erst einmal hindurch lassen. Hebt der Copter wieder ab, so halten die Nylonschnüre so fest zusammen, dass der Ball nicht hindurchfallen kann und wie in einem Netz von dem Copter transportiert werden kann. Abbildung 13 zeigt unseren Prototyp.

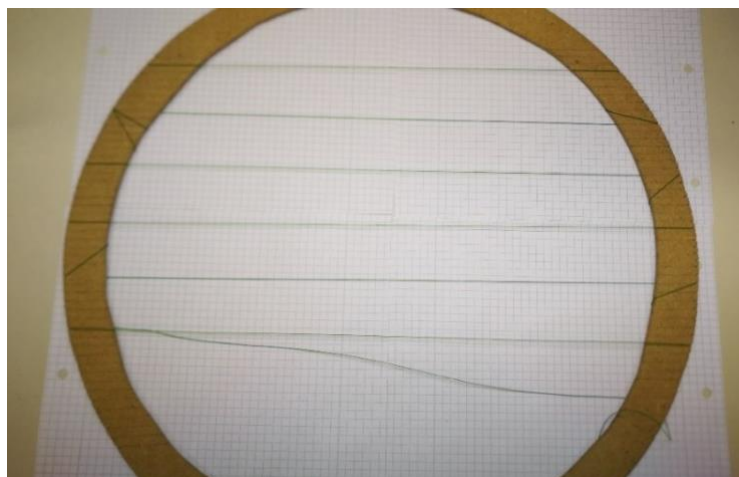


Abbildung 13, Neues Modell eines Ballaufnehmers mit Schnüren, Foto: Marc Panse, 12.01.2018

Dabei ist der Abstand der Nylonschnüre von großer Bedeutung. Dieser wurde mithilfe einer Versuchsreihe ermittelt. In dieser Versuchsreihe wurde die Zuverlässigkeit des Ballaufnehmers mit verschiedenen Abständen der Nylonschnüre zwischen $d = 1\text{cm}$ bis $d = 4\text{cm}$ verglichen. Der Versuch wurde 15 Mal pro Abstand durchgeführt.

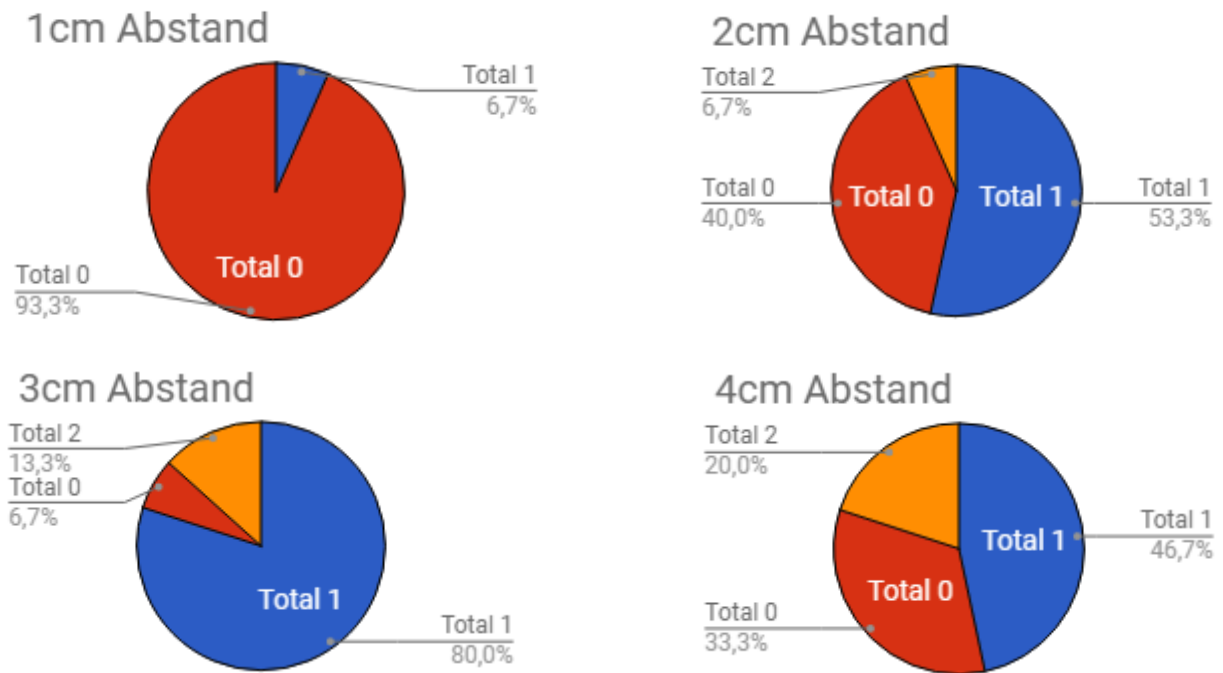


Abbildung 14, Versuch zur experimentellen Bestimmung des Abstandes der Schnüre, 30.12.2017

Total 0: Ball fällt raus oder wird nicht aufgenommen
 Total 1: erfolgreiche Aufnahme: Ball bleibt stecken
 Total 2: Fehler: Ball befindet sich unter der Schnur, kann aber nicht aufgenommen werden.

Es stellte sich heraus, dass ein Abstand der Schnüre von 3 cm die ideale Lösung darstellt.

3. Versuch zur Bestimmung der maximalen Steiggeschwindigkeit

Des Weiteren wurde ein Versuch zur Bestimmung der Flugzeit durchgeführt. Hierfür wurde ein Leistungsprüfstand für einen Motor entwickelt (siehe Abbildung 16). Für den Versuch dient einer der sechs Hexacopter-Motoren, welcher vertikal im Raum ausgerichtet wird. Ein Winkel, dessen Scheitelpunkt beweglich mithilfe eines Rohres befestigt wurde, bildet die Kraftübertragung auf eine horizontal ausgerichtete Waage. Die angezeigte Masse entspricht nach Umrechnen der Schubkraft des Motors, da beide Seiten des Winkels gleich lang sind. Während des Versuches wurde die Schubkraft in Abhängigkeit zur Spannung V am Eingang des "Elektronic Speed Controller[s]" gemessen. Eine Erhöhung der Spannung von 18 Volt auf 25 Volt wurde in 0,5 Volt Schritten vollzogen. Zuletzt haben wir die Werte für einen vollen 6S Akku von 25.2V (Hexacopter) ermittelt.

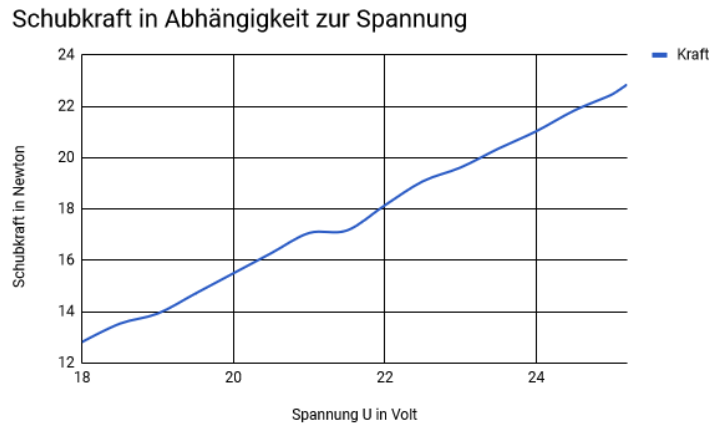


Abbildung 15, Schubkraft in Abhängigkeit zur Spannung

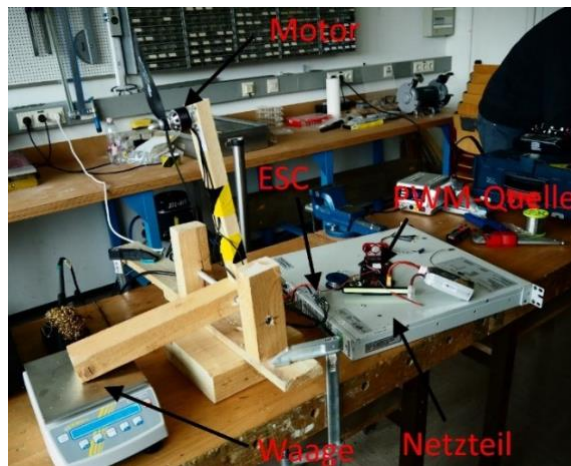


Abbildung 16, Leistungsprüfstand für Motor, 30.12.2017, Foto: Hannes Reichle

Die Beschleunigung a wird mithilfe folgender Formel errechnet:

$$\frac{F}{m} = a$$

Die Spannung des Akkus schwankt zwischen 18Volt und 25,2Volt.

Beschleunigung bei 18 Volt:

$$\frac{6 * 12,82167 \text{Newton}}{4,680 \text{kg}} = 16,438 \text{m/s}^2$$

Beschleunigung effektiv:

$$a_{eff} = 16,438 \frac{\text{m}}{\text{s}^2} - 9,81 \frac{\text{m}}{\text{s}^2} = 6,628 \frac{\text{m}}{\text{s}^2}$$

Bei einem vollen Akku weist der Copter eine deutlich größere Steigbeschleunigung von 19,49 m/s² auf. Die ermittelte Steigbeschleunigung bestätigt daher auch, dass für Extrembedingungen (leerer Akku) genügend Schub zur Verfügung steht.

4. Versuch zur Bestimmung der Effizienz

Die bestmögliche Effizienz spiegelt sich durch einen Tiefpunkt in der Leistungskurve wieder. Der Prüfstand aus dem vorherigen Versuch eignet sich hervorragend für dieses Vorhaben. Lediglich wird die Versuchsdurchführung abgewandelt.

Die Schubkraft des Motors wird konstant auf einem Wert gehalten, in diesem Fall 7.652 Newton. Dieses Gewicht muss jeder Motor im Schwebeflug des Hexacopters hochheben.

Die Leistung P setzt sich zusammen aus dem Produkt der Spannung U und der Stromstärke I . Die Spannung wird von 18 Volt auf 25 Volt erhöht. Dazu notieren wir die Änderungen der Stromstärke. Es ergibt sich folgender Graph.

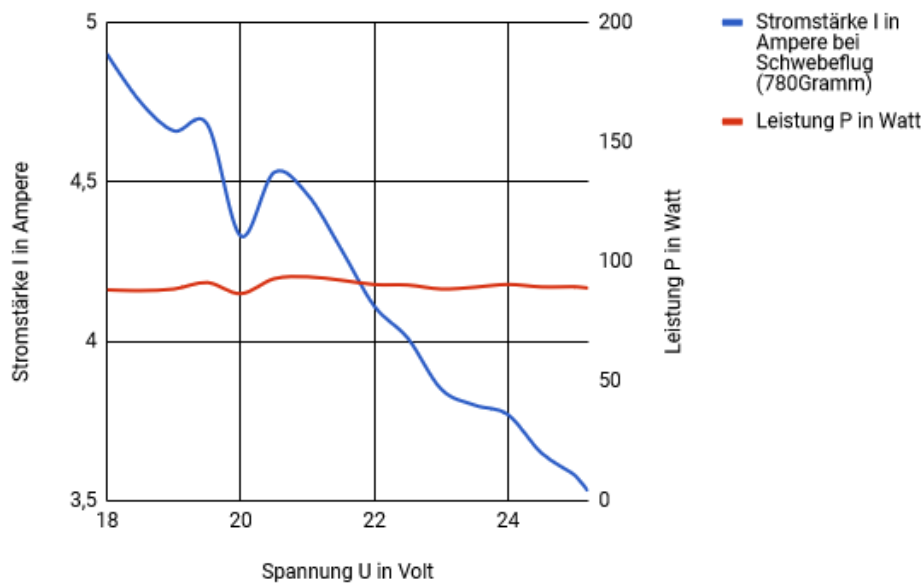


Abbildung 17, Leistungsdiagramm

Wie man sieht, handelt es sich bei der Effizienz um eine Gerade. Der Einbruch bei 20 Volt liegt nicht im Bereich eines Messfehlers, sondern war reproduzierbar. Dies liegt wohl am Regler.

Der Mittelwert der Leistung P beträgt 90 Watt pro Motor. Da unser Hexacopter über sechs solcher Motoren verfügt ergibt sich eine Gesamtleistung von 540 Watt im Schwebeflug.

Der Akku vom Hexacopter mit einer Nennspannung von 22,2 Volt und einer Kapazität von 10 Ampere-stunden würde demnach für 20 Minuten Flugzeit ausreichen.

5. Ergebnisdiskussion

Unser autonom fliegender Apportiercopter könnte in naher Zukunft die Arbeit von Hausmeistern erleichtern. Diese klettern nicht selten einmal pro Woche auf Turnhallendächer etc., um diese von Fußbällen und Tennisbällen zu befreien.

Bisher beruht die Bilderkennung jedoch nur auf Unterschieden hinsichtlich der Farbe. Dies stellt ein Problem dar, wenn zum Beispiel Fußbälle erkannt werden sollen. Hier wäre wahrscheinlich eine Bilderkennung, die Kreise erkennt, besser. Jedoch müsste dazu noch das Kamerabild entzerrt werden, da unser 4mm-Objektiv der Kamera die Objekte am Bildrand um bis zu 4% staucht.

Außerdem kann man überlegen, ob durch die Nutzung von kleineren CPUs/GPUs wie zum Beispiel dem Nvidia Jetson Tx Modul die ganze Lösung mit einem kleinerem Copter durchgeführt werden könnte.

Eine bessere Lösung, die schon einmal aufgegriffen wurde, jedoch aufgrund ihrer Komplexität wieder fallen gelassen wurde, sind sogenannte neuronale Netze. Solche Netze sind keine Seltenheit in unserem heutigen modernen Leben. Schon 1943 gelang es Warren McCulloch und Walter Pitts Verknüpfungen von elementaren Einheiten zu beschreiben als eine Vernetzung von Neuronen ähnliche Art von Netzwerk, mit dem sich praktisch jede logische oder arithmetische Funktion berechnen lassen könnte. (Q5)

Damit wäre es theoretisch möglich, hellgelbe Flächen von Tennisbällen zu unterscheiden und Vieles mehr. Zudem könnte der Copter bei jedem Flug Bilder aufnehmen, die seine „Intelligenz“ stetig steigern.

TensorFlow, eine offene Library für maschinelles Lernen, wurde ursprünglich vom Google-Brain-Team für den Google-internen Bedarf entwickelt und später unter der Apache-2.0-Open-Source-Lizenz veröffentlicht (Q6).

Mithilfe von TensorFlow, wurde ein Netz angelernt, dass Bilder auf das Vorhandensein von Tennisbällen absuchen konnte. Für diesen Lernvorgang, der circa eine Woche in Anspruch nahm, wurden 3 000 Bilder von Tennisbällen aus dem Internet heruntergeladen, beziehungsweise selbst erstellt.

Unser Netz konnte die Wahrscheinlichkeit für einen Ball im Bild ausgeben. Es stellte sich aber heraus, dass die Positionsbestimmung sehr komplex wird, weshalb die Idee erst einmal auf Eis gelegt wurde.

Unser gesetztes Ziel eines autonom fliegenden Quadrocopters mit Apportierfunktion haben wir theoretisch erreicht. Jedoch erfolgte die Umstellung auf den Hexacopter erst kurz vor Abgabe dieser Langfassung, weshalb die Software zu diesem Zeitpunkt noch nicht getestet wurde.

Ob dadurch jedoch wirklich Einsparungen zu erwarten sind, bleibt fragwürdig. In Deutschland ist das Fliegen eines autonom unbemannten Flugobjektes nur gestattet, wenn eine Kontrollperson im ständigen Sichtkontakt mit dem Copter steht und im Notfall die Steuerung übernehmen kann.

6. Zusammenfassung

Der Bau eines autonom fliegenden Apportiercopters stellt sich als höchst komplex dar. Viele Parameter, wie Rechenkraft, Zuverlässigkeit etc., die im ersten Moment nicht ersichtlich sind, müssen berücksichtigt werden.

Bisher gelang es uns, durch Lieferverzug von diversen Teilen für die Stromversorgung des Computers, leider noch nicht die Software in Verbindung mit dem Copter zu testen. Deshalb steht die Verlinkung der einzelnen Aspekte noch in Arbeit, jedoch ist zu erwarten, dass diese in naher Zukunft erfolgt.

Zudem steht der Copter dann in einer Phase der Optimierung. Eine Verbesserung des Gewichts sowie der Bilderkennung, welche bisher nur bis etwa 3 Meter Abstand zuverlässig funktioniert, sind die nächsten Entwicklungsschritte.

Am Anfang unserer Forschungsarbeit haben wir es uns als Ziel gesetzt, Bälle aus schlecht erreichbarem Gelände zu bergen. Mit unserem zweitem, neu überdachten Copter (Hexacopter), der über eine ausreichende Rechenleistung verfügt und keine Probleme mit Verbindungen zum Boden aufweist, ist es nun theoretisch möglich, „verloren gegangene“ Bälle zu bergen.

Zudem ermöglicht ein Copter, welcher eine On-Board-Bildverarbeitung hat und autonom agieren kann, noch andere weitreichende Einsatzgebiete. Ein Beispiel wäre hier die Kanadische Goldrute, welche unter anderem in schwer erreichbaren Gebieten wie Mooren ein invasiver Neophyt ist. Mit unserem Copter könnte man in der Blütezeit die hellgelbe Blüte der Pflanze zerstören, was einer Ausbreitung entgegenwirkt. Weitere Einsätze, wie das Aufspüren vermisster Personen mithilfe einer KI, könnten in Zukunft auch möglich sein.

7. Quellen- und Literaturverzeichnis

- Q1: https://www.ted.com/talks/raffaello_d_andrea_the_astounding_athletic_power_of_quadcopters, 31.12.2017 20:30
- Q2: <http://ardupilot.org/copter/docs/precision-landing-with-irlock.html>, 10.1.2018 18:04, Open-Source-Dokument zur Benutzung von Beacons
- Q3: <https://www.thingiverse.com/thing:772440>, 13.01.2018, 15:39, Thingiverse, Download Standfüße für F450
- Q4: http://www.shuttle.eu/fileadmin/resources/download/docs/spec/barebones/NC01U7_d.pdf, 14.10.2017 20:34, Shuttle, Datenblatt des On-Board-PCs
- Q5: https://de.wikipedia.org/wiki/K%C3%BCnstliches_neuronales_NetzWikipedia, 31.12.2017 22:15, Wikipedia, Geschichte von neuronalen Netzen
- Q6: <https://de.wikipedia.org/wiki/TensorFlow>, 01.01.2018, 3:30, Wikipedia, Funktionsweise und Geschichte von TensorFlow,
- Q7: <https://wiki.ubuntuusers.de/xdotool/>, 06.01.2018, 20:20, ubuntuusers, Installation und Dokumentation xdotool
- Q8: <http://python.dronekit.io/>, 30.12.2017 21:40, 3D Robotics Inc., Download DroneKit und Documentation.
- <http://ardupilot.org/ardupilot/index.html>, 19.12.2017 14:30, ArduPilot Dev Team, Download von Mission Planner, APM Planner und diverse Documentations
- <https://de.ids-imaging.com/home.html>, 15.12.2017 20:15, IDS Development Systems GmbH, Downloads des IDS Camera Managers und Dokumentation zur Industriekamera
- <https://www.tensorflow.org/>, 01.01.2018 17:15, TensorFlow, Google Brain Team, Documentation TensorFlow
- <https://www.pyimagesearch.com/2015/09/14/ball-tracking-with-opencv/>, 06.01.2018 22:12, pyimagesearch, Beispielprogramm Farbfiler unter OpenCV
- Willemer, Arnold: Python Der Sprachkurs für Einsteiger und Individualisten, 2015 Weinheim

8. Unterstützungsleistung

Ein großes Dankeschön an alle, die uns während der Arbeit unterstützt haben.