

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 285

Drohnen als Tangible User Interface zur Veranschaulichung von Experimenten

Alexander Achberger

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Albrecht Schmidt
Betreuer/in:	Pascal Knierim, M.Sc., Dipl.-Inf. Markus Funk
Beginn am:	25. November 2015
Beendet am:	25. Mai 2016
CR-Nummer:	H.5.2

Kurzfassung

Tangible User Interfaces (TUI) bieten dem Benutzer ein greifbares physikalisches Objekt, um mit einem System zu interagieren. Viele TUIs lassen sich lediglich auf einer Ebene bewegen und sind passiv. Das heißt, sie können nur vom Benutzer aus bewegt und nicht aktuiert werden. Daher ist diese Art von TUI eingeschränkt, wenn sie dem Benutzer Feedback geben sollen. Vor dem Hintergrund dieser Nachteile, erscheint es als eine gute und innovative Forschungsfrage, Drohnen, insbesondere Quadrocopter, als TUI zu nutzen, da sie sich aktiv bewegen können und somit dem Benutzer auch als Ausgabegerät dienen können. Zusätzlich lassen sie sich frei im Raum positionieren.

Diese Bachelorarbeit setzt sich mit dem Thema Drohnen als Tangible User Interface auseinander. Dabei wurden mehrere Interaktionsmöglichkeiten entwickelt, die die Drohne sowohl als Input-, als auch als Output-Gerät nutzten. Die Systemevaluierung hat ergeben, dass die Drohne eine Position schnell erreichen und anschließend sehr genau halten kann. Dadurch konnten Funktionalitäten mit der Drohne implementiert werden, die es Benutzern ermöglichen, durch die Drohne Daten zu manipulieren. In einer Studie wurde untersucht, wie Benutzer mit dem System interagieren. Dabei wurde herausgefunden, dass die Interaktionen mit der Drohne als sehr angenehm und in vielen Fällen sogar als spielerisch empfunden wurden. Dadurch konnte gezeigt werden, dass eine Drohne als Tangible User Interface großes Potential hat und dies in späteren Forschungsarbeiten weiter entwickelt werden sollte.

Inhaltsverzeichnis

1	Einleitung	11
2	Related Work	15
2.1	Tangible User Interfaces	15
2.2	Drohnen	19
3	System-Architektur	23
4	Hintergrund	25
4.1	Drohnen	25
4.2	Regelung	28
4.3	Setup	29
5	Prototyp Entwicklung	33
5.1	Software-Architektur	33
5.2	Implementierte Funktionalität	35
5.3	Proof-of-concept	39
6	Evaluation	47
6.1	Studienaufbau	47
6.2	Ergebnisse	49
6.3	Zusammenfassung und Diskussion	61
7	Zusammenfassung und Ausblick	63
7.1	Zusammenfassung	63
7.2	Herausforderungen	63
7.3	Ausblick	64

Abbildungsverzeichnis

2.1	Der reacTable.	18
2.2	Der shape display.	19
2.3	Ein Benutzer der mit dem System BitDrones interagiert.	22
4.1	Dieses Bild zeigt die Bauweise eines Quadrocopters, Quelle: [20].	26
4.2	Bild der benutzten Drohne, Eachine H8 Mini RC.	27
4.3	Hier sieht man die Konstruktion des Käfigs der Drohne.	28
4.4	Dieses Bild zeigt die theoretische Benutzung von Reglern. Quelle: [21]	29
4.5	Diese Abbild zeigt ein Beispiel einer geregelten Größe [22].	30
4.6	Dieses Bild zeigt die verwendeten Optitrack-Kameras.	31
4.7	Dieses Bild zeigt die Verbindung zwischen dem Computer und der Fernbedienung über einen Arduino UNO. Links die Fernbedienung, in der Mitte der Arduino UNO, rechts ein Steckboard.	32
5.1	Klassendiagramm des implementierten Systems.	34
5.2	Die graphische Oberfläche des Systems.	35
5.3	Diese Abbildung zeigt die aufgenommenen Höhenkoordinaten während eines Fluges.	37
5.4	Flugbahn eines Testfluges, in der die Drohne an einer festen Position hovern sollte.	40
5.5	Abweichungen pro Flug.	41
5.6	Dieses Schaubild zeigt sowohl die insgesamt berechnete Abweichung als auch die aller Achsen.	42
5.7	Flugbahn zwischen zwei Punkten in Abhängigkeit von der Zeit.	43
5.8	Flugbahn in einem Viereck.	44
5.9	Flugbahn auf einer Zickzack-Route.	45
6.1	Der häufigste Drohnen Griff.	50
6.2	Von den Probanden angegebene Deutlichkeit der Impulse auf einer 7-point Likert Scale.	52
6.3	Deutlichkeit des zweiten Output-Modus auf einer 7-point Likert Scale und deren Standardabweichungen.	54
6.4	Die bevorzugte Impulsrichtung.	55
6.5	Proband der die Drohne nach vorne drückt.	56
6.6	Ein Proband setzt die Drohne an der Position oben-rechts ab.	58

6.7	Distanz der Positionen und deren Standardabweichung.	59
6.8	Akzeptanz der verschiedenen Positionen.	59

Tabellenverzeichnis

5.1	Minimale und maximale Abweichungen beim Hovern.	42
6.1	Durchschnittlichen Verzögerungszeiten und Fehler im ersten Input-Modus. . .	51
6.2	Distanz in Meter zwischen dem Impulsradius und an der Stelle, an der Proband den Impuls gespürt hat und die Standardabweichung (STD).	53
6.3	Wohlbefinden der Probanden, die Drohne in die entsprechende Richtung zu drücken, auf einer 7-point Likert Scale und deren Standardabweichungen. . .	56

1 Einleitung

Digitale Daten repräsentieren oft Modelle oder Eigenschaften der natürlichen Umwelt sind aber dennoch nur Daten, die eingekapselt in einem Rechner sind. Sie lassen sich nicht anfassen und können dem Benutzer kein haptisches Feedback geben. Durch virtual reality wird jedoch ermöglicht in eine fremde digitale Welt einzutauchen, auch wenn man physisch in der realen Welt ist. 1993 behaupteten Wellner et. al. im Artikel "Back to the Real World" [1], dass die Menschen von der virtuellen Realität und der natürlichen Umgebung entfremden. Dies ergab den Wunsch, digitale Daten mit der realen Umwelt zu verbinden [2]. Um dies zu erreichen, müssen digitale Informationen so mit physischen Objekten gekoppelt werden, dass die Daten greifbar sind und durch Bewegungen in der realen Welt manipulierbar werden. Somit entstand der Wunsch nach Tangible User Interfaces.

Die Erfindung der Marble Answering Machine von Durrell Bishop gab eine große Inspiration für die Erforschung und Entwicklung von Tangible User Interfaces [3]. Dabei handelt es sich um einen Anrufbeantworter, der nach einer empfangenen Sprachnachricht eine kleine Kugel ausgibt. Der Benutzer kann diese Kugel entweder in eine Einkerbung der Maschine legen, damit die Sprachnachricht abgespielt wird, oder auf das Telefon ablegen, um automatisch einen Rückruf zu starten. Dies ist ein gutes Beispiel digitale Informationen greifbar zu machen, indem die Daten mit physikalischen Objekten verbunden werden. Mit dieser Idee beschäftigte sich auch Ishii et. al. in deren Arbeit Tangible Bits, mit dem Ziel, die Lücke zwischen virtuellen Räumen und der physikalischen Umwelt zu überbrücken, indem man digitale Informationen greifbar macht. Dabei hat man drei Hauptkonzepte, interaktive Oberflächen, Verbindung zwischen Bits und Atomen und umgebende Medien [3]. Das besondere an dieser Arbeit ist, dass sie digitale Informationen mit Hintergrundeigenschaften wie zum Beispiel, Licht, Wind oder Wasserbewegung verbindet. Wird beispielsweise bei jedem Besuch einer Webseite ein Tropfen in ein Wasserbecken gegeben, so nimmt der umgebende Benutzer im Unterbewusstsein wahr, wie die Besucherzahl der Webseite ist. Bei konstanten Tropfen wird der Benutzer nicht bewusst an die Webseite denken, doch hört das Tropfen auf, so wird der Benutzer dies mit der Zeit bemerken und nach eventuellen Serverproblemen suchen. Diese Verbindung zwischen realen Hintergrundinformationen und digitalen Informationen wird in dieser Arbeit erforscht [3].

All diese Ansätze und Arbeiten brachten die Forschung der Tangible User Interfaces voran. Es entstanden sogenannte "2D" und "2.5D" Tangible User Interfaces. Der Unterschied hierbei ist, dass 2D Tangible User Interfaces auf zwei Dimensionen beschränkt sind und deshalb nur auf einer Ebene funktionieren. Daher sind die Möglichkeiten der Interaktionen begrenzt. 2.5D

Tangible User Interfaces bieten dem Benutzer die Möglichkeit, mit den physikalischen Objekten nicht nur auf einer Ebene zu interagieren, sondern auch, mit einer räumlichen Beschränkung, in die Höhe. Dies wurde zum Beispiel von Follmer et. al. in ihrem System inForm erreicht [4]. Dabei wird ein shape display benutzt, der dem Benutzer ermöglicht, mit physischen Objekten in drei Dimensionen zu interagieren. Jedoch sind Interaktionen in die Höhe nach einer bestimmten Distanz nicht mehr möglich. Deshalb entstand die Forschungsfrage, wie man Tangible User Interfaces im gesamten dreidimensionalen Raum benutzen kann.

In dieser Bachelorarbeit wird diese Forschungsfrage mithilfe einer Drohne beantwortet. Stellt man sich folgendes Szenario vor: Man sitzt Zuhause und will seinen Fernseher steuern aber findet seine Fernbedienung nicht. Somit könnte man ein Signal senden, wie zum Beispiel mit dem Finger schnipsen, und eine Drohne kommt vor den Benutzer angeflogen. Nun kann man durch antippen der Drohne die Kanäle des Fernsehers wechseln und durch manuelle Verschiebung der Drohne die Lautstärke regeln. Nach Erreichen der maximalen Lautstärke soll die Drohne dem Benutzer einen Impuls geben, damit dieser versteht, dass es nun maximal laut ist. In diesem Szenario ist die Drohne das Tangible User Interface, das mit den digitalen Informationen verbunden ist. In diesem Fall dient die Drohne nicht nur als Input Gerät, sondern auch als Output Gerät indem sie dem Benutzer Informationen über haptische Feedbacks übermittelt.

Mittlerweile gibt es eine Menge verschiedener Drohnen, unter anderen kleine Quadrocopter, die klein genug sind um mit der Hand umgriffen zu werden. Doch die Drohnen müssen auch gefahrlos betätigt werden können und dem Benutzer als sympathisch erscheinen. Dies wird von Nowacka et. al. untersucht, die in ihrer Arbeit zeigen, dass Menschen eine große Akzeptanz gegenüber autonomen Benutzerschnittstellen haben [5]. Hierbei interagierten mehrere Probanden mit zwei autonom fliegenden Helium Ballons. Die Studie zeigte, dass sich die Probanden das autonome Interface als eine Art Haustier vorstellen. Das Interface hat einige Charakteristika die Tieren, insbesondere Haustieren ähneln. Beispielsweise benehmen sich beide Seiten oft doppeldeutig, man kann nicht richtig mit ihnen reden, ab und zu brauchen sie die Hilfe vom Menschen und beide machen Fehler. Dadurch wird die Technologie als sympathischer eingestuft, da Fehler machen auch zu unserer menschlichen Fehlbarkeit gehört [6]. Dies ist ein entscheidender Punkt, damit Benutzer auch mit einer Drohne als Tangible User Interface eine gewisse Sympathie entwickeln.

Wichtig ist auch die richtige und intuitive indirekte Interaktion mit Drohnen, um sie zum Beispiel an eine richtige Position zu setzen oder eine bestimmte Aktion durchführen zu lassen. Mit diesem Thema setzte sich Jessica R. Cauchard et. al. auseinander, die verschiedene Interaktionsmöglichkeiten mit einer Drohne präsentiert [7]. Jedoch gibt es noch keine Forschungsarbeit, die davon handelt, wie man direkt mit Drohnen interagiert, indem man diese greift und bewegt oder sie einem haptische Informationen zukommen lässt.

In dieser Bachelorarbeit wird ein System vorgestellt, dass Drohnen als Tangible User Interface benutzt. Dabei werden mehrere Modi implementiert, die die Drohne nicht nur als Input Gerät nutzt, sondern dem Benutzer durch Impulse auch haptisches Feedback geben kann. Im Ganzen

wird in dieser Bachelorarbeit erforscht, wie Menschen am besten mit Drohnen interagieren können und wie die Akzeptanz gegenüber dem System ist.

Gliederung

Zu Beginn der Arbeit, wird sich mit themenverwandten Arbeiten auseinandergesetzt. Daraufhin werden alle benötigten Anforderungen des implementierten Systems und die dazu benötigte Hardware beschrieben. Anschließend werden die theoretischen und technischen Grundlagen näher erläutert, die zum Entwickeln des Systems notwendig sind. Nach einer Beschreibung der Software Architektur werden die implementierten Funktionalitäten des Systems vorgestellt und anschließend eine Systemevaluierung präsentiert, die die Stabilität und Genauigkeit des Systems beschreibt. Daraufhin wird der Aufbau der durchgeführten Studie und deren Ergebnisse vorgestellt und näher erläutert. Abschließend werden die Resultate und alle Problematiken der Arbeit zusammengefasst und Anknüpfungspunkte vorgestellt.

2 Related Work

Um Drohnen als Tangible User Interface zu benutzen, muss zunächst geklärt werden, wie sich Tangible User Interfaces überhaupt zusammensetzen, wie man sie benutzt, was für Möglichkeiten sie bieten und welche Auswirkungen sie auf den Benutzer haben. Auf diesem Gebiet gibt es schon viele Forschungsarbeiten, die in diesem Kapitel vorgestellt und näher erläutert werden. Zudem muss auch die Schnittstelle zwischen Drohne und Tangible User Interface näher betrachtet werden, wie man zum Beispiel generell mit Drohnen interagiert, wie benutzerfreundlich und praktisch Drohnen im Alltag sind und wie man autonome Systeme mit dem Menschen am sympathischsten interagieren lässt. Auch das sind Themen, die in diesem Kapitel anhand verwandter Arbeiten behandelt werden.

2.1 Tangible User Interfaces

In den letzten Jahrzehnten sind eine Menge unterschiedlicher Tangible User Interfaces entwickelt und erforscht worden. Bei der Entstehung der Idee von Tangible User Interfaces half Hiroshi Ishii mit seinen Studenten bei deren Arbeit Tangible Bits [3]. Die Idee dahinter war, Bits direkt manipulierbar und zugänglich zu machen, indem die Umwelt als Interface genutzt wird, um somit Daten mit physischen Objekten zu verbinden. Ein Beispiel hierzu ist deren daraus entstandener Prototyp, der metaDesk [8]. Hierbei geht es um einen Tisch, der zugleich ein Bildschirm ist. Nun kann man ein kleines Haus auf den Tisch stellen und um das platzierte physikalische Objekt erscheint im Bildschirm des Tisches eine geographische Karte. Durch die Interaktion mit diesem Objekt, das aus drehen oder verschieben besteht, kann auch die Karte verschoben oder rotiert werden. Somit besteht eine direkte Verbindung zwischen den Daten des Systems und dem physikalischen Objekt.

Später arbeitete Hiroshi Ishii an seiner Arbeit weiter und veröffentlichte “Tangible Bits: Beyond Pixels“, in der er unter anderem die wichtigsten Eigenschaften und Anforderungen von Tangible User Interfaces beschreibt [9]. Dort erläutert er, dass eine der größten Herausforderungen ein passendes Tangible User Design zu finden ist, wie man physische Gegenstände und deren Manipulationsmöglichkeiten auf digitale Daten abbildet, so dass sie dem Benutzer ein sinnvolles Feedback geben können. Die richtige Wahl des physischen Objekts hängt jedoch auch von der Aufgabe des Systems ab. Zudem sollten die Interaktionen mit dem Objekt simpel und gut zu verstehen sein, damit die Benutzer das System intuitiver und fehlerfreier benutzen können [9]. In Bezug auf meine Bachelorarbeit ist es also wichtig, die Interaktionsmöglichkeiten mit der Drohne auf einen sehr simplen Niveau zu halten. Somit macht es Sinn, die Drohne durch kurze

und leichte Bewegung zu steuern, um den Benutzer nicht unnötig zu überfordern. Zudem wird in der Arbeit "Tangible Bits: Beyond Pixels" beschrieben, dass es wichtig ist dem Benutzer ein Echtzeit Feedback zu geben. Das heißt dass die Zeit zwischen einer Input Eingabe und der Ausgabe minimal sein soll. Dadurch wird die Verbindung zwischen dem Tangible User Interface und den Daten noch enger. Deshalb ist es von großer Bedeutung, nach Interaktionen auf die Drohne dem Benutzer ohne große Verzögerung ein Feedback zu geben. Würde der Benutzer beispielsweise durch bewegen der Drohne die Lautstärke seines Fernsehers regeln wollen, die Änderung aber erst nach einigen Sekunden nach der Interaktion passiert, würde der Benutzer die Lautstärke nicht genau oder nur mit großer Anstrengung einstellen können.

Eine andere interessante Arbeit von Hiroshi Ishii et. al., die für meine Bachelorarbeit relevant ist, ist die "Tangible Interfaces for Remote Collaboration and Communication" [10]. Hier geht es darum entfernte Zusammenarbeit oder Kommunikation durch Tangible User Interfaces zu ermöglichen. Dies soll den Benutzern das Gefühl geben, dass sie mit gemeinsam geteilten Objekten arbeiten. Dies wurde mit zwei verschiedenen Prototypen getestet, dem psyBench und der inTouch. Der psyBench steht für Physically Synchronized Bench und bietet den Benutzern einen gemeinsam geteilten Arbeitsplatz mit Tangible User Interfaces um so eine Schnittstelle zu CSCW zu bekommen. CSCW steht für Computer Supported Collaborative Work und beschreibt die Kooperation zwischen Menschen und deren Unterstützungsmöglichkeit mit Rechnern [11]. Der psyBench besteht aus einer Platte, die aus 10×8 Felder besteht. In jedes dieser Felder kann ein physisches Objekt abgestellt oder bewegt werden. Die Objekte besitzen im Inneren einen Magnet, damit sie durch den unter der Platte befindlichen Magneten automatisch bewegt werden können, oder man die aktuelle Position der Objekte ermitteln kann. Bewegt man nun eines dieser Objekte von einem Feld zu einem anderen, so wird die Positionsänderung erfasst und zu der zweiten identischen Platte gesendet. Dort bewegt sich das Objekt selbständig zu der Position, wo es nun auf der ersten Platte steht. Somit denkt der Benutzer, dass er mit gemeinsam geteilten Objekten arbeitet. Rotierende Bewegungen mit den Objekten sind leider nicht möglich.

InTouch ist ein System, um miteinander aus größerer Entfernung haptisch zu kommunizieren. Der Prototyp besteht aus zwei identischen Objekten, die miteinander haptische Informationen austauschen können. Die Objekte bestehen jeweils aus einem Untergrund und drei darauf platzierte Rollen. Die Rollen sind jeweils mit einem Motor verbunden, um sie computergesteuert drehen lassen zu können. Dreht nun ein Benutzer die Rollen bei dem einen Gerät, so drehen sich die Rollen bei dem anderen Gerät mit der gleichen Geschwindigkeit. Wollen nun beide Benutzer jeweils an ihrem Gerät die Rollen in Bewegung setzen, so spüren sie die Interaktion des jeweils anderen Benutzers. Somit können sie auf Distanz haptisch miteinander kommunizieren. Oft können Berührungen mehr aussagen als Wörter. Es ist persönlicher, herzlicher und direkter. Deshalb ist es wichtig, dies bei einer Kommunikation nicht zu vernachlässigen, auch wenn man geographisch weiter entfernt ist.

Beide Prototypen, psyBench und InTouch könnte man auch mit Drohnen realisieren und sogar in ein einziges System integrieren. Man stelle sich zwei geographisch getrennte Arbeitsplätze vor, die beide eine gleiche Anzahl an identischen Drohnen besitzen. Nun stelle man sich jede Drohne als gemeinsam geteiltes Objekt vor. Bewegt ein Benutzer eine Drohne auf eine neue

Position, so verändert sich in Echtzeit auch die Position der Drohne am anderen Arbeitsplatz entsprechend. Ein Vorteil hierbei wäre, dass man die Drohnen im dreidimensionalen Raum bewegen und positionieren könnte. Zudem würde die Drohne nicht in ihrer Rotation eingeschränkt sein und könnte dem Benutzer dadurch weitere Informationen zukommen lassen. Wenn nun jeweils ein Benutzer die gleiche geteilte Drohne bewegt, so könnte man dem Benutzer wie beim Prototyp InTouch ein haptisches Signal zukommen lassen. Beispielsweise könnten sich so zwei Menschen in verschiedenen Räume die "Hand geben", indem sie die gleiche geteilte Drohne nehmen und diese wie eine Hand schütteln. Somit könnte man beide vorgestellten Prototypen mithilfe von Drohnen in ein einziges System integrieren.

Eine andere Arbeit, die von einer weiteren Art von User Interfaces handelt, ist die von Tanja Döring et. al. mit dem Titel "A Design Space for Ephemeral User Interfaces" [12]. Hier werden User Interface Elemente angesprochen, die jeweils nur kurzlebig sind. In der Arbeit wird der Begriff Ephemeral User Interface wie folgt definiert:

Ephemeral user interfaces are a class of user interfaces that contain at least one UI element that is intentionally created to last for a limited time only. The durability of the UI element is determined by its intrinsic material properties in combination with its surrounding ecosystem. While their ephemeral UI element(s) exist(s), ephemeral user interfaces provide a rich and multisensory user experience. They may deliberately be designed to offer only partial or imperfect user control.

Menschen mögen für gewöhnlich kurzlebige Momente und Ereignisse, wie zum Beispiel Feuerwerk oder Regenbögen. Ereignisse die nur für einen kurzen Moment vorhanden sind, werden bei den Menschen als besonders empfunden. Dieses Phänomen wird in dieser Arbeit benutzt um User Interfaces für Benutzer spannender und aufregender zu gestalten. Eine wichtige Eigenschaft der Ephemeral User Interfaces ist, dass ihre Lebensdauer zeitlich begrenzt ist. Deshalb werden sie nur für eine kurze Zeit benutzt und anschließend wieder zerstört oder neu erschaffen. Dadurch wird die Aufmerksamkeit des Benutzers durch die Interaktion mit dem User Interface erhöht und die mentale Belastung verkleinert. In der Arbeit wird zudem beschrieben, dass Benutzer eines Ephemeral User Interface mehr Interesse und Motivation haben, ein System mit einer solchen Schnittstelle zu benutzen. Zudem können sie dem Benutzer eine Menge Informationen geben, die durch die natürlichen Eigenschaften des Materials abhängen. Beispielsweise bei bestimmten Oberflächen, Temperaturen oder Materialstruktur können sie dem Benutzer jeweils eine andere Art von haptischen Informationen zukommen lassen.

In Bezug auf meine Arbeit könnte man eine Drohne als Interaktionssystem auch als Ephemeral User Interface betrachten. Die Drohne "lebt" nur so lange, wie der Akku sie mit Energie versorgt. Danach stürzt sie ab und kann nicht mehr als Interaktionsobjekt dienen. Danach muss der Akku wieder aufgeladen werden oder ein neuer Akku eingesetzt. Demnach ist die Drohne als User Interface zeitlich begrenzt und könnte den Benutzern auch die Vorteile von Ephemeral User Interfaces vermitteln.

Im weiteren Verlauf der Erforschung der Tangible User Interfaces wurde Elektro-Akustische-Musik mit einem Tangible User Interface manipuliert. Die Idee hatte Martin Kaltenbrunner et. al. mit dem reacTable [13]. Hierbei lassen sich wieder auf einem Tisch mehrere physikalische

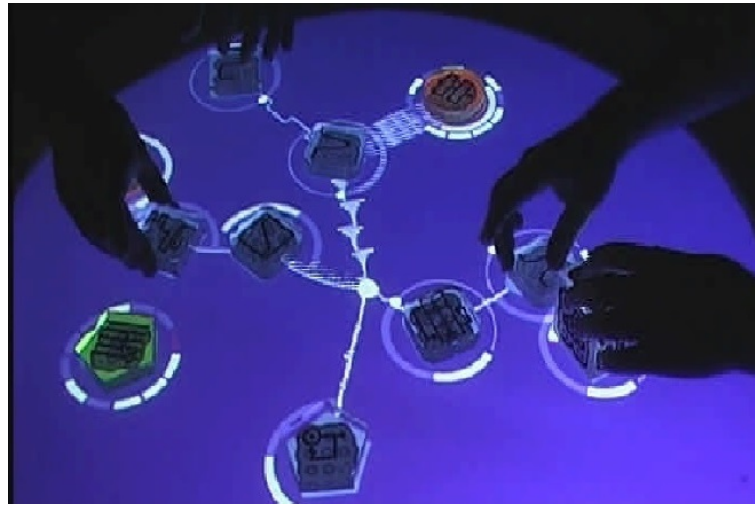


Abbildung 2.1: Der reacTable.

Objekte platzieren, siehe Abbildung 2.1. Jedoch hat jedes Objekt eine andere Auswirkung auf das System. Es gibt beispielsweise “Generator“ Objekte, die Musik oder Töne erzeugen. Zudem gibt es noch “Controllers“, die mit anderen Objekten verbunden werden können, um somit deren Audioausgänge zu manipulieren. Die Positionierung und Ausrichtung der Objekte auf dem Tisch haben Auswirkung auf die entstehende Musik. Somit wird die Musikausgabe physisch manipulierbar gemacht.

Die bisher benannten Tangible User Interfaces sind auf eine Ebene beschränkt, daher sind es 2D User Interfaces. Sie können beispielsweise keine 3D Inhalte erzeugen oder die verbundenen physischen Objekte können nicht im ganzen Raum manipuliert werden. Es gibt aber Tangible User Interfaces, die begrenzt im Raum arbeiten. Daher werden diese 2.5D User Interfaces genannt. Sie können nicht nur auf einer Ebene Daten manipulieren, sondern zu einem gewissen Grad, auch in die Höhe. Ein solches User Interface haben beispielsweise Follmer et. al. in ihrem System inForm erschaffen [4]. In dieser Arbeit geht es um shape displays, die in der Lage sind, dem Benutzer dynamische Tangible User Interface Methoden in einem 2.5D Raum zu bieten. Shape display, ist eine physikalische Fläche, die aus mehreren kleinen Feldern besteht. Jedes dieser Felder besteht aus einem rechteckigen Stab, die eine quadratische Oberfläche haben, siehe Abbildung 2.2. Diese Stäbe können, bis zu einem gewissen Grad, durch einen Computer ihre Höhe verändern und halten. Dadurch ist es möglich verschiedene Oberflächen zu erschaffen und diese dynamisch zu verändern. In der Arbeit werden verschiedene Interaktionsmöglichkeiten mit dem display vorgestellt. Zum einen lassen sich Knöpfe durch hervorstehende Stäbe repräsentieren. Dadurch ist es dem Benutzer möglich, dem System verschiedene Aktionen zu übermitteln. Zum anderen lassen sich mit dem display auch 1D- und 2D-Touchscreens erzeugen. Zusätzlich können physikalische Objekte auf den display gelegt werden, die darauf in das System als Tangible User Interface integriert werden, als Beispiel siehe Abbildung 2.2. Diese Objekte können gegriffen und bewegt werden. Durch den display kann nun die Bewegungsfreiheit des Objektes beschränkt werden, damit es sich

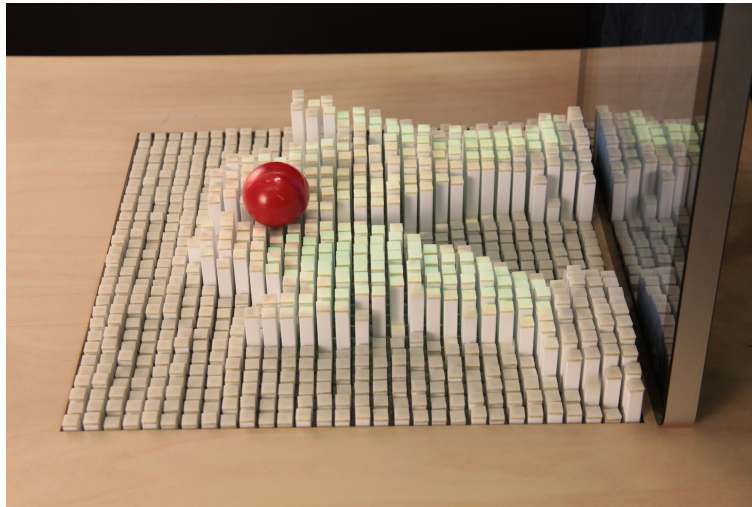


Abbildung 2.2: Der shape display.

beispielsweise nur noch auf einer Achse bewegen lässt. Zusätzlich kann durch eine Art Rampe, die Bewegung des Objekts in eine Richtung erschwert werden, um dadurch dem Benutzer ein haptisches Feedback zu geben. Der Vorteil an diesem Tangible User Interface ist, dass sich das Interface jederzeit dynamisch an der gleichen Position verändern kann. Wird beispielsweise das Objekt an eine bestimmte Stelle geschoben und dadurch eine Aktion beendet, kann der display daraufhin die Interaktionsmöglichkeiten mit dem Objekt verändern. Durch dieses System kann die Interaktionsweise mit physikalischen Objekten erweitert werden, ist jedoch in ihrer Höhe beschränkt. Daher gibt es auch 3D Tangible User Interfaces, durch die sich Daten mit physikalischen Objekten im ganzen Raum verbinden lassen. In dieser Arbeit werden Drohnen eingesetzt, um ein 3D Tangible User Interface zu erschaffen und die Nachteile der 2D und 2.5D Tangible User Interfaces zu überwinden.

2.2 Drohnen

Um ein System zu entwickeln, in der eine Drohne als Tangible User Interface fungieren soll, ist es nicht nur wichtig zu wissen, wie Tangible User Interfaces aufgebaut sind und welche Vorteile sie mit sich bringen, sondern auch, wie man Drohnen dafür einsetzt. Dazu sollte geklärt werden, wie generell autonome Systeme auf Menschen wirken, was für Konzepte es gibt mit Drohnen zu kommunizieren und in welchen Bereichen sie bisher als Hilfsmittel dienen. Diese Fragen werden in den folgenden vorgestellten Arbeiten geklärt.

Wie im vorherigen Kapitel schon angedeutet, ist ein gefahrloser Umgang mit Drohnen ein Hauptkriterium, das unbedingt erfüllt sein muss. Man darf nicht in die laufenden Rotoren fassen können und die Drohne darf keine Gegenstände durch instabiles Flugverhalten im Raum zerstören. Abgesehen davon muss die Drohne dem Benutzer als ein angenehmes und

sympathisches Interface erscheinen. Damit haben sich Diana Nowacka et. al. befasst, indem sie in einer Studie Probanden mit einem autonom fliegenden Helium Ballon interagieren ließen [5]. Die Studie bestand aus zwei Teilen. Im ersten Teil musste eine Gruppe von vier Probanden zusammen Aufgaben lösen während zwei autonom fliegende Ballons namens Diri #1 und Diri #2 im Raum umher flogen und beliebige Bilder schossen. Im zweiten Teil der Studie musste ein Team von Probanden Aufgaben lösen. Diesmal mussten sie jedoch ihre Ergebnisse von den Diris fotografieren lassen. Dabei wurde ihnen erklärt, dass Diri #1 zu einem zufälligen Zeitpunkt Bilder macht und Diri #2, wenn er ein Gesicht vor sich erkennt. Die Probanden konnten die Ballons greifen und bewegen, um somit bessere Fotos zu machen. Anschließend wurden den Probanden Fragen gestellt, wie zum Beispiel: "Wie würden sie sich fühlen ein solches Objekt in ihrem Umfeld (zu Hause oder in der Arbeit) zu benutzen?" Ein Proband der Studie verletzte seinen Finger an einem Propeller der Ballons und wurde anschließend gefragt, wer denn nun dafür verantwortlich sei. Er antwortete mit "die Person die ihn entwickelt hat". Auch andere Probanden machten die autonomen Flugobjekte nicht für ihre Handlung verantwortlich sondern empfanden es nur als eine Art Equipment. Dies ist ein interessanter Punkt für andere autonome Systeme, wie auch mein implementiertes Drohnensystem, um das Verhältnis zwischen Benutzer und Drohne besser zu verstehen.

Wichtig ist auch die indirekte Interaktion mit Drohnen, damit mit meinem System auch aus weiterer Entfernung kommuniziert werden kann. Mit diesem Thema haben sich Cauchard et. al. in ihrer Arbeit "Drone & Me: An Exploration into Natural Human-Drone Interaction" auseinandergesetzt [7]. Dabei haben sie eine Studie durchführen lassen, indem sie Probanden vor eine Drohne gestellt haben und ihnen Aufgaben gegeben, was die Drohne machen soll. Beispielweise gab es die Aufgaben "landen", "fotografiere einen Baum", "aufhören sich zu bewegen", "dem Benutzer folgen". Die Probanden hatten keine Vorgaben, wie sie der Drohne ihre Aufgabe übermitteln und sollten sich eine für sie natürliche Art auswählen. Eine Person stand hinter dem Probanden und fernsteuerte die Drohne. Der Proband sollte jedoch denken, dass die Drohne autonom ist. Bei der Studie kam heraus, dass am häufigsten Körpergesten, Sprache oder eine Kombination aus beidem benutzt wurde. Dabei wurden Körpergesten am meisten benutzt, Sprache weniger. Nachdem die Probanden vertrauter mit dem Umgang der Drohne waren, wurde jedoch mehr Sprache eingesetzt. Um die Drohne zu navigieren wurden meistens Winkbewegungen oder durchgängiges streichen benutzt. Bei der Auswertung der Studie kam heraus, dass 90% aller Probanden dachten, sie hätten die Drohne unter Kontrolle. 95% fanden es natürlich mit einer Drohne derart zu kommunizieren. Viele der Probanden interagierten mit der Drohne wie mit einem Menschen und sagten zu ihr zum Beispiel "Bitte" und "Danke". Dadurch zeigt sich, dass der Umgang mit Drohnen natürlich wirken kann und es somit durchaus Sinn machen kann, Drohnen in anderen Bereichen zu nutzen.

Ein Beispiel hierfür liefert die Arbeit von Schneegrass et. al. in deren Arbeit "Midair Displays: Exploring the Concept of Free-Floating Public Displays", die ein großes Potential darin sehen, Drohnen mit Displays auszustatten, um somit an öffentlichen Plätzen Informationen auszugeben [14]. Beispielszenarien, die hier vorgestellt werden sind unter anderem, Sportbildschirme, um zum Beispiel Kletterern die verschiedenen Schwierigkeitsstufen der entsprechenden Route zu zeigen oder Notfallbildschirme, die Menschen beispielsweise nach einem Erdbeben zu einem

sicheren Ort führen können. In der Arbeit wurde eine solcher Prototyp erstellt und verschiedenen Probanden vorgestellt. Alle Tester sahen ein großes Potential in dieser Technologie. Deshalb könnte man die Drohne in meinem System ebenfalls mit einem Bildschirm ausstatten, um somit den Benutzern noch mehr Informationen zukommen zu lassen, oder wie in der Arbeit “Displaydrone: A Flying Robot Based Interactive Display“ vorgestellt, einen Projektor an die Drohne bauen um somit Informationen an beliebige Positionen zu projizieren.

Eine andere Arbeit, die zeigt wie eine Drohne im Alltag helfen kann, ist die von Meas et. al. mit dem Namen “A Flying Pantograph: Interleaving Expressivity of Human and Machine“ [15]. Dabei geht es um ein System, die Handzeichnungen aufnimmt und in Echtzeit durch eine Drohne an eine Leinwand gezeichnet wird. Dadurch kann man automatisch die Handzeichnung skalieren oder spiegeln lassen. Zudem kann auch erst das Bild manuell gemalt werden, die Zeichnung wird aufgenommen und kann zu einem anderen Zeitpunkt gezeichnet und auch dupliziert werden. In diese Richtung orientiert sich auch die Forschungsarbeit von Harshit Agrawl et. al. in der Arbeit “L’evolved: Autonomous and Ubiquitous Utilities as Smart Agents“, in der sie verschiedene Ansätze präsentieren, wie eine Drohne im Haushalt dienen könnte [16]. Hier dient zum Beispiel eine Drohne als Lichtassistent, die zum einen den Benutzer im Dunkeln folgen kann, damit dieser etwas sehen kann und zum anderen kann sie dabei helfen Objekte zu finden, indem sie sie anleuchtet. Zudem wird die Drohne noch als fliegender Tisch benutzt, indem auf der Drohne eine glatte Oberfläche angebracht wurde auf der gearbeitet werden kann. Bewegt sich der Benutzer, der gerade auf dem Tisch arbeiten möchte, so bewegt die Drohne diesen Tisch damit der Benutzer weiter ohne Unterbrechungen arbeiten kann. Dadurch sieht man, bei wie vielen Gelegenheiten Drohnen eingesetzt werden können.

In der Arbeit “DroneNavigator: Using Drones for Navigating Visually Impaired Persons“ von Avila et. al. wird eine Drohne als Navigationshilfe für Sehbehinderte eingesetzt [17]. Dabei fliegt eine kleine Drohne in einem Abstand von 1-2m vor der sehbeeinträchtigten Person und gibt ihr den Weg zu dem gewünschten Ziel vor. Die Person weiß anhand der natürlichen Fluggeräusche die Position der Drohne und kann ihr folgen. Dies wurde in einer Studie getestet und die sehbehinderte Person hat jedes mal ihr Ziel problemlos erreicht. Da die Drohne ein Geräusch macht beim fliegen, kann sie gut geortet werden, wie die Studie bewiesen hat. Somit kann mein System eventuell auch von Sehbehinderten bedient werden.

Eine Arbeit, die von Tangible User Interfaces handelt und recht ähnlich zu meiner Bachelorarbeit ist, ist die “BitDrones: Towards Using 3D Nanocopter Displays as Interactive Self-Levitating Programmable Matter“ von Calvin Rubens et. al. [18]. Dabei geht es ebenfalls um Drohnen, die jeweils einen schwebenden Block in einen interaktiven, greifbaren 3D Bildschirm repräsentieren. Hierbei können die Drohnen auch als Input und Output benutzt werden, indem beispielsweise ein paar Drohnen einen Touchscreen tragen oder RGB LEDs benutzen, um dem Benutzer Feedback zu geben. In “BitDrones“ können die Drohnen auch genommen und verschoben werden oder wie ein Knopf gedrückt werden. Jedoch wird hierbei auch die Hand des Benutzers mit Markern versehen um diese tracken zu können. Dadurch wird die Interaktion mit den Drohnen stabiler. Allerdings muss jeder Benutzer mit Marker versehen werden, was in meiner Arbeit nicht der Fall ist. Zudem werden in BitDrones andere Use-Cases implementiert,



Abbildung 2.3: Ein Benutzer der mit dem System BitDrones interagiert.

Beispielsweise wird hier nicht auf Impulse der Drohne eingegangen wenn man sie in der Hand hält. Somit bezieht sich meine Bachelorarbeit auf die anderen Use-Cases und das interagieren mit der Drohne ohne Handtracking.

Abschließend sieht man an den vorgestellten Arbeiten, dass in den Bereichen, 2D und 2.5D Tangible User Interfaces schon viel erforscht wurde. Es wurde gezeigt, wie physikalische Objekte mit Daten verbunden werden können und diese anschließend manipuliert werden. Durch 2.5D Tangible User Interfaces lassen sich 3D Inhalte in der physikalischen Welt erzeugen. Allerdings sind die vorgestellten Tangible User Interfaces beschränkt in den Möglichkeiten, 3D Daten zu erzeugen und zu manipulieren. Sie können sich nicht frei im Raum bewegen und dem Benutzer keine dreidimensionale haptische Feedbacks geben. Deshalb wird in dieser Arbeit, eine Drohne als Tangible User Interface benutzt, um diese Nachteile zu beheben.

3 System-Architektur

In dieser Bachelorarbeit soll ein System entwickelt werden, das Drohnen als Tangible User Interfaces benutzt. In diesem System soll es möglich sein, Daten mit einer Drohne so zu verknüpfen, dass Informationen durch Interaktionen mit der Drohne manipuliert und präsentiert werden können. Dabei soll die Drohne zum einen als Output-Gerät dienen, die dem Benutzer Informationen veranschaulicht oder ein haptisches Feedback gibt. Zum anderen soll die Drohne gleichzeitig auch als Input-Gerät dienen, um Daten manipulieren zu können. Um dies zu erreichen, muss die Drohne durch ein implementiertes System gesteuert und geregelt werden. Sie muss eine Position stabil halten können und auch auf andere Positionen autonom fliegen können. Zudem muss das System erkennen, wann und wie mit der Drohne interagiert wird, indem verschiedene Interaktionsmöglichkeiten implementiert werden.

Um dieses Vorhaben umzusetzen, wird eine Drohne benötigt, die klein genug ist, um sie greifen zu können, aber dennoch stark genug ist, um dem Benutzer spürbare Impulse geben zu können. Zudem muss sie in der Lage sein, leichte Gewichte tragen zu können, damit um die Drohne eine Art "Käfig" angebracht werden kann, sodass die Drohne zum einen besser gegriffen werden kann und zum anderen die Verletzungsgefahr beim Greifen geringer ist. Dieser Kasten muss ebenfalls noch konstruiert werden. Dabei ist zu beachten, dass das Material leicht, aber dennoch stabil sein muss und die Form des Kastens die Flugaerodynamik nicht negativ beeinflussen darf. Damit das System weiß, wo die Drohne ist, wird ein Tracking-System benötigt, das die Koordinaten der Drohne im Raum ausgibt. Es sollte im Millimeterbereich arbeiten, eine große Fläche tracken und mit einer hohen Framerate arbeiten können. Damit diese Koordinaten verarbeitet werden können und schließlich die Drohne geregelt wird, muss eine Verbindung zwischen dem Tracking-System, dem Computer und der Drohne konstruiert werden. Um das System als API zu erweitern, müssen zudem bestimmte Programmierschnittstellen implementiert werden, wie zum Beispiel ein Netzwerkprotokoll, um das System mit externen Geräten, wie beispielsweise Smartphones bedienen zu können.

Das zu implementierende System muss zum einen die Koordinaten der Drohnen erhalten und zum anderen die Drohnen steuern, beziehungsweise regeln können. Das System sollte den Drohnen Befehle geben, wie zum Beispiel, landen, starten und Aktionen, die mit der Drohne ausgeführt werden, empfangen können. Jede Drohne sollte sich vom System her selbst regeln können und erkennen, wann und wie mit ihr interagiert wird. Anschließend sollen diese Informationen an externe Geräte gesendet werden.

4 Hintergrund

In diesem Kapitel wird die Funktionsweise von Quadrocoptern erklärt, anschließend die Theorie der Regelung, die für die Drohne benötigt wird, und zuletzt wie die Drohne angesteuert wird und wie entwickeltes System die Koordinaten der Drohne erhält.

4.1 Drohnen

Unter Drohnen werden im Allgemeinen unbemannte Fahrzeuge verstanden, die mittels eines Computers oder über eine Fernbedienung gesteuert werden. Dabei gibt es verschiedene Fahrzeugtypen, die sich autonom bewegen können. Darunter gehören Landfahrzeuge, Luftfahrzeuge, U-Boote und Wasserfahrzeuge. In dieser Arbeit werden jedoch lediglich Luftfahrzeuge benutzt, speziell Quadrocopter.

4.1.1 Quadrocopter

Quadrocopter gehören zur Familie der Hubschrauber und besitzen vier in einer Ebene liegende Rotoren, mit denen sie sich in der Luft bewegen können. Nachdem im Jahr 2000 neue Gyroskopsensoren auf den Markt kamen und später in Quadrocopter eingebaut wurden, verbesserte sich die Stabilität der Quadrocopter im Flug. Da sie einfach zu fliegen sind und von der Mechanik einfacher und kostengünstiger zu bauen sind als Modell-Helikopter, sind Quadrocopter heutzutage sehr beliebt im Hobby-Bereich [19]. Jedoch werden die Drohnen in mehreren Gebieten auch kommerziell genutzt. Einen großen Bereich stellen etwa die Luftbildfotografie und -videografie dar. Dabei werden die Drohnen mit Kameras ausgestattet und fliegen über die Gebiete, die sie aufnehmen oder fotografieren sollen. Landschaftsaufnahmen in modernen Spielfilmen werden hauptsächlich mit Quadrocoptern aufgenommen. Zudem dienen die Drohnen zur Aufklärung in Krisengebieten, damit keine Menschen dort in Gefahr gelangen. Bei Umweltkatastrophen werden Drohnen zu Opfern geschickt, die z.B. nach einem Erdbeben verschüttet und schwer zugänglich sind, um sie mit lebensnotwendigen Hilfsmitteln zu versorgen.

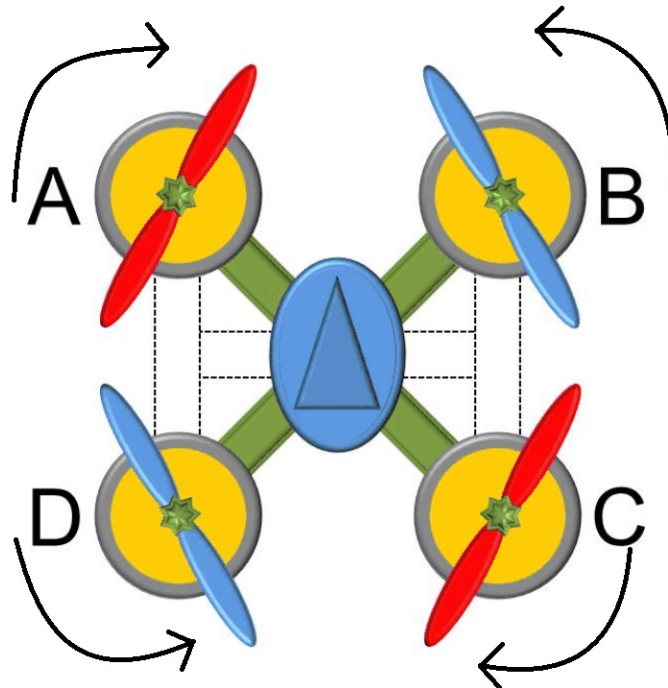


Abbildung 4.1: Dieses Bild zeigt die Bauweise eines Quadrocopters, Quelle: [20].

4.1.2 Technik der Quadrocopter

Quadrocopter sind mit vier Rotoren ausgestattet, dabei gibt es bisher zwei Bauarten, die x-Bauweise oder die H-Bauweise. Im Weiteren wird nur auf die x-Bauweise eingegangen. Abbildung 4.1 zeigt den typischen Aufbau der x-Bauweise. Dabei sind jeweils die gegenüberliegenden Rotoren gleich gebaut und drehen sich in die gleiche Richtung. Die Rotoren nebeneinander drehen sich in die entgegengesetzte Richtung. Es gibt drei Winkel, die die Ausrichtung einer Drohne beschreibt. Der Rollwinkel, dieser ist der Winkel um die Längsachse (Rollachse), der Nickwinkel ist der Winkel um die Querachse (Nickachse) und schließlich der Gierwinkel ist der Winkel um die Vertikalachse (Gierachse). Damit der Quadrocopter nach vorne beschleunigen kann, müssen sich die zwei hinteren Rotoren proportional schneller drehen als die vorderen zwei. Gilt das Gegenteil, beschleunigt die Drohne nach hinten. Um sich nach rechts zu bewegen, müssen sich die beiden linken Rotoren proportional schneller bewegen als die rechten Rotoren. Gilt hier wieder das Gegenteil, so fliegt die Drohne nach links. Eine rechts-Rotation um die Gierachse wird durch die proportionale Erhöhung der Rotoren A und C (siehe Abb. 4.1) erreicht.

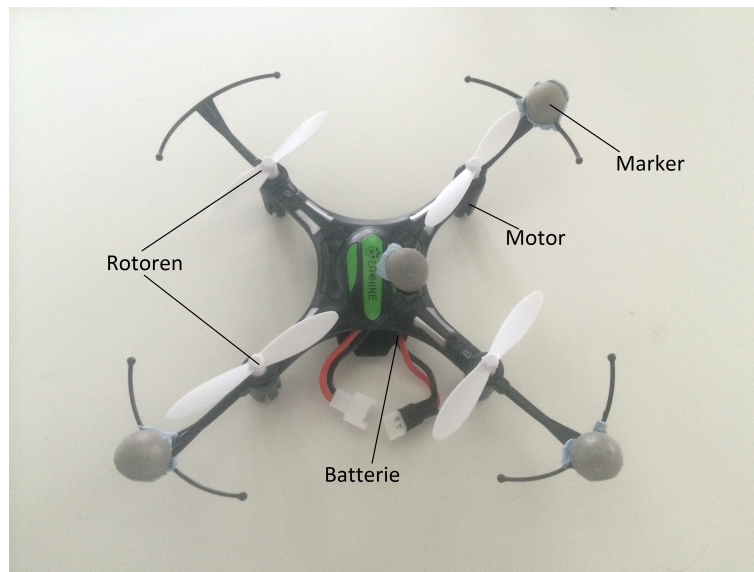


Abbildung 4.2: Bild der benutzten Drohne, Eachine H8 Mini RC.

4.1.3 Quadrocopter der Bachelorarbeit

In dieser Bachelorarbeit wurde die Eachine H8 Mini RC Drohne benutzt, (Abbildung 4.2). Sie ist eine sehr kostengünstige Drohne mit wenig Extras. Ihre Größe beträgt 10,5cm * 10,5cm und sie ist 2,8cm hoch. Sie wiegt in etwa 20g und kann zwischen 5-7 Minuten lang fliegen, ohne aufgeladen zu werden. Ihr einziger Sensor ist ein 6-Achsen Gyroskop, der für die Stabilisierung im Flug zuständig ist. Zusätzlich besitzt die Drohne einen Speed-Modus, der mittels Fernbedienung aktiviert werden kann. Wird dieser Modus aktiviert, so erhöht sich die Sensibilität der Drohne, wodurch sie größere Kippwinkel im Flug erreicht und dadurch schneller fliegen kann.

Durch ihr kleine, leichte und stabile Bauart kann die Drohne gut in geschlossenen Räumen benutzt werden, ohne dass man sich verletzt oder Gegenstände beschädigt. Dies war mit ein Grund dafür, weshalb diese Drohne verwendet wird. Zudem lässt sich die Drohne durch ihre Größe mit der Hand fassen, wodurch man mit ihr interagieren kann. Damit man der Drohne jedoch nicht so leicht in die Rotoren fassen kann und sie allgemein besser greifbar ist, wurde eine spezielle Konstruktion für die Drohne entwickelt, (Abbildung 4.3). Es ist ein Käfig, der die Maße 10,5cm * 10,5cm * 10,5cm hat. Das Material besteht aus Pappe mit einem Gewicht aus 120g pro Quadratmeter, damit die Konstruktion beim Greifen nicht zerstört wird, aber dennoch leicht genug ist, um von der Drohne getragen zu werden. Der Käfig ist an den Seiten nicht vollkommen mit Pappe umgeben, da die Konstruktion sonst zu schwer wäre und die Aerodynamik im Flug sich verschlechtern würde. In Test wurde herausgefunden, dass die besten Flugeigenschaften erzielt werden, wenn der Quadrocopter in mittlerer Höhe des Käfigs befestigt wird. In dieser Konfiguration wird eine maximale Flug Stabilität erreicht und unbeabsichtigtes Kippen der Quadrocopter, was zum Absturz führt, minimiert. Wie gut und



Abbildung 4.3: Hier sieht man die Konstruktion des Käfigs der Drohne.

angenehm der Käfig gegenüber Benutzern gewirkt hat, wurde in einer Studie getestet, die später im Kapitel 6 beschrieben wird.

4.2 Regelung

Um einen technischen Prozess oder ein System zu regeln, werden meistens Regler verwendet. Diese regeln einen Istwert trotz Störeinflüssen zu einem vorgegebenen Sollwert. Dabei wird im Regler die Differenz $e(t)$ (Regelabweichung) aus der Führungsgröße (Sollwert) und Regelgröße (Istwert) zu einer Stellgröße $u(t)$ berechnet, die die Regelstrecke so beeinflusst, dass die Regelabweichung minimal wird. Abbildung 4.4 veranschaulicht diesen Vorgang.

Es gibt viele verschiedene Arten von Reglern, für die Drohnenregelung wurde jedoch ausschließlich ein stetig linearer PID-Regler verwendet.

4.2.1 PID-Regler

PID-Regler steht für Proportional-Integral-Differential und besteht aus drei Komponenten, den proportionalen, integrierten und den differenzierten Anteil. Der proportionale Anteil berechnet sich aus der Regelabweichung $e(t)$ multipliziert mit einer Verstärkungskonstanten K_p :

$$u_p(t) = K_p * e(t)$$

Der integrierte Anteil ist die Integration der Regelabweichung $e(t)$ über die Zeit t multipliziert mit der Verstärkungskonstanten K_i :

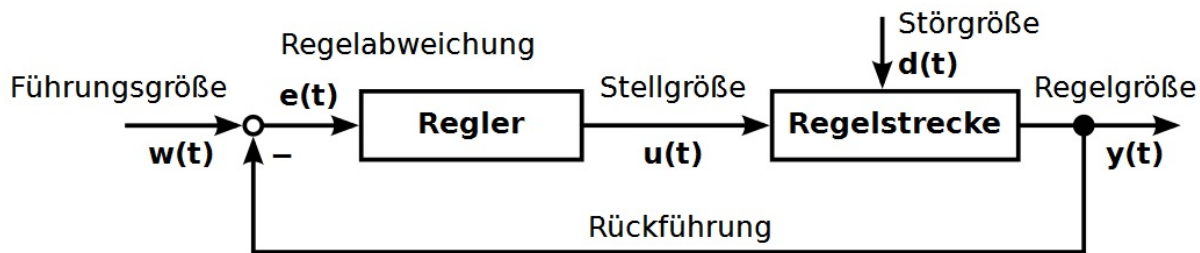


Abbildung 4.4: Dieses Bild zeigt die theoretische Benutzung von Reglern. Quelle: [21]

$$u_i(t) = K_i * \int_0^t e(t) dt$$

Der differenzierte Anteil besteht aus der Verstärkungskonstanten K_d multipliziert mit der Ableitung der Regelabweichung $e(t)$ zum Zeitpunkt t :

$$u_d(t) = K_d * \frac{d}{dt} e(t)$$

Die endgültige Stellgröße ergibt sich schließlich aus der Addition aller drei Komponenten:

$$u(t) = u_p(t) + u_i(t) + u_d(t)$$

Jeder dieser drei Anteile hat seinen eigenen Zweck. Der proportionale Anteil reduziert die Rise-Time, vergrößert den Overshoot und verringert den Steady-State Error. In Abbildung 4.5 lässt sich der zeitliche Verlauf einer geregelten Größe erkennen. Hierbei ist die Rise-Time die Zeit, die benötigt wird, bis die Regelgröße das erste Mal die Führungsgröße erreicht hat. Overshoot bezeichnet die Überkorrektur. Der Steady-State Error ist die Differenz von der Regel- und Führungsgröße, nachdem der PID-Controller seine Korrektur beendet hat. Die Setting-Time ist die Zeit, die der Regler braucht, um die Regelgröße in eine anhaltende akzeptable Größe zu bringen [22]. Die Integral-Komponente verringert die Rise- und Setting-Time, vergrößert den Overshoot und eliminiert den Steady-State Error. Der differentiale Anteil verringert den Overshoot und die Setting Time [23].

Je nach dem zu regelnden System müssen die Verstärkungskonstanten K bestimmt werden. Dieser Vorgang wird PID-Tuning genannt und kann durch verschiedene Ansätze vollzogen werden. In dieser Arbeit wurde das PID-Tuning experimentell durchgeführt.

4.3 Setup

Der Flugbereich der Drohne befand sich in einem Labor der Universität Stuttgart, dort wurde sie mittels eines Optitracksystems getrackt und anschließend mit einem Computer gesteuert. Die Schnittstellen zwischen dem OptiTrack, der Drohne und dem Computer werden in den nächsten Unterkapiteln näher erläutert.

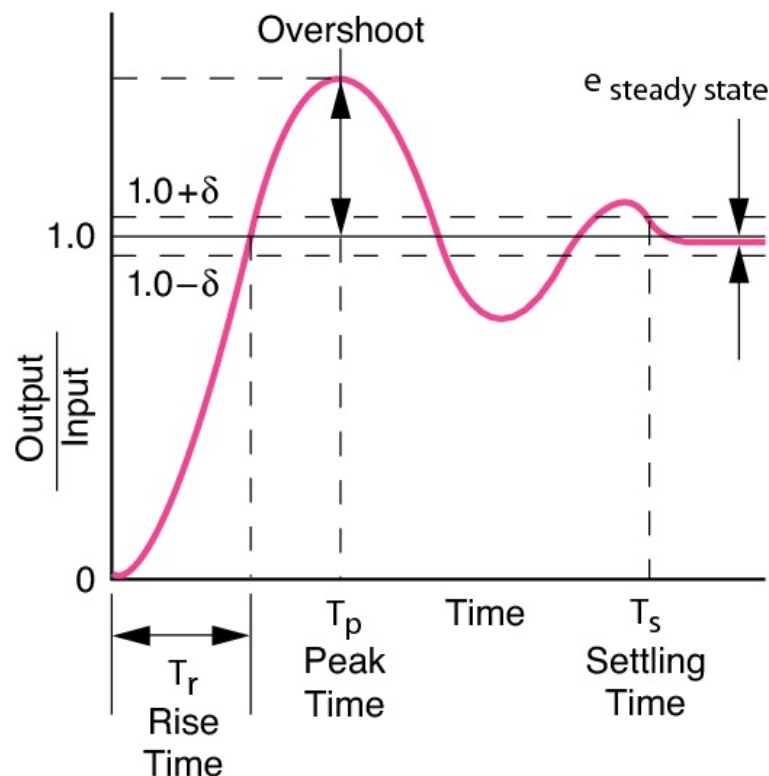


Abbildung 4.5: Diese Abbild zeigt ein Beispiel einer geregelten Größe [22].

4.3.1 Optitrack

OptiTrack™ ist der größte Anbieter zum Positionstracking. Er bietet verschiedene Kameras an, die die Positionen von Markern erkennen können, sowie Software wie zum Beispiel “Motive“ und “Expression“, welche Gesichtsmimiken erfassen kann. Unternehmen, die mit OptiTrack arbeiten, sind unter anderem Activision, Google, Disney und NASA [24].

Um die Drohne zu tracken, wurden 8 Prime 13W Kameras von OptiTrack™ verwendet. Diese haben eine Auflösung von 1.3MP (1280*1024), eine maximale Framerate von 240FPS und einen horizontalen Sehbereich von 82°. Dadurch liefern sie eine sehr gute Qualität, weshalb dieses System eingesetzt wurde. Damit die Kameras die Position der Drohne bestimmen können, wurde diese mit Markern versehen, die eine Infrarot reflektierende Oberfläche haben. Die Kameras senden Infrarotlicht aus, welches von den Markern der Drohne reflektiert und wieder von den Kameras aufgenommen. Diese berechnen dann die Informationen in Echtzeit zu 3D-Koordinaten um. Dabei können sie auch die Rotation um die X-, Y- und Z-Achse ermitteln. Diese Daten werden an einen Router gesendet, der diese per UDP-Verbindung verteilt. Die Kameras wurden auf einen Wagen montiert, damit mein System mobil ist und an anderen Positionen verwendet werden kann, (Abbildung 4.6).

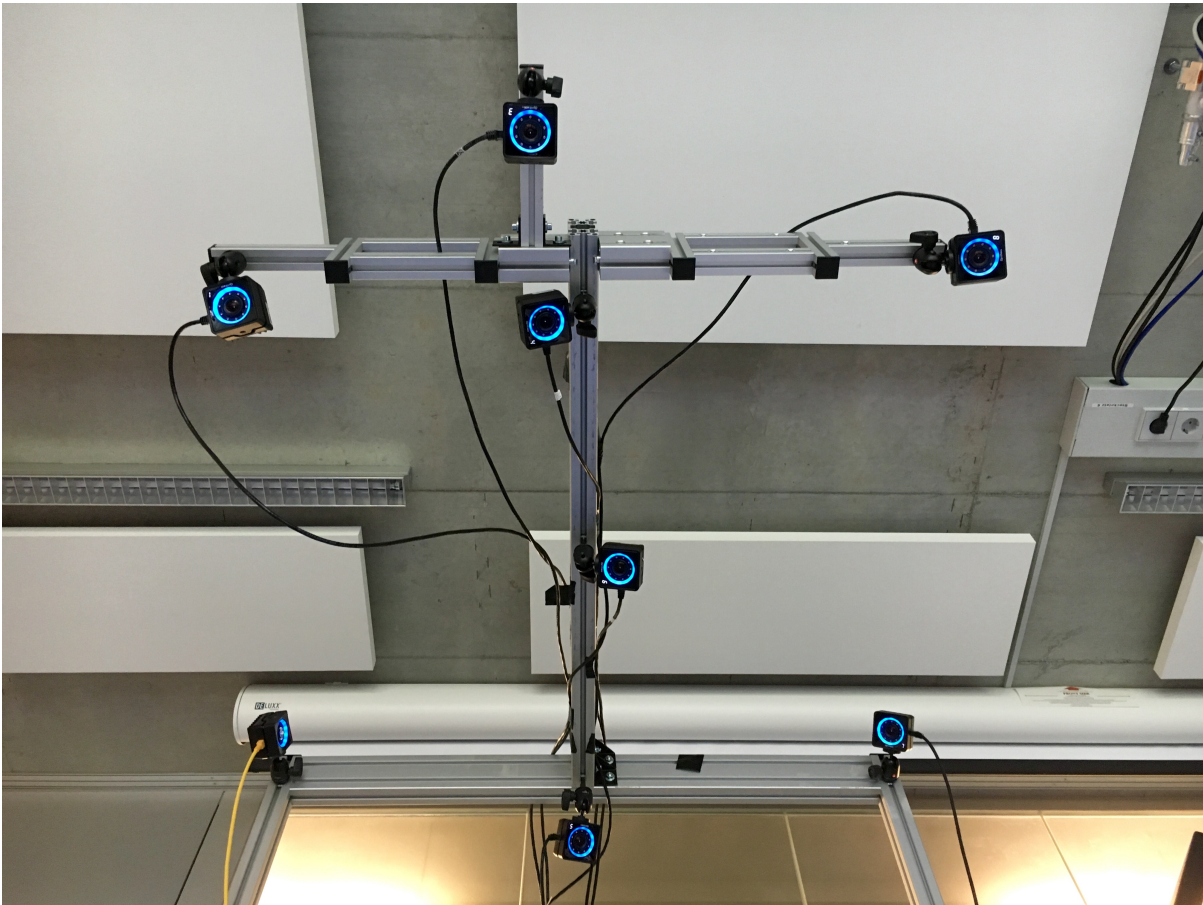


Abbildung 4.6: Dieses Bild zeigt die verwendeten Optitrack-Kameras.

4.3.2 Steuerung der Drohne

Damit die Drohne per Computer gesteuert werden kann, wurde die Fernbedienung über ein Arduino UNO an den Computer verbunden, (Abbildung 4.7). Die zugehörige Fernbedienung der Drohne kann lediglich Daten senden, nicht hingegen solche empfangen. Somit ist es z.B. nicht möglich, den Akkustand der Drohne zu erfassen. Die Hebel mit dem dazugehörigen Potentiometer wurden entfernt und die verbliebenen Schnittstellen mit dem Arduino verbunden. Der Arduino UNO ist ein programmierbarer Mikrocontroller mit 14 digitalen Input/Outputs, 6 analogen Inputs und einer USB-Schnittstelle für die Verbindung mit dem Computer. Mit Hilfe des Arduinos kann ein Signal generiert werden, welches das des Potenziometers nachahmt. Dadurch lässt sich die Drohne nach oben/unten, links/rechts, vorne/hinten und rechts oder links um die Gierachse rotieren. Dem Arduino werden nach dem von mir entwickelten System jeweils Werte von 0 bis 255 gesendet, dabei ist beispielsweise 0 voller Schub nach hinten und 255 voller Schub nach vorne. Bei der Höhensteuerung werden dem Arduino ebenfalls Werte von 0 bis 255 gesendet, jedoch werden hier bei 0 die Rotoren ausgeschaltet und bei 255 mit voller Leistung betrieben, um mit dem maximalen Schub nach oben zu fliegen. Bei der

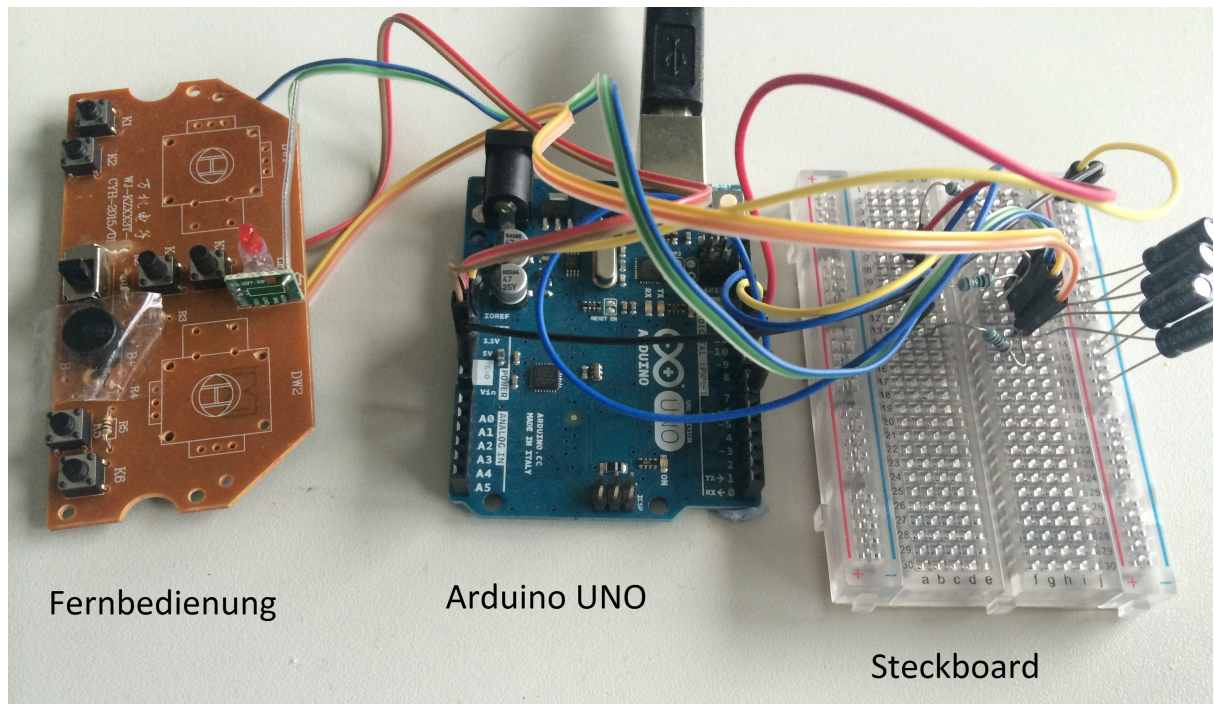


Abbildung 4.7: Dieses Bild zeigt die Verbindung zwischen dem Computer und der Fernbedienung über einen Arduino UNO. Links die Fernbedienung, in der Mitte der Arduino UNO, rechts ein Steckboard.

horizontalen Bewegung müsste theoretisch der Wert 128 jeweils keine Bewegung bedeuten, wegen der schlechten Verarbeitung der Drohne ist dies leider nicht der Fall. Der Wert, der dafür zuständig ist, dass die Drohne sich im Flug nicht seitlich bewegt, muss erst ermittelt werden.

5 Prototyp Entwicklung

In diesem Kapitel wird zu Beginn eine grobe Übersicht über die Software-Architektur gegeben. Es wird zunächst beschrieben, was für Klassen es gibt und wie diese miteinander kommunizieren. Daraufhin werden die verschiedenen implementierten Modi vorgestellt, die in Output und Input aufgeteilt sind. Anschließend wird die Genauigkeit und Stabilität der Modi, das Starten, Hovern, Landen und Route Abfliegen präsentiert, was in einer Systemevaluation untersucht wurde.

5.1 Software-Architektur

Das implementierte System wurde in C# programmiert. Die Hauptklassen sind MainWindow, TCPServer, UDP, Drone und Controller (Abbildung 5.1). Dabei hat MainWindow eine zentrale Rolle. Die Klasse lässt die anderen Klassen miteinander kommunizieren und kümmert sich dabei darum, dass alle Drohnen die richtigen Funktionen ausführen. Dabei enthält es zudem eine graphische Benutzeroberfläche, die einem Benutzer die indirekte Interaktion mit der Drohne erlaubt und ihm Informationen über die Drohne ausgibt. Die graphische Oberfläche wird später näher erläutert. Das Programm ist dafür ausgelegt, mehrere Drohnen zu unterstützen und zu regeln. Deswegen kann die MainWindow-Klasse mehrere Drohnen-Klassen erzeugen und verwalten. Das Optitrack System schickt die Koordinaten aller Objekte, die getrackt werden, über eine UDP-Verbindung in das Netzwerk. Daraufhin greift die UDP-Klasse des Systems zu und empfängt alle Koordinaten. Zu den Koordinaten schickt das Programm über die UDP-Verbindung zusätzlich die ID des getrackten Objekts und die Ausrichtung (Roll-, Nick-, und Gierwinkel) mit. Die MainWindow-Klasse greift auf die empfangenen Koordinaten zu und verteilt diese über die entsprechenden IDs an die richtigen Drohnen. Somit weiß MainWindow zu jeder Zeit, wo sich welche Drohne befindet. Jede Drohnen-Klasse besitzt eine eigene Controller-Klasse. In dieser Klasse wird berechnet, was die Drohne der Fernbedienung schicken muss. Sie besitzt vier PID-Regler, jeweils für die x-, y- und z-Position sowie die Ausrichtung des Gierwinkels. Dabei berechnet sie jeweils aus dem aktuellen Wert und dem Setpoint den Wert, der der Fernbedienung zugeschickt werden muss, damit die Drohne den Setpoint erreichen und halten kann. Insgesamt werden somit vier Werte ausgegeben und zurück an die Drohnen-Klasse geschickt. Die Drohnen-Klasse besitzt die Schnittstelle zu dem Arduino, mit dem die Werte an die Fernbedienung gesendet werden können. Will die Drohne nun an eine neue Position fliegen und an dieser Stelle hovern, so schickt die Drohnen-Klasse einen neuen Setpoint zu der Controller-Klasse.

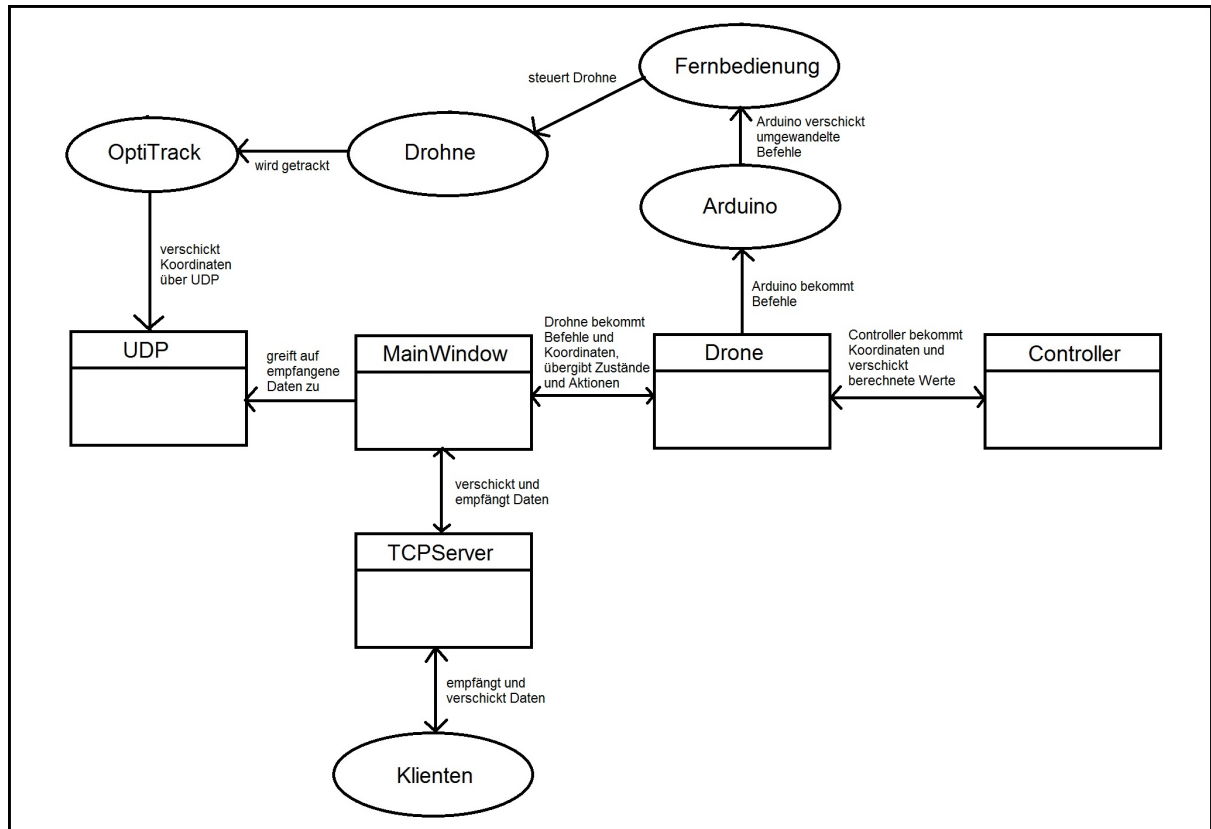


Abbildung 5.1: Klassendiagramm des implementierten Systems.

Zusätzlich hat die MainWindow-Klasse eine Schnittstelle zu einem asynchronen TCP-Server, in dem Informationen der Drohne an verschiedene Klienten geschickt werden können. Wenn die Drohne eine neue Position einnimmt oder mit ihr interagiert wird, so sendet die Drohnen-Klasse diese Informationen an die MainWindow-Klasse und diese wiederum an den TCP-Server. Der TCP-Server kann auch Befehle von den Klienten zugesendet bekommen. Dadurch kann beispielsweise die Drohne auf eine neue Position gebracht oder eine Landung gestartet werden. In der MainWindow-Klasse werden den Drohnen die verschiedenen Aufgaben zugeteilt, wie zum Beispiel, “starte“, “lande“, “fliege zur Position x,y,z“.

Damit der Benutzer dies steuern kann, wurde eine graphische Benutzerschnittstelle erstellt, die in Abbildung 5.2 zu sehen ist. Im oberen linken Bereich kann die Drohne ausgewählt werden, mit der man interagieren möchte. Rechts davon werden die Koordinaten und die Werte der Ausrichtung ausgegeben. Diese werden in Echtzeit geplottet, was in den Schaubildern unten rechts zu sehen ist. Dabei steht die schwarze Linie für die aktuelle Position der Drohne und die rote für den Setpoint. Dadurch lässt sich erkennen, wie weit sie von ihrem eigentlichen Ziel entfernt ist und wie schnell sie sich bewegt. Die Auflistungen der Punkte unten links sind für den Modus “Route abfliegen“, der im nächsten Unterkapitel näher erläutert wird. Hier kann man die Punkte, die die Drohne abfliegen soll, eingeben und den Modus starten. Neben dem Punkt, der momentan angefliegen wird, aktualisiert sich der Ladebalken entsprechend dem

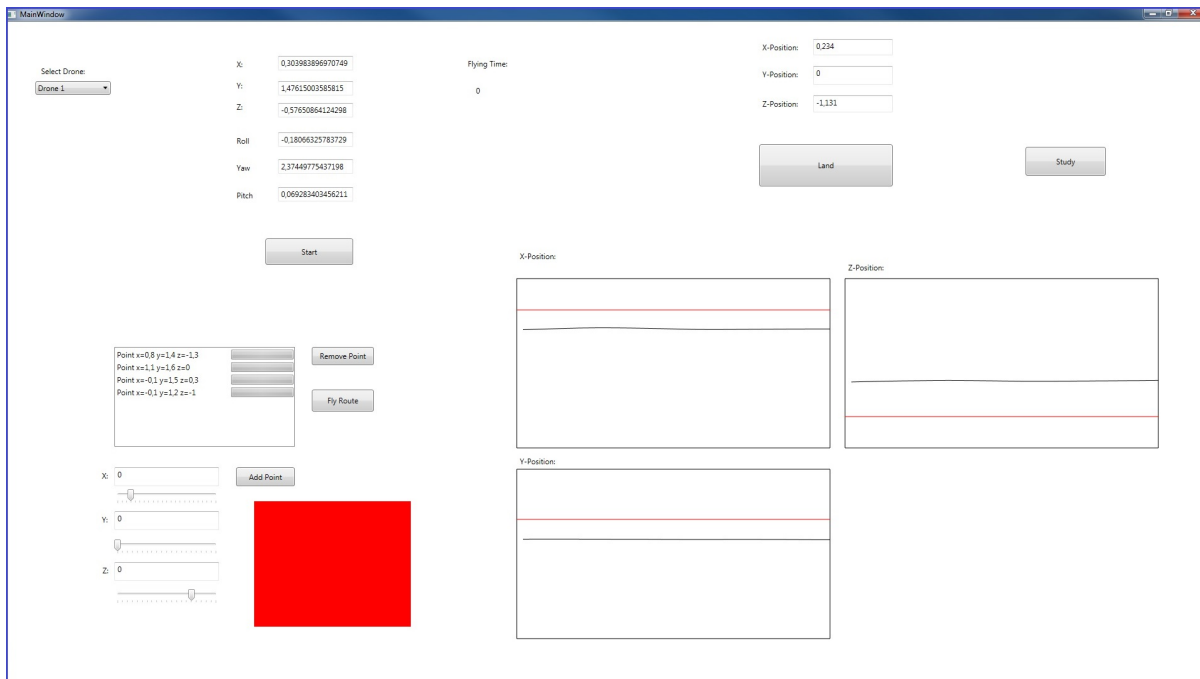


Abbildung 5.2: Die graphische Oberfläche des Systems.

Abstand zu dem Zielpunkt. Ist der Punkt erreicht, wird der Ladebalken des nächsten Punktes aktualisiert. Dadurch kann der Benutzer besser erkennen, welchen Punkt die Drohne ansteuert und wie weit sie davon entfernt ist. Das rote Rechteck gibt den Zustand der Drohne wieder, der in den Input-Modi vorkommt. Dabei gibt es die Farben rot, grün und blau. Rot bedeutet die Drohne kalibriert, grün steht dafür, dass die Drohne bereit ist zum Interagieren, und blau gibt an, dass das System die Drohne als gegriffen erkennt. Dadurch hat der Benutzer eine bessere Übersicht, was die Drohne gerade macht. Wann diese Zustände zustande kommen, wird im anschließenden Kapitel erklärt. Oben rechts der GUI befindet sich der Lande-Button, um die Drohne zu landen.

5.2 Implementierte Funktionalität

Der wichtigste Modus, damit die anderen Modi auch stabil und zuverlässig ablaufen können, ist der Hover-Modus. Hierbei ist das Ziel, dass die Drohne einen bestimmten Punkt im Raum stabil und ohne große Schwankungen halten kann. Um dies zu erreichen, wird die Drohne mit vier PID-Reglern geregelt. Für die Bewegung in x-, y- und z-Richtung sowie die Rotation um die Gierachse wird jeweils ein PID-Regler verwendet. Damit die Drohne schnell einen Punkt erreichen und stabil halten kann, ist es wichtig, die Verstärkungskonstanten der jeweiligen Regler richtig einzustellen. Ein weiterer wichtiger Punkt ist, die Anfangswerte der seitlichen Bewegung schon relativ genau ermittelt zu haben. Somit braucht der Integral-Teil des Reglers

nicht so lange, um den richtigen Wert zu berechnen. Die Verstärkungskonstanten wurden experimentell ermittelt. Zu Beginn wurden Standardwerte genommen und das anschließende Flugverhalten gab Informationen, wie die Verstärkungskonstanten geändert werden müssen. Bei zu großen Schwingungen musste der Differentialteil erhöht werden und eventuell der Integralteil vermindert. Wenn die Drohne mit einer großen Differenz zu ihrem eigentlichen Zielpunkt hovert, musste der Proportionalteil erhöht werden. Die Drohne wurde mehrmals getestet und die Verstärkungskonstanten aktualisiert, bis eine ausreichende Flugstabilität herrschte. Um die richtigen Anfangswerte für den Integralteil zu ermitteln, wurde die Drohne, nachdem die Verstärkungskonstanten bestimmt wurden, getestet und die Koordinaten und die Werte, die der Fernbedienung zugeschickt wurden, aufgezeichnet. Daraufhin wurde ein Durchschnittswert dieser Werte berechnet und als neuer Integral-Startwert benutzt. Dieser Vorgang wurde so oft wiederholt, bis die Werte zu einem bestimmten Wert konvergierten.

In jedem implementierten Modus war die Ausrichtung um die Gierachse konstant. Versuchte man die Drohne zu drehen, versuchte sie selbstständig wieder zur Ausgangsausrichtung zurückzukehren. Dadurch war es leider nicht möglich, sie im Flug zu drehen oder sie in eine andere Ausrichtung zu bringen. Dies ist ein Punkt, den man bei späteren Arbeiten an dem System verbessern könnte. Der implementierte Start beginnt die Regelung der Drohne und setzt einen neuen SetPoint, der sich nur in der Höhe ändert, die x- und z- Koordinaten bleiben gleich. Somit ist es möglich, von überall zu starten. Bei der Landung wird ebenfalls nur die Höhe zu 0 gesetzt. Zudem werden aber auch die Verstärkungskonstanten der Regler neu definiert. Die der x- und z-Regler bleiben erhalten, bei dem y-Regler erhöht sich allerdings die Integral-Verstärkungskonstante und die Proportional-Verstärkungskonstante wird zu einem sehr kleinen Wert geändert. Dadurch regelt der Integralteil hauptsächlich die Landung, was zur Folge hat, dass diese einer konstanten Höhenänderung unterliegt. Abbildung 5.3 zeigt die Höhe der Drohne während eines Testfluges. Dabei sieht man, dass die gewünschte Flughöhe von 1,75m innerhalb in etwa 1,8 Sekunden zum ersten Mal erreicht wird. Danach braucht die Drohne weitere 2 Sekunden, bis sie die Flughöhe stabil halten kann. Nach 30 Sekunden wurde eine Landung gestartet, die nach ca. 4 Sekunden beendet war.

5.2.1 Output-Funktionalität

Nachdem der Hover-Modus stabil funktionierte, wurden weitere Modi implementiert. Ein Modus, der in die Kategorie Output fällt, ist der "Route Abfliegen"-Modus. Hier ist das Ziel, mehrere eingegebene Punkte im Raum möglichst schnell und genau zu erreichen und danach zu dem nächsten Punkt zu fliegen. Ein Punkt gilt als erreicht, wenn die Drohne in einem Radius von 15,5 cm des Punktes liegt. Dieser Wert lässt sich variabel verändern, wurde für die Systemevaluation jedoch in allen abgeflogenen Routen verwendet. Wurden alle Punkte abgeflogen, wird wieder zum ersten Punkt geflogen und der Vorgang beginnt von vorne. Wichtig ist hierbei, den Integralteil der x- und z-Berechnung zu minimieren. Geschieht dies nicht, wird nach Erreichen eines Punktes die gesamte Fehlerdifferenz auf das Integral hinzu addiert und die Drohne fliegt über den nächsten Punkt hinaus. Es kann jedoch passieren, dass die Drohne zu ungenau auf den nächsten Punkt fliegt. Um dies zu verhindern, muss der Integralteil

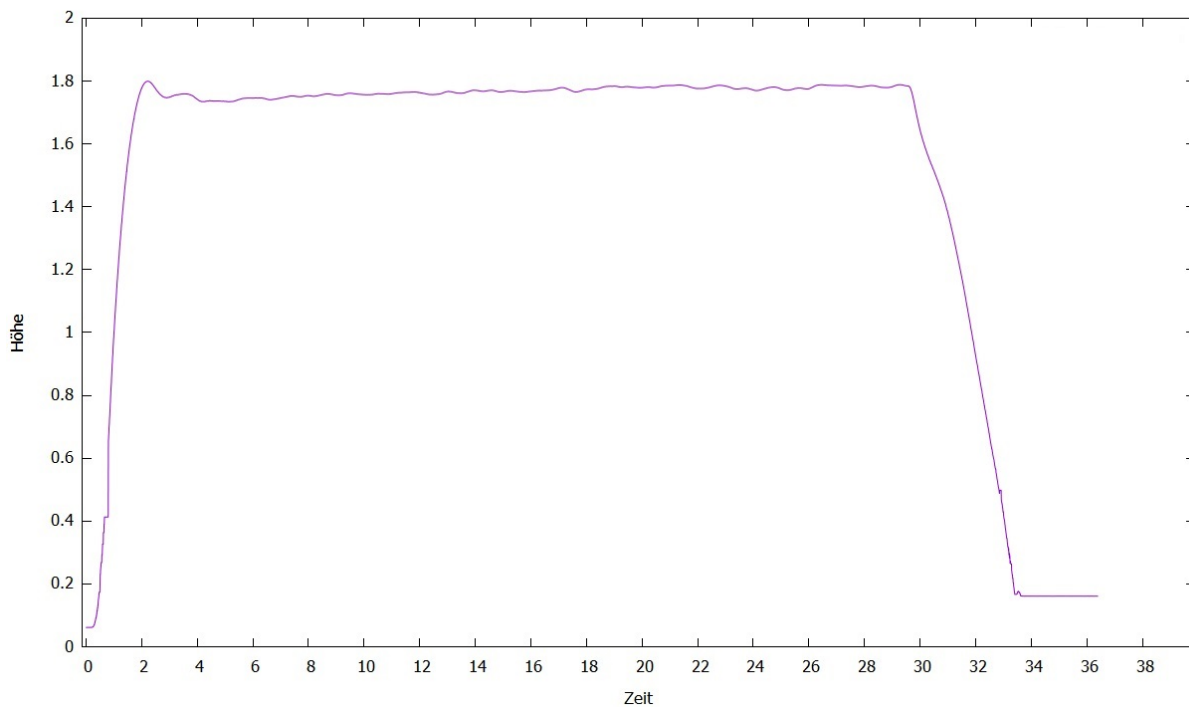


Abbildung 5.3: Diese Abbildung zeigt die aufgenommenen Höhenkoordinaten während eines Fluges.

wieder vergrößert werden. Deshalb wurde hier ein Timer verwendet. Wenn die Drohne nach einer kurzen Zeit den Punkt nicht genau anfliegen kann, so wird die Verstärkungskonstante des Integralteils wieder erhöht, bis der Punkt erreicht wurde. Dieses Verfahren funktionierte gut, was in einen späteren Unterkapitel näher betrachtet wird. Durch diesen Modus kann eine Drohne jeden eindimensionalen bis dreidimensionalen Datenpunkt im Raum darstellen. Somit könnte man die Drohne zum Beispiel Schwingungen von beliebigen Simulationsdaten darstellen lassen.

Es wurde zusätzlich ein anderer Modus implementiert, der dem Benutzer Informationen zukommen lassen kann. Dabei handelt es sich um einen Modus, der dem Benutzer haptische Feedbacks gibt. Dies funktioniert nur, solange der Benutzer die Drohne in der Hand hält. Greift der Benutzer nach der Drohne, versucht sie anfangs normal zu hovern. Dadurch spürt der Benutzer durch die Drohne keine großen Richtungsimpulse. Aktiviert man nun den Modus, so versucht die Drohne dem Benutzer ein haptisches Signal zu geben, indem sie versucht, mit vollem Schub in eine Richtung zu fliegen. Dadurch bemerkt der Benutzer, dass die Drohne ihm etwas mitteilen will und spürt diesen Druck. Nun wurde diese Methode auch so verwendet, dass, wenn der Benutzer die Drohne greift und sie in eine Richtung bewegt, sie, nachdem sie einen bestimmten Punkt überschritten hat, dem Benutzer wieder einen Impuls zukommen lässt. Dadurch kann man dem Benutzer beispielsweise Grenzen bestimmter Skalen deutlich machen.

Ist die Skala auf einen Meter in x-Richtung ausgelegt und wird die Drohne darüber hinaus bewegt, versucht sie, mit vollem Schub in die entgegengesetzte Richtung zu fliegen. Somit weiß der Benutzer, wann er die Skalagrenze überschritten hat. Dieser Modus wurde durch eine Probandenstudie evaluiert, die im Unterkapitel proof-of-concept beschrieben wird.

5.2.2 Input-Funktionalität

Damit der Benutzer die Drohne auch als Eingabegerät benutzen kann, wurden hierfür zwei verschiedene Funktionalitäten implementiert. Ein Modus davon ist der "Button Press"-Modus. Hierbei hovers die Drohne an einem beliebigen Punkt. Der Benutzer kann nun zur Drohne gehen und sie wie einen Knopf drücken. Dabei kann er sie nach vorne, hinten, links, rechts und oben drücken. Das System erkennt anhand der Geschwindigkeitsänderung, in welche Richtung die Drohne gedrückt wurde. Wurde die Drohne stärker gedrückt und dadurch etwas bewegt, so fliegt sie auf ihre Ausgangsposition zurück. Nach jedem Klick wartet das System, bis die Drohne wieder am vorherigen Platz stabil hovers. Erst danach kann das System neue Klicks erkennen. Dies geschieht deshalb, damit Klicks, die von der Drohne erzeugt wurden während sie wieder auf die richtige Position fliegt, zu vermeiden. Jede Klick-Richtung kann eine andere Auswirkung haben, je nachdem, welche Aktion man mit welchem Klick ausführen lassen möchte. Die Drohne nach unten drücken wurde nicht implementiert, da sonst die Marker für das OptiTrack System mit der Hand überdeckt werden und somit die Drohne nicht mehr getrackt wird und deshalb abstürzen würde. Wurde ein Klick betätigt, so wurde dies an den TCP-Server geschickt, der die Informationen zu allen Klienten weitersendet.

Der andere Input-Funktionalität funktioniert durch Verschieben der Drohne. Hat die Drohne fertig kalibriert, das heißt in diesem Fall, sie hovers an einem Punkt und die Geschwindigkeit der Drohne ist minimal, so kann die Drohne gegriffen werden. Nachdem der Benutzer die Drohne in der Hand hält, kann er sie an eine neue Position bringen und sie anschließend wieder loslassen. Nun hovers die Drohne an der neuen Position. In diesem Modus hat die Drohne drei Zustände. Der erste ist der Kalibrierzustand, in dem die Drohne versucht, sich an einer Position zu stabilisieren. In diesem Zustand darf die Drohne nicht gegriffen werden. Hat die Drohne sich nun kalibriert, fällt sie in den Zustand "Bereit". Nun darf die Drohne gegriffen werden und beliebig neu positioniert. Wenn nun eine plötzliche Geschwindigkeitsänderung vorkommt, erkennt das System dies als Handbewegung mit der Drohne. Somit fällt die Drohne in den letzten Zustand "wird bewegt". Hier werden die Rotoren der Drohne so angesteuert wie beim letzten Hovern, damit die Drohne dem Benutzer keine Kräfte in eine Richtung zukommen lässt. Wird die Drohne nun nicht mehr bewegt, sind also die Geschwindigkeiten in alle Richtungen minimal, so erkennt das System dies als neue Position an, die Drohne fällt wieder in den Modus "kalibriert" und versucht, an der neuen Position zu hovern. Dies kann nun beliebig oft wiederholt werden. Nach jedem Statuswechsel wird der neue Status an den TCP-Server gesendet, der diesen an die verbundenen Klienten schickt. Damit lassen sich zum Beispiel dreidimensionale Daten manipulieren. Präsentiert die Drohne beispielsweise drei verschiedene Lautstärken für drei Geräte, so könnte jede Lautstärke durch eine Dimension mit der Drohne

verbunden werden. Bewegt man die Drohne in x-Richtung, so ändert sich die Lautstärke von Gerät 1, bewegt man sie in y-Richtung, so ändert sich die Lautstärke von Gerät 2 usw.

Beide vorgestellten Input-Modi lassen sich kombinieren, damit abwechselnd geklickt oder die Drohne neu positioniert werden kann. Ein Problem hierbei ist jedoch, wie das System erkennt, ob die Drohne nun geklickt oder bewegt wurde. Deshalb wurde zusätzlich ein Timer implementiert, der bei der Kombination der beiden Modi aktiviert wird. Wird von dem System eine rasante Geschwindigkeitsänderung wahrgenommen, so weiß es zunächst nicht, ob die Drohne geklickt oder gegriffen wurde. Deswegen startet nun ein Timer. Wenn sich die Drohne nach einer bestimmten Zeit immer noch relativ nahe ihrer Ursprungsposition befindet, so wird die Bewegung der Drohne als Klick registriert. Wenn die Drohne sich jedoch zu weit von ihrer eigentlichen Position entfernt hat, wird dies als "Bewegung durch Hand" registriert. Diese Methode funktioniert nur, wenn die Drohne nicht übertrieben stark geklickt bzw. geschlagen wird, und man nicht versucht, die Drohne nur um wenige Zentimeter zu verschieben. Eine andere Voraussetzung, dass der Modus funktioniert, ist, dass die Drohne stabil fliegt. Hat sie aufgrund irgendwelcher Faktoren Probleme und fliegt sie deshalb teilweise unruhig, so könnte ab und zu ein Klick oder eine Handinteraktion vom System erkannt werden, die eigentlich keine ist. Fatal ist es, wenn das System die Drohne dadurch als "gegriffen" erkennt. Aufgrund dessen würde die Drohne konstant in eine Richtung fliegen, bis sie entweder manuell gestoppt wird oder außerhalb des Trackingbereichs kommt und schließlich abstürzt. Dies könnte verhindert werden, indem der Benutzer mit einem trackbaren Handschuh ausgestattet werden würde. Dies wurde in diesem System aber absichtlich nicht implementiert, da mit dem System ohne Hilfsmittel interagiert werden soll. Somit kann jeder Mensch mit dem System interagieren. Soll beispielsweise mit der Drohne spontan ein Gerät im Haushalt gesteuert werden, so soll der Benutzer davor sich nicht ausrüsten müssen um das System benutzen zu können. Dies würde die Bequemlichkeit und Spontanität des Systems zerstören.

5.3 Proof-of-concept

Um die Stabilität und Genauigkeit des Systems zu untersuchen, wurden mehrere Testflüge unternommen. Dabei sollte herausgefunden werden, welche Aktionen man mit den Flugeigenschaften der Drohne durchführen kann und welche nicht. Zudem war es wichtig, die Abweichungen von bestimmten Funktionalitäten herauszufinden, um damit den Zustand der Drohne besser beschreiben zu können. Ist die Abweichung beispielsweise um einiges größer als die berechnete Standardabweichung, so stimmt vermutlich etwas mit der Drohne nicht oder es wird mit ihr interagiert. Bei jedem Testflug wurde alle 6 Millisekunden die Position der Drohne gespeichert. Es wurde unter anderem die Genauigkeit des Hovers und des Route Abfliegen untersucht, was im Folgenden separat vorgestellt wird.

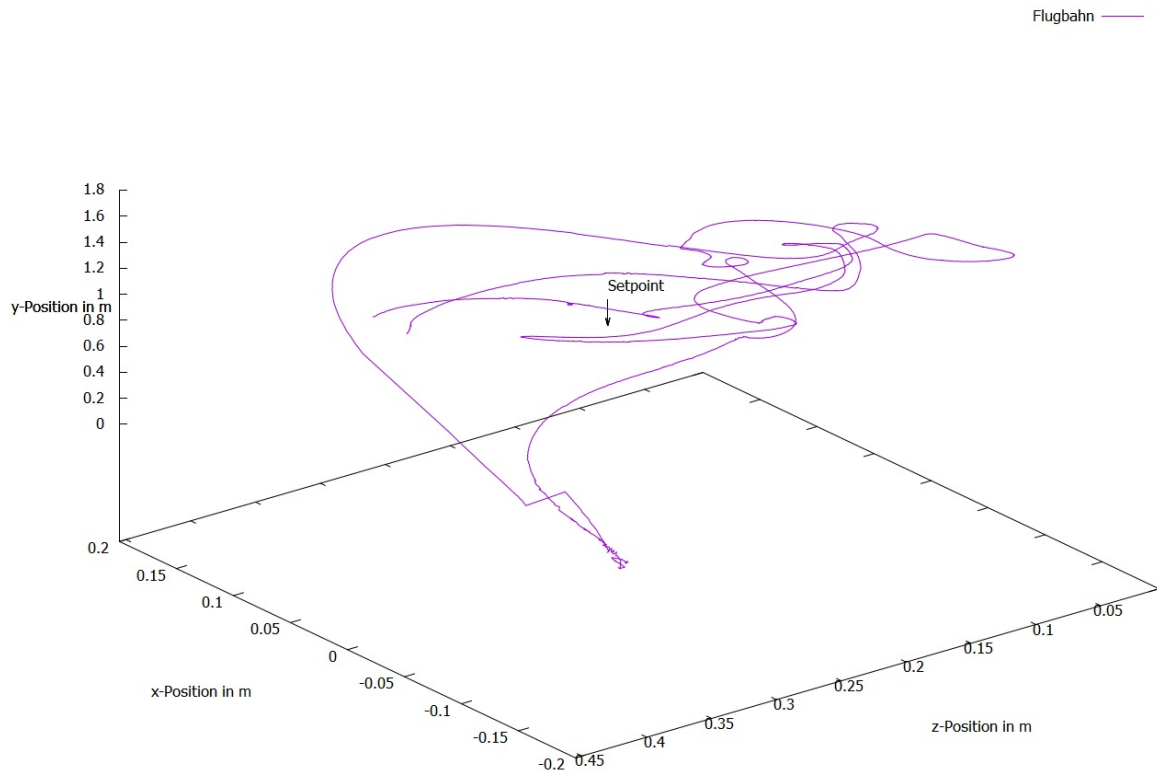


Abbildung 5.4: Flugbahn eines Testfluges, in der die Drohne an einer festen Position hovern sollte.

5.3.1 Hovern

Um die Genauigkeit des Hovers relativ genau zu bestimmen, musste die Drohne zehnmal an der gleichen Position hovern. Nachdem die Drohne 30 Sekunden lang geflogen war, wurde die Landung angesetzt. Bei der Berechnung der Abweichungen wurde der Start und der Landevorgang nicht miteinbezogen. Aus den gespeicherten Koordinaten während des hovers wurde pro Koordinate der Betrag des euklidischen Abstands zum Setpoint berechnet. Schließlich wurde aus allen berechneten Abständen der Durchschnitt berechnet. Zusätzlich zum Gesamtabstand wurden die einzelnen Koordinatenabstände berechnet (x, y, z), um dadurch erkennen zu können, in welche Richtungen die Drohne gegebenenfalls noch Verbesserungen braucht. Abbildung 5.4 zeigt ein Beispielplot, in dem die Drohne hovern sollte. Hierbei erkennt man, dass die Drohne beim Starten stärkere Ausschweifungen hat, die anschließend kleiner werden.

Nach den 10 aufgenommenen Flügen wurde eine insgesamt Durchschnittsabweichung von 10,3 cm berechnet. Das heißt, dass die Drohne durchschnittlich in einem Radius von 10,3 cm um den Setpoint umher schwankt. Um jedoch zu sehen, ob diese Schwankungen in jede Richtung gleich stark sind, wurden zudem die einzelnen Abweichungen jeder Achse berechnet. Abbildung 5.6 zeigt die einzelnen Werte im Vergleich. Hierbei ist zu beachten, dass die insgesamt

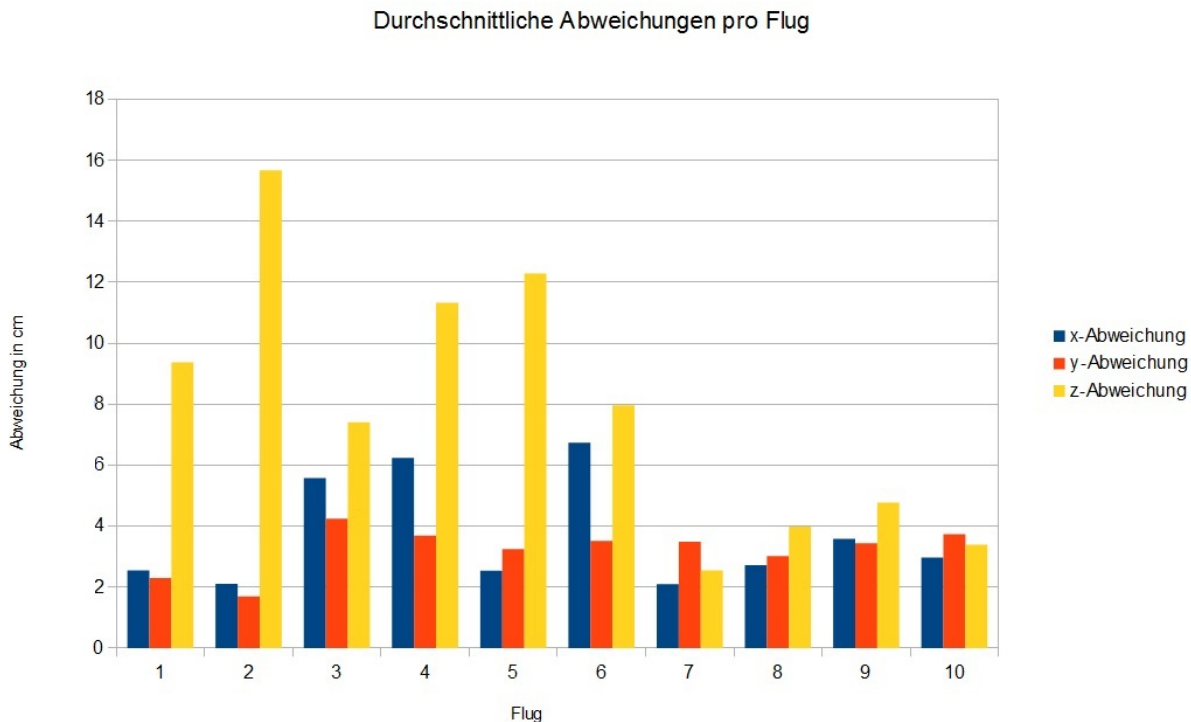


Abbildung 5.5: Abweichungen pro Flug.

Durchschnittsabweichung höher als der Durchschnitt der einzelnen Achsen Abstand ist, da hier nur der Abstand auf einer Achse berechnet wurde. Dabei beträgt die Abweichung in x-Richtung 3,7 cm, in y-Richtung (Höhe) 3,2 cm und in z-Richtung 7,9 cm. Hierbei sieht man deutlich, dass die Abweichung in z-Richtung wesentlich höher ist als in die anderen Richtungen. Dies lässt sich auch in Abbildung 5.4 gut erkennen. Folglich kann festgestellt werden, dass der PID-Regler für die Bewegung in z-Richtung noch nicht optimal konfiguriert ist, was man in späteren Arbeiten verbessern könnte. Es wurde für die x-Regler die gleichen Konfigurationen wie für die z-Regler benutzt, jedoch könnte durch leichte Asymmetrie oder eine schlechte Balance der Drohne eine andere Konfiguration gebraucht werden. In Abbildung 5.5 sieht man die einzelnen Flugabweichungen der 10 aufgenommenen Flüge. Hierbei kann beobachtet werden, dass die ersten Flüge eine deutlich höhere Abweichung haben als die Restlichen. Während den Flügen wurden allerdings keine Änderungen im System durchgeführt und auch keine Einstellungen in den PID-Reglern geändert. Dadurch erkennt man, wie ungleichmäßig sich die Drohne verhält. Zusätzlich wurde in jedem Flug die minimale und maximale Abweichung im Raum und in x-,y- und z-Richtung berechnet. Die Ergebnisse lassen sich in Tabelle 5.1 ablesen. Wichtig zu beachten ist, dass das Minimum in Millimeter aufgelistet ist. Hierbei bedeuten die Maximum-Werte, wie weit die Drohne im Durchschnitt auf der jeweiligen Achse oder im Raum maximal entfernt war. Anhand der Werte erkennt man, dass die Maximum-Werte auf der z-Achse in etwa doppelt so hoch ist wie die der anderen Achsen. Dies liegt vermutlich an den noch nicht optimal eingestellten z-Regler. Anhand der Minimum-Werte kann abgelesen werden, wie nah

Tabelle 5.1: Minimale und maximale Abweichungen beim Hovern.

Richtung	Maximum	Minimum
x	9.86 cm	0.02 mm
y	9.09 cm	4.21 mm
z	18.96 cm	0.12 mm
im Raum (x,y,z)	21.2 cm	27.4 mm

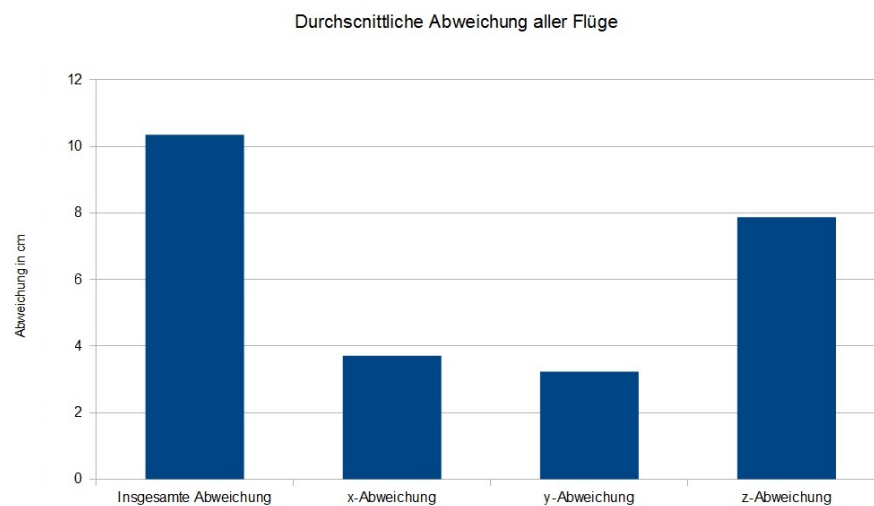


Abbildung 5.6: Dieses Schaubild zeigt sowohl die insgesamt berechnete Abweichung als auch die aller Achsen.

die Drohne ihren Set-Point gekommen ist. Auffällig ist, dass der Minimum-Wert im Raum deutlich höher als die der einzelnen Achsen ist. Dies liegt daran, dass wenn beispielsweise die Drohne in x-Richtung 0,02 mm an den SetPoint herangekommen ist, die Abweichungen auf der z- und y-Achse größer waren. Im Schnitt kam die Drohne ihrem SetPoint minimal 2,74 cm nah.

Im Hover-Modus wurde zusätzlich die Landung mit aufgenommen, welche genau an der Position stattgefunden hat, an der die Drohne auch gestartet ist. Berechnet wurde daher der durchschnittliche Abstand zwischen Startpunkt und Landepunkt. Dadurch kann beispielsweise abschätzen werden, in welchen Bereichen die Drohne landen könnte, wenn dies von Bedeutung ist. Berechnet wurde ein Durchschnitt von 16,1 cm, wobei 2,8 cm das Minimum aller Flüge darstellte und 48,4 cm das Maximum.

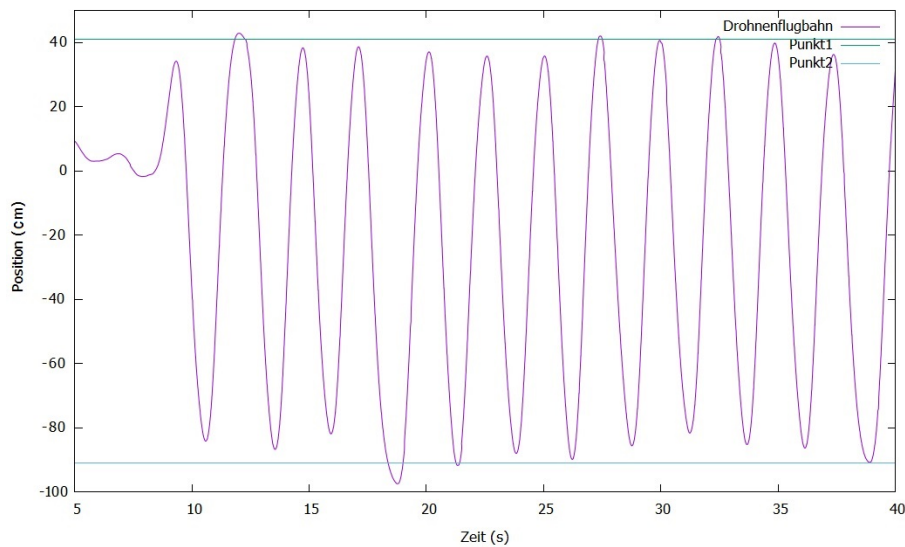


Abbildung 5.7: Flugbahn zwischen zwei Punkten in Abhängigkeit von der Zeit.

5.3.2 Route abfliegen

Neben dem Hovern und Landen wurden auch verschiedene Routen abgeflogen. Zum einen wurde zwischen zwei Punkten auf einer Achse hin und her geflogen, um dadurch zu ermitteln, welche Geschwindigkeit die Drohne fliegen kann. Zum anderen wurde eine Viereck-Route abgeflogen sowie eine Zickzack-Route, um die Fluggenauigkeit in schrägen Fluglagen zu untersuchen.

Abbildung 5.7 zeigt die Flugkurve in Abhängigkeit von der Zeit. Die Entfernung zwischen den beiden Punkten liegt bei ca. 1,3 m. In dem Schaubild erkennt man deutlich, dass die Drohne teilweise schon vor Erreichen des einen Punktes den anderen Punkt wieder angeflogen hat. Dies liegt an der Einstellung des Modus "Route abfliegen". Wie oben beschrieben, gilt ein Punkt als erreicht, wenn die Drohne sich diesen 15,5 cm nähert. Die exakte Position würde die Drohne nie oder eventuell nur nach einer sehr langen Zeit erreichen, da dass Optitrack im Millimeterbereich arbeitet. Um die Suchzeit der Drohne zu verringern, wurde der Radius des Punktes, der erreicht werden sollte, vergrößert. Zusätzlich kann man in der Abbildung 5.7 ablesen, dass die Drohne ca 2,5 Sekunden für eine Periode braucht. Somit ist die Maximalperiode der Drohne bei einer Amplitude von 1,3 m ca. 2,5 Sekunden. Allerdings wurde bei jeder getesteten Route, der Drohnen-Modus "Normal" benutzt. Mit dem Speed-Modus könnte die Drohne eventuell eine kleinere Frequenzzeit erreichen, dies wurde jedoch nicht getestet. Diese Informationen sind wichtig, wenn die Drohne beispielsweise Daten in Echtzeit präsentieren soll, um zu wissen, wie groß die Datenänderungen maximal sein dürfen.

Anschließend wurde eine Viereck-Route abgeflogen. Die Punkte des Vierecks lagen alle auf einer Ebene und die Seiten waren jeweils 1,4 m lang. Abbildung 5.8 zeigt die Flugbahn sowie den Start und die Landung. Zu beachten ist hierbei, dass die Landung diesmal nicht auf der

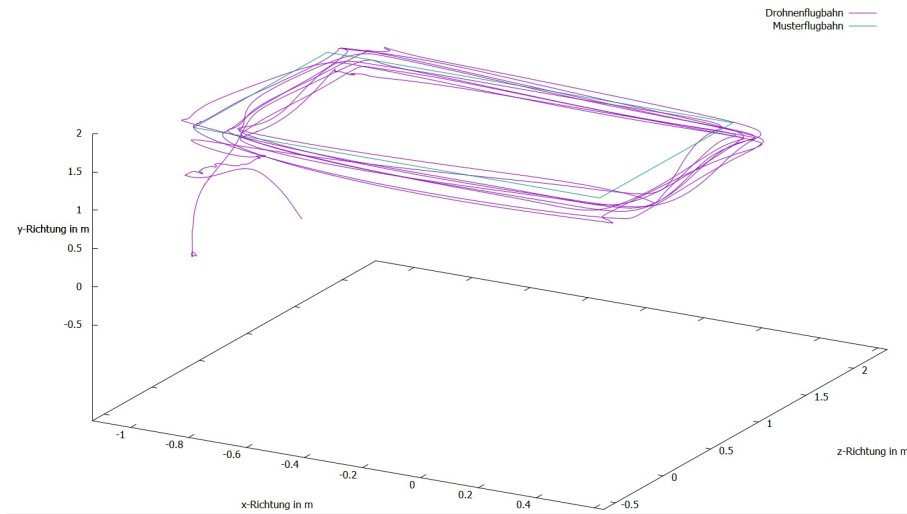


Abbildung 5.8: Flugbahn in einem Viereck.

Startposition stattfinden soll, sondern auf der Position, die die Drohne aktuell hatte, als der Landeknopf gedrückt wurde. Geflogen wurde im Uhrzeigersinn. Was man in der Abbildung zudem beobachten kann, ist, dass die Flugbahn in x-Richtung relativ parallel zur Musterflugbahn ist. In z-Richtung ist die Flugbahn weniger parallel. Dies liegt vermutlich daran, dass der z-Regler noch nicht optimal konfiguriert ist, wie vorhin beschrieben.

Die letzte Route, die abgeflogen wurde, lässt sich in Abbildung 5.9 betrachten. Insgesamt wurden 5 Punkte abgeflogen, die in ein Zickzack-Muster ergaben. Dadurch sollte die Stabilität der seitlichen Flüge untersucht werden. Wie man in der Abbildung erkennt, sind die seitlichen Flugbahnen relativ parallel zu der Musterflugbahn, wodurch man erkennt, dass die Drohne keine Probleme hatte diese abzufliegen. Wichtig ist, dass die Drohne in jedem Flug gleich gerichtet war, das heißt, sie hat sich in einer Kurve nicht gedreht, um in die nächste Richtung nach vorne fliegen zu können. Durchschnittlich kam die Drohne den Punkten minimal 9,9 cm nah.

Abschließend lässt sich über die Stabilität und Genauigkeit des Systems sagen, dass es in vielen Eigenschaften relativ zuverlässig und genau ist. Dennoch müssen die Aufgaben, die mit der Drohne erfüllen werden sollen, in Abhängigkeit der Genauigkeit meines Systems bestimmt werden. Möchte man die Drohne bestimmte Aufgaben erledigen lassen, die eine Genauigkeit von wenigen Millimetern erfordern, so ist dieses System dafür bisher noch nicht ausgereift genug. Werden jedoch nur ein paar Zentimeter Genauigkeit gefordert, so ist das System ausreichend.

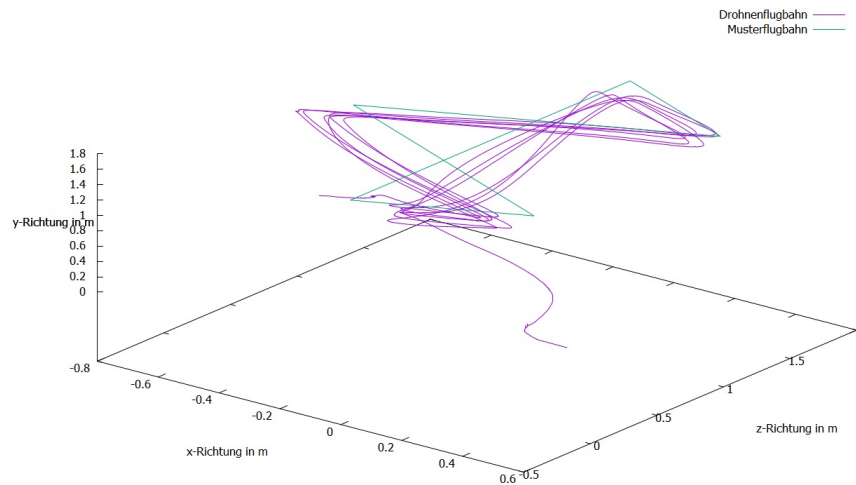


Abbildung 5.9: Flugbahn auf einer Zickzack-Route.

5.3.3 Zukünftige Arbeiten

Was man in späteren Arbeiten noch verbessern könnte, sind die Unregelmäßigkeiten der einzelnen Flüge und eventuell die bessere Konfiguration der PID-Regler. Dies kann erreicht werden, indem die aufgenommenen Daten analysiert werden und man anschließend die PID-Regler anhand der Analysen ändert. Dieser Vorgang wird sooft wiederholt, bis sich die Flugeigenschaft nicht weiter ändert. Sind die Flugeigenschaften anschließend immer noch nicht ausreichend, so könnten andere Drohnen benutzt und getestet werden, um ein noch besseres Flugverhalten zu erzielen.

6 Evaluation

Der Hauptbestandteil dieser Bachelorarbeit ist neben der Implementierung des Systems die Durchführung und Auswertung einer Studie. Dabei sollen die Hauptfunktionen des Systems mittels Probanden getestet werden. Durch die Studie sollen Erkenntnisse hervorgebracht werden, die Hinweise auf Verbesserungen und Akzeptanz des Systems geben. Zudem soll sie Informationen liefern, wie die einzelnen Funktionalitäten des Systems am angenehmsten und effizientesten benutzt werden und welche Aktionen mit diesen Interfaces noch verbinden werden könnten.

6.1 Studienaufbau

In der Studie mussten die Probanden vier verschiedene Funktionalitäten des Systems testen mit je unterschiedlichen Conditions. Nach jedem Durchlauf mussten sie einen kurzen Fragebogen ausfüllen. Die vier Funktionalitäten bestanden aus zwei Output- und zwei Input-Modi, die im Folgenden einzeln vorgestellt werden.

6.1.1 Funktionalität 1 Output: Impulse spüren

In diesem Modus ging es darum, wie deutlich Probanden Impulse einer Drohne spüren und wie angenehm diese sind. Zudem sollte hierbei getestet werden, wie deutlich die Probanden die Richtung des Impulses spüren. Während des gesamten Versuchs musste der Proband die Drohne in einer Hand halten. Nun kam in einem Abstand von zufälligen 2-4 Sekunden je ein Impuls, der 3-5 Sekunden anhielt. Ein Impuls bestand daraus, dass die Drohne mit vollem Schub entweder nach vorne, hinten, links oder rechts gesteuert wurde. Nachdem der Proband einen Impuls gespürt hatte, sollte er mit einem Presenter klicken, den er in der anderen Hand hielt und angeben, in welche Richtung er den Impuls spürt. Dadurch konnte gemessen werden, wie groß Verzögerung der einzelnen Impulse ist, ob der Impuls gespürt wurde und in welche Richtung der Impuls gespürt wurde. In jedem Durchgang kam jede Richtung des Impulses dreimal vor. Nach jedem Durchgang mussten die Probanden einen Fragebogen ausfüllen, in dem sie an einer 7-point Likert Scale angeben sollten, wie deutlich sie jeweils den Impuls nach vorne, hinten, links und rechts gespürt haben. Daraufhin mussten sie angeben, welche Richtung des Impulses sie am angenehmsten fanden und weshalb. Insgesamt musste der Proband diese Aufgabe zweimal durchführen mit jeweils unterschiedlichen Conditions. Bei

der einen Condition befand sich die Drohne im normalen Modus und in der anderen im Speed-Modus, der im vorherigen Kapitel beschrieben wurde. Nach Ausführung beider Durchläufe musste der Proband einen anschließenden Fragebogen ausfüllen, in dem gefragt wurde, wie deutlich er einen Unterschied zwischen den zwei Conditions wahrgenommen hat und welche davon er als angenehmer empfand und weshalb. Daraufhin konnte der Proband noch angeben, welche Aktionen man mit einem solchen User Interface erledigen könnte.

6.1.2 Funktionalität 2 Output: Impulsskalen erkennen

Im nächsten Modus wurde getestet, wie deutlich und angenehm die Impulserkennung ist, während man die Drohne manuell bewegt. Hierbei hielt der Proband die Drohne wieder in einer Hand und in der anderen einen Presenter. Anschließend bekam der Proband eine Richtung (vorne, hinten, links, rechts) zugesagt, in die er die Drohne bewegen sollte. Nach einem bestimmten Impulsradius gab die Drohne so lange einen Impuls von sich, wie sie außerhalb dieses Impulsradius war. Nachdem der Proband einen Impuls gespürt hatte, musste er mit dem Presenter klicken und anschließend die Drohne zu ihrer Ausgangsposition zurückbringen. Hat er keinen Impuls gespürt, so wurde er nach einer kurzen Zeit zurückgerufen. Wurde die Drohne nach vorne über den Impulsradius bewegt, so gab die Drohne einen Impuls nach hinten. Wurde sie nach rechts bewegt, gab sie einen Impuls nach links, für die anderen Seiten galt dem entsprechendes. In diesem Modus gab es zum einen zwei unterschiedliche Conditions, den normalen Drohnen-Modus und der Speed Drohnen-Modus. Zum anderen gab es zwei verschiedene Impulsradien, einer mit 30cm, der andere mit 50cm. Die Conditions wurden in allen Möglichkeiten zusammen kombiniert, wodurch insgesamt vier verschiedene Durchläufe entstanden. In jedem Durchlauf kam jede Richtung dreimal in unterschiedlicher Reihenfolge vor. Gemessen wurde, ob der Proband den Impuls spürt und wenn ja, in welchem Abstand zum Impulsradius. Nach jedem Durchgang musste der Proband wieder auf einer 7-point Likert Scale angeben, wie deutlich er die Richtungen gespürt hat, welche Richtung am angenehmsten war und weshalb. Nach allen vier Durchgängen musste ein letzter Fragebogen ausgefüllt werden, in dem gefragt wurde, welcher Impulsradius und welcher Drohnenmodus angenehmer war und welche Aktionen man sich vorstellen könnte, mit so einem User Interface durchzuführen.

6.1.3 Funktionalität 3 Input: Button press

Im ersten Input-Modus mussten sich die Probanden die Drohne als dreidimensionalen fliegenden Knopf vorstellen. In dieser Aufgabe bekamen die Probanden abwechselnd eine Richtung gesagt, in welche sie die Drohne drücken sollten. Die Richtungen bestanden aus vorne, hinten, links, rechts und oben. Den Probanden wurde dabei jedoch nicht erklärt, wie sie die Drohne drücken sollten oder an welcher Stelle. Sie sollten eine von sich aus angenehme Bewegung durchführen. In jedem Durchgang kam jede Richtung in einer zufälligen Reihenfolge dreimal vor. Nach jedem Drücken flog die Drohne zu ihrer Ausgangsposition zurück. Durch eine Videoanalyse wurde gemessen, wie und an welcher Stelle die Probanden die Drohne gedrückt

haben. Als Condition gab es drei unterschiedliche Höhen (1,20m, 1,40m, 1,60m) und pro Höhe einen Durchgang. Nach jeder Höhe mussten die Probanden einen Fragebogen ausfüllen, wie angenehm sie es fanden, die Drohne in die jeweilige Richtung zu drücken. Nach allen drei Durchgängen mussten die Probanden angeben, welche Höhe sie am angenehmsten fanden und weshalb. Anschließend wurden sie gebeten, sich Aktionen auszudenken, die man mit solch einem User Interface erledigen könnte.

6.1.4 Funktionalität 4 Input: Drohne neu positionieren

Im zweiten Input-Modus mussten die Probanden die Drohne nehmen und sie an eine neue Position bewegen. Insgesamt gab es neun verschiedene Positionen, die dem Probanden nacheinander zugerufen wurden. Es gab die Positionen oben-links, oben, oben-rechts, links, vorne, rechts, unten-links, unten und unten-rechts. Die Probanden wurden aufgefordert, die Positionen so zu wählen, dass es noch eine angenehme Bewegung ergibt. Nachdem die Drohne auf die neue Position bewegt wurde, flog sie auf ihre Ausgangsposition zurück und der Proband bekam eine neue Position zugesagt. Gemessen wurden hier die Koordinaten der Positionen abhängig von der Ausgangsposition und davon, wie die Probanden die Drohne bewegt und gegriffen haben. Anschließend mussten die Probanden erneut einen Fragebogen ausfüllen, in dem sie bei jeder Position angeben mussten, wie angenehm sie war. Daraufhin wurden sie wieder gefragt, was für eine Aktion sie sich vorstellen könnten, mit einem solchen User Interface durchzuführen.

Nachdem alle Modi vom Probanden durchgeführt wurden, musste er einen Abschlussfragebogen ausfüllen, der aussagen sollte, wie die Akzeptanz gegenüber dem System ist. Dabei wurde unter anderem gefragt, wie sinnvoll die Drohne in dem jeweiligen Modus war und welche Verbesserungsvorschläge sie haben. Abschließend wurden sie gefragt, wie sie die Idee von einer Drohne als Tangible User Interface im allgemeinen finden und welche Vor- und Nachteile dies hat.

6.2 Ergebnisse

Die Studie wurde mit 17 Probanden durchgeführt. Dabei waren 6 Probanden weiblich und 11 männlich. Fast alle Probanden waren Studenten und rechtshändig. Nur ein Proband war beidhändig. Die Durchschnittsgröße betrug 178cm. Jeder Proband musste auf einer Skala von 1 bis 5, wobei 1 unerfahren und 5 erfahren bedeutete, angeben, wie erfahren er im Umgang mit Drohnen ist. Fast alle Probanden waren eher unerfahren, weshalb ein Durchschnitt von 1,4 entstand mit einer Standardabweichung von 0,60. Im Folgenden werden die Ergebnisse der einzelnen Funktionalitäten vorgestellt.



Abbildung 6.1: Der häufigste Drohnen Griff.

6.2.1 Funktionalität 1 Output: Impulse spüren

In diesem Modus wurde ausgewertet, ob die Probanden die jeweiligen Impulse gespürt haben und falls ja, mit welcher Verzögerung. Zudem wurde durch eine Videoanalyse beobachtet, wie die Probanden die Drohne gegriffen haben. Zusätzlich wurde durch einen Fragebogen ermittelt, wie angenehm sie die jeweiligen Impulse fanden. Da es in diesem Modus zwei Conditions gab und somit jeder Proband den Modus zweimal durchführen musste, kamen insgesamt 34 Durchgänge zustande.

Den Probanden wurde nicht genau erklärt, wie sie die Drohne greifen sollen. Sie sollten sie intuitiv in die Hand nehmen und einen Griff finden, der für sie angenehm ist. In den meisten Fällen wurde die Drohne frontal gegriffen, was in Abbildung 6.1 zu sehen ist. In nur 5 Durchgängen wurde die Drohne von der Seite gegriffen. Dies ist entscheidend für die Auswertung der jeweiligen Impulse. Da die Probanden immer gleich zur Drohne standen, gingen für diejenigen, die die Drohne frontal gegriffen haben, die Impulse nach vorne von der Hand weg und die Impulse nach hinten zum Handrücken hin. Bei den Probanden, die die Drohne seitlich gegriffen haben, gingen diese Impulse nur in Richtung der entsprechenden Finger. Durch den frontalen Griff, konnte sich die Drohne zum einen mehr in und entgegen der Richtung Handrücken neigen, da sie an ihrer rechten und linken Seite gegriffen wurde. Zum anderen konnten keine großen Kippwinkel bei den Impulsen links und rechts entstehen. Bei den Probanden, die die Drohne seitlich gegriffen haben, galt jeweils das Gegenteil. Zusätzlich empfanden die Probanden in 10 Durchgängen teilweise die Impulse Vorne, wie nach unten, und die Impulse hinten

Tabelle 6.1: Durchschnittlichen Verzögerungszeiten und Fehler im ersten Input-Modus.

Impulsrichtung	Verzögerung / Standardabweichung		Fehler / Standardabweichung	
	normaler Modus	speed Modus	normaler Modus	speed Modus
Modus				
vorne	1.45 s / 0,79	1.14 s / 0,45	1.44 / 0,94	1 / 0,97
hinten	1.88 s / 0,72	1.41 s / 0,64	1.44 / 1,17	1.13 / 0,95
rechts	1.62 s / 0,73	1.47 s / 0,62	1.88 / 0,9	1.06 / 1,05
links	2.06 s / 0,81	1.42 s / 0,58	1.5 / 1	1.31 / 1,17
Durchschnitt	1.75	1.36	1.56	1.13

wie nach oben.

In der Tabelle 6.1 lassen sich die Verzögerungszeiten und die Fehlerquoten mit den dazugehörigen Standardabweichungen ablesen. Die Verzögerungszeiten kamen zustande, indem die Zeit zwischen dem Start des Impulses und dem Klicken des Probanden gemessen wurde. Hat ein Proband bei einem Impuls nicht geklickt, so wurde dies nicht mit in die Verzögerungszeit eingerechnet. Dies wurde jedoch als Fehler gewertet. Da jede Richtung im Modus dreimal vorkam, war eine maximale Fehlerquote von drei möglich. Anhand den Werten der Tabelle ist deutlich erkennbar, dass es einen Unterschied zwischen den Conditions gab. Durch den größeren Kippwinkel des Modus “speed“, war es den Probanden möglich, den Impuls schneller zu spüren als im normalen Modus. Zusätzlich wurden im “speed“ Modus weniger Fehler gemacht als im normalen Modus. Zwischen den einzelnen Richtungen lassen sich hingegen keine großen Unterschiede feststellen. Feststellen lässt sich jedoch, dass die Richtung vorne in jedem Modus am besten abschneidet. Betrachtet man die durchschnittliche Anzahl der Fehler, fällt auf, dass viele Impulse nicht gespürt wurden. Am wenigsten wurde der Impuls nach rechts im normalen Modus bemerkt, der in etwa 63% der Fälle nicht gespürt wurde. Das beste Ergebnis erzielte der Impuls nach vorne im speed Modus, der in nur ein drittel der Fälle nicht registriert wurde.

Um zu testen, ob die Probanden die Richtungen der Impulse auch so empfunden haben, wie sie geklickt haben, mussten sie zusätzlich nach jedem Durchgang auf einer 7-point Likert Scale angeben, wie deutlich sie den Impuls gespürt hatten. Abbildung 6.2 zeigt die Ergebnisse. Hierbei lassen sich zwischen den Modi wenig Unterschiede feststellen; die Richtungen vorne und hinten wurden jedoch am deutlichsten gespürt. Zusätzlich mussten die Probanden angeben, welchen Impuls sie am angenehmsten fanden und weshalb. Am häufigsten, insgesamt 14-mal, wurde die Richtung vorne ausgewählt. Am zweithäufigsten wurde mit 10-mal die Richtung rechts angegeben. Die Begründungen, weshalb dieser Impuls am angenehmsten war, waren meist die gleichen. Hauptsächlich lautete die Begründung, dass dieser Impuls am deutlichsten spürbar war. Zusätzlich äußerten sich ein paar Probanden dahingehend, dass es unangenehm gewesen sei, wenn die Drohne in ihre Richtung gedrückt hat. Ein Proband empfand die Richtung nach vorne unangenehm, da er Angst hatte, dass die Drohne ihm aus der Hand fliegt. Zusätzlich zur Durchführung des Modus äußerten sich einige Probanden zu Konstruktion des sich um die Drohne herum befindlichen Käfigs. Besonders die weiblichen Probanden meinten,

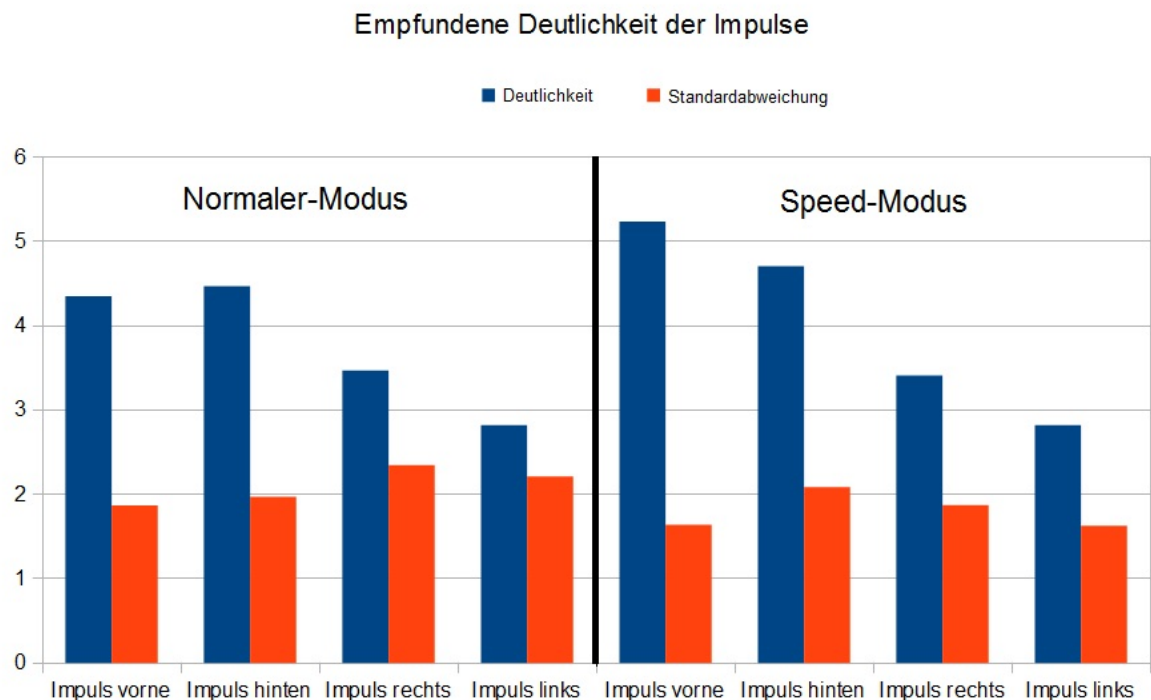


Abbildung 6.2: Von den Probanden angegebene Deutlichkeit der Impulse auf einer 7-point Likert Scale.

es sei unangenehm, die Drohne zu greifen, da der Käfig zu groß sei. Dies wurde auch in den Versuchen deutlich. Vier Probanden, ausschließlich weibliche mit kleinen Händen, hatten Probleme, die Konstruktion zu halten, und ließen sie während der Studie versehentlich los. Zusätzlich beschwerten sich einige Probanden über die Vibration des Käfigs und die Lautstärke der Drohne.

Abschließend gaben die Probanden auf einer 7-point Likert Scale an, wie deutlich sie den Unterschied zwischen den Conditions gespürt haben, welche sie angenehmer fanden und weshalb. Als Durchschnittswert ergab sich ein Wert von 4,5 mit einer Standardabweichung von 1,7. Die Antworten auf die Frage, welcher Modus angenehmer war, waren sehr ausgeglichen. Begründet wurde die jeweilige Antwort in den meisten Fällen entweder damit, dass die Impulse in diesem Modus besser erkennbar waren, oder damit, dass die Impulse in diesem Modus nicht so stark waren. Hieran erkennt man, dass die Probanden ein sehr unterschiedliches Wohlbefinden bezüglich der Stärke der Impulse haben. Auf die Frage, welche Aktionen die Probanden sich vorstellen könnten, mit einem solchen User Interface zu erledigen, gaben die Probanden unterschiedliche Antworten. Mehrere Probanden fanden, dass dieses User Interface als Bewegungshilfe, besonders für Blinde, dienen könnte. Ein anderer Proband meinte, es könnte der Rehabilitation von Unfallopfern dienen. Ein Proband konnte sich dieses User Interface als haptisches Feedback in Computerspielen vorstellen, insbesondere in Virtual

Tabelle 6.2: Distanz in Meter zwischen dem Impulsradius und an der Stelle, an der Proband den Impuls gespürt hat und die Standardabweichung (STD).

Bewegungsrichtung	normaler Modus		speed Modus	
	30cm / STD	50cm / STD	30cm / STD	50cm / STD
Radius				
vorne	0.27 / 0,18	0.22 / 0,16	0.4 / 0,25	0.23 / 0,12
hinten	0.4 / 0,31	0.19 / 0,15	0.28 / 0,21	0.24 / 0,2
rechts	0.35 / 0,19	0.24 / 0,23	0.35 / 0,26	0.2 / 0,15
links	0.35 / 0,26	0.19 / 0,12	0.33 / 0,24	0.18 / 0,12
Durchschnitt	0.34	0.21	0.34	0.21

Reality, wenn der Spieler ein haptisches Signal bekommen soll, beispielsweise einen Schlag auf den Körper.

Abschließend lässt sich über diesen Modus sagen, dass Menschen Impulse einer Drohne und deren Richtungen spüren können. Sollen damit Aktionen verbunden werden, sollte dies jedoch von der Verzögerungszeit abhängig gemacht werden. Die Studie hat gezeigt, dass die Probanden durchschnittlich länger als eine Sekunde gebraucht haben, um einen Impuls zu erkennen. Möchte man dem Benutzer indes Informationen zukommen lassen, die er innerhalb weniger Millisekunden registrieren soll, ist dieses User Interface noch nicht gänzlich ausgereift.

6.2.2 Funktionalität 2 Output: Impulsskalen erkennen

In diesem Modus wurde untersucht, nach welcher Entfernung die Probanden einen Impuls spürten und wie genau eine Impulsskala mit diesem User Interface kreiert werden könnte. Da die Drohne leider nach einigen Probanden technische Probleme hatte, konnten nur 11 der 17 Probanden einbezogen werden.

Fast alle Probanden griffen die Drohne wieder so, wie im ersten Modus beschrieben, siehe Abbildung 6.1. Bei der Bewegung der Drohne bewegten sich die meisten Probanden mit. Durch die Ergebnisse lässt sich erkennen, wie genau dem Probanden eine Skalagrenze durch haptische Informationen kenntlich gemacht werden kann. In Tabelle 6.2 sind die einzelnen berechneten Distanzen aufgelistet, die aussagen, nach welcher Distanz die Skalagrenze gespürt wurde. Zwischen den beiden Conditions (normaler und speed Modus) lassen sich im Durchschnitt keine Unterschiede erkennen. Zwischen den zwei Impulsradien lässt sich jedoch ein Unterschied feststellen. Die Impulse, die nach einem Radius von 50cm gesendet, wurden, in beiden Modi früher erkannt als die Impulse bei einem Impulsradius von 30cm. Zusätzlich mussten die Probanden angeben, wie deutlich sie die jeweiligen Impulse gespürt haben. In Abbildung 6.3 lässt sich die durchschnittlich empfundene Deutlichkeit ablesen. Im Durchschnitt wurde bei dem normalen Modus 30 cm 4,03, bei 50 cm 4,43 und bei dem speed Modus 30 cm 4,45, bei 50 cm 4,08 gewählt. Hierbei sieht man, dass sich die Werte zwischen den Impulsradien kaum ändern. Die Probanden haben also bei dem Impulsradius von 50 cm zwar in etwa die

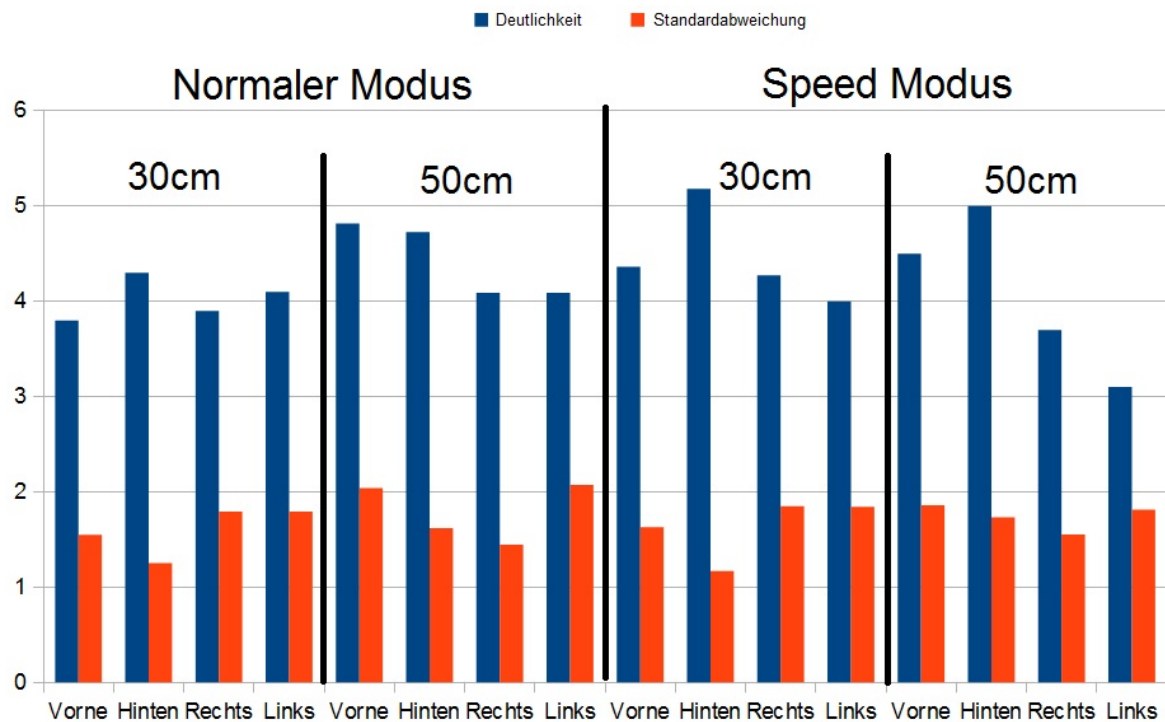


Abbildung 6.3: Deutlichkeit des zweiten Output-Modus auf einer 7-point Likert Scale und deren Standardabweichungen.

gleichen Impulse wie bei einem Impulsradius von 30 cm gespürt, sie haben sie jedoch früher wahrgenommen.

Die Ergebnisse auf die Frage, welche Richtung der Impulse für sie am angenehmsten war, lässt sich in Abbildung 6.4 ablesen. Die Begründungen, weshalb diese Richtung am angenehmsten war, war bei fast allen Probanden die gleiche. Sie gaben an, dass sie den Impuls in diese Richtung am stärksten und deutlichsten wahrgenommen haben. Bei der Frage, was für Aktionen mit einem solchen User Interface ausgeführt werden könnten, gaben viele Probanden dieselben Aktionen wie für den ersten Modus an. Die Drohne könnte als Hilfe für Sehbehinderte genutzt werden, in dem sie dem Benutzer einen Impuls gibt, wenn sich dieser einem Hindernis nähert. Nützlich könnte sie ferner bei der Steuerung von Geräten sein, die einen beschränkten Bewegungsradius haben.

Abschließend lässt sich über die Ergebnisse dieses Modus sagen, dass die Drohne als ein solches User Interface, welches dem Benutzer die Grenzen einer Skala deutlich machen kann, gut funktioniert. Allerdings muss bei Aktionen, die damit verbunden werden sollen, darauf geachtet werden, dass die Impulse erst nach einer kurzen Distanz und zudem nicht immer gespürt wurden. Spätere Arbeiten konnten versuchen, die Distanzen zu verringern, indem sie stärkere Drohnen benutzen.

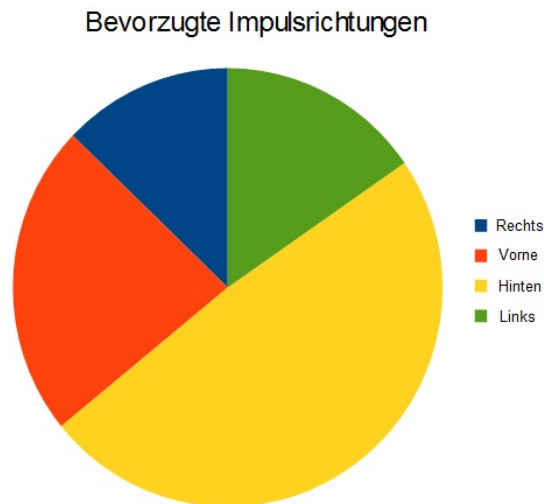


Abbildung 6.4: Die bevorzugte Impulsrichtung.

6.2.3 Funktionalität 3 Input: Button press

In diesem Modus wurde untersucht, wie Probanden die Drohne als Button benutzen. Dadurch soll ermittelt werden, wie eine Drohne am besten als solches User Interface (Drohne als button) benutzt werden könnte. Das könnte beispielsweise dadurch geschehen, dass diejenigen Aktionen die am häufigsten verwendet werden, mit der Richtung verbunden werden, die für die Benutzer am angenehmsten war.

Grundsätzlich konnte man zwei verschiedene Techniken beobachten, wie die Drohne gedrückt wurde. Zum einen gab es die "Tipp"-Technik, bei der die Probanden der Drohne nur einen kurzen Impuls gaben. Zum anderen wurde eine "Schiebe"-Technik beobachtet, bei der die Probanden die Drohne in die vorgegebene Richtung schoben. Aufgrund dessen, dass das System einen Klick in Abhängigkeit von der Geschwindigkeit der Drohne erkennt und die Drohne bei beiden Technik schnell genug bewegt wurde, wurde bei beiden in fast allen Fällen ein Klick erkannt. Abbildung 6.5 zeigt, wie ein Proband die Drohne nach vorne drückt. Dabei lässt sich gut erkennen, dass die Vorderseite des Käfigs mit der gesamten Handfläche gedrückt wurde. So interagierten die meisten Probanden mit der Drohne bei den Richtungen vorne, links und rechts. Ab und zu konnte man auch beobachten, dass die Probanden den Käfig nur an mittlerer Stelle der Vorderseite berührt haben, da sie merkten, dass die Drohne leicht kippt, wenn man sie nur unten berührt. Lediglich bei den Richtungen hinten und oben gab es verschiedene Techniken, die die Probanden benutzten. Bei der Richtung hinten wurde die Drohne meistens mit dem Arm umringt und auf die Hinterseite des Käfigs getippt oder gedrückt. Manche Probanden nahmen die Drohne hingegen mit zwei Händen an den Seiten und schoben sie zu sich hin. Andere griffen die Drohne an der Vorderseite und zogen sie zu sich her. Bei der Richtung oben drückten die meisten Probanden die Drohne mit der vollen Handfläche an der Unterseite des Käfigs nach oben. Da die Drohne durch den Auftrieb der Rotoren jedoch selbstständig



Abbildung 6.5: Proband der die Drohne nach vorne drückt.

Tabelle 6.3: Wohlbefinden der Probanden, die Drohne in die entsprechende Richtung zu drücken, auf einer 7-point Likert Scale und deren Standardabweichungen.

Richtung	1.20m / STD	1.40m / STD	1.60m / STD	Durchschnitt
vorne	5.06 / 1,78	5.47 / 1,37	5.71 / 1,31	5.41
hinten	4.35 / 1,69	3.88 / 1,93	4 / 1,94	4.08
links	5.41 / 1,12	5.47 / 1,18	5.24 / 1,30	5.37
rechts	5.35 / 1,17	5.29 / 0,99	4.94 / 1,43	5.2
oben	4.63 / 1,86	4.71 / 1,79	4.35 / 2,12	4.56
Durchschnitt	4.96	4.96	4.85	

nach oben fliegt, wenn die Hand zu nah an die Unterseite der Drohne kommt, mussten die Probanden bei dieser Bewegung schnell sein. Einige Probanden nahmen die Drohne hingegen mit beiden Händen an den Seiten und drückten sie nach oben. In vereinzelt Fällen stürzte die Drohne durch die Interaktion mit ihr leider ab. Dies passierte, wenn die Drohne zu langsam und gleichzeitig nicht in der Mitte, sondern am Rand der Unterseite gedrückt wurde. Außerdem stürzte sie ab, wenn der Proband die Drohne bei einer seitlichen Bewegung zu sehr im unteren Bereich gedrückt hat. Dadurch kippte sie um und schaffte es nicht, sich wieder zu regeln.

An der Tabelle 6.3 lässt sich ablesen, wie angenehm die Probanden die jeweiligen Bewegungen fanden auf einer 7-point Likert Scale. Hierbei kann man deutlich erkennen, dass die Richtung nach hinten für die Probanden am wenigsten angenehm war. Viele Probanden erschrecken, als

die Drohne auf sie zu flog, nachdem sie diese zu sich gedrückt hatten. Dies und die Tatsache, dass die Bewegung hierbei am größten ist, könnten Faktoren sein, weshalb diese Richtung als unangenehmsten empfunden wurde. Die Richtung oben hat gegenüber den sonstigen anderen Richtungen ebenfalls schlechter abgeschnitten. Einige Probanden fanden den Windstoß der Drohne unangenehm oder hatten Angst, dass sie in die Drohne greifen würden, da der Käfig unten offen war. Zwischen den einzelnen Höhen lassen sich keine großen Unterschiede erkennen. Dennoch wurden die Probanden gefragt, welche Höhe sie weshalb am angenehmsten fanden. 9 Probanden wählten die Höhe von 1,40m mit der Begründung, dass die Drohne dort am einfachsten und natürlichsten zu bewegen sei und dass sie einem nicht direkt vor dem Gesicht herumgeflogen ist. 7 Probanden fanden demgegenüber die Höhe von 1,60m am angenehmsten. Dies begründeten ein paar Probanden damit, dass sie es angenehm fanden, dass sich die Drohne auf Augenhöhe befand. Ein Proband fand es bei dieser Höhe angenehmer, die Drohne nach hinten zu bewegen. Nur ein Proband wählte die Höhe von 1,20m. Anschließend beschrieben die Probanden, welche Aktionen sie sich vorstellen könnten, mit einem solchen User Interface zu erledigen. Einige Probanden konnten sich vorstellen, dieses User Interface für verschiedene Alltagssituationen zu benutzen, wie zum Beispiel das Regeln von Geräten in der Wohnung als eine Art Universalfernbedienung. Zudem meinte ein Proband, dass dieses User Interface mit seinem Handy verbunden werden könnte, um damit die Aktionen zu verbinden, wie beispielsweise das Liken auf Facebook.

Abschließend lässt sich über diesen Modus als User Interface sagen, dass sich die Drohne angenehm klicken lässt. Den meisten Probanden machte diese Aufgabe sehr viel Spaß. Sie empfanden die Interaktion mit der Drohne als spielerisch. Daher hat diese Interaktionsart ein großes Potential und sollte weiter getestet werden, indem zum Beispiel Geräte damit steuert werden.

6.2.4 Funktionalität 4 Input: Drohne neu positionieren

In diesem Modus wurde untersucht, wie die Probanden die Drohne neu positionierten. Dadurch soll herausgefunden werden, auf welche Positionen man die Drohne am besten bewegen sollte, wenn diese mit Daten verbunden ist. Wenn beispielsweise die Lautstärke des Fernsehers durch eine Bewegung mit der Drohne gesteuert werden soll, sollte dies mit einer angenehmen Bewegung geschehen.

In fast allen Fällen wurde die Drohne wieder frontal gegriffen. In Abbildung 6.6 kann man erkennen, wie ein Proband die Drohne an die Position oben-rechts setzt. Hierbei lässt sich feststellen, dass die Position, die gewählt wurde, einem Abstand von durchschnittlich 58 cm entspricht, was in etwa die Strecke eines gestreckten Armes ist. Diese Position benutzte fast jeder Proband, abhängig von seiner Armlänge. Insgesamt gab es neun verschiedene Positionen, die in der Grafik 6.8 mit ihrer jeweiligen Akzeptanz aufgelistet sind. Dabei kann beobachtet werden, dass die Positionen auf der linken Seite am unangenehmsten für die Probanden waren. Da fast alle Probanden rechtshändig waren, griffen sie die Drohne meistens mit der rechten Hand und brachten sie auf die linken Positionen. Dadurch entstand eine unangenehme Bewegung. Bis auf diese und die Position unten wurden alle Positionen als angenehm eingestuft.



Abbildung 6.6: Ein Proband setzt die Drohne an der Position oben-rechts ab.

Im Schaubild 6.7 lassen sich die Distanzen und deren Standardabweichungen ablesen, an denen die Drohne hingesezt wurde. Die Distanz bildete sich durch den euklidischen Abstand zwischen der Position, in der die Drohne vor dem Probanden flog und der Position, an der sie schließlich abgesetzt wurde. Die Distanz, in der die Drohne nach vorne gesetzt wurde beträgt im Durchschnitt 0,49 m, bei einer Standardabweichung von 0,14. Anschließend mussten die Probanden wiederum angeben, welche Aktionen sie sich vorstellen könnten, mit einem solchen User Interface zu erledigen. Hier antworteten einige Probanden erneut, dass damit Alltagsgegenstände im Haushalt gesteuert werden könnten. Andere Probanden könnten sich dieses User Interface in Verbindung mit Projektionen vorstellen, in denen sich abgebildete Objekte bewegen, indem die Drohne bewegt wird.

6.2.5 Allgemeine Akzeptanz gegenüber der Drohne als Tangible User Interface

Zuletzt mussten die Probanden einen Abschlussfragebogen ausfüllen, in dem sie die Drohne allgemein als Tangible User Interface bewerten und Vor- und Nachteile ansprechen sollten. Zunächst mussten sie jeweils auf einer Skala von 1 bis 5 angeben, wie sinnvoll sie die Drohne in den beiden Output-Modi fanden. Der “Impulse spüren“-Modus wurde durchschnittlich mit einer 2,8, bei einer Standardabweichung von 1,35, bewertet und der “Impulsskalen erkennen“-Modus mit einer 3,2, bei einer Standardabweichung von 1,14. Einige Probanden meinten hierbei, dass ihnen keine konkreten Anwendungen einfielen, bei denen solch ein User Interface von

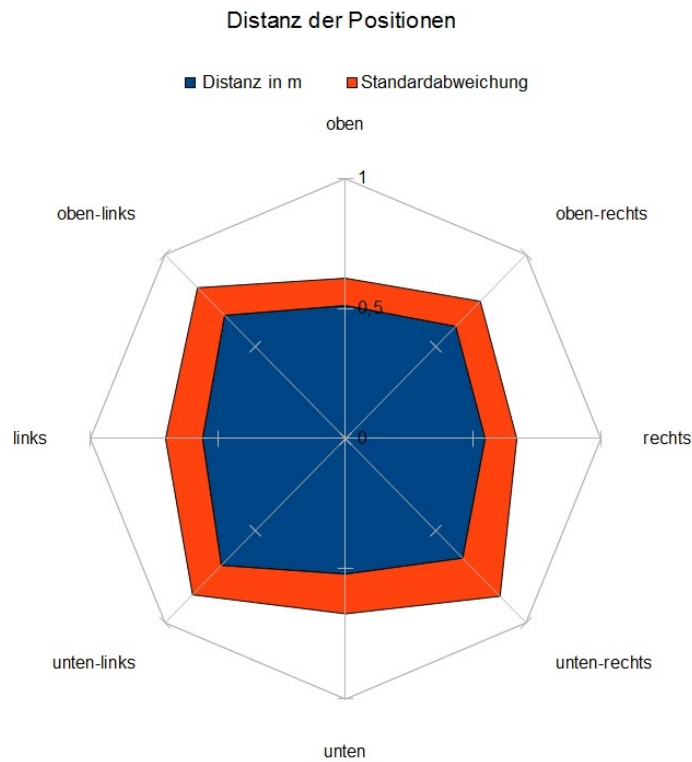


Abbildung 6.7: Distanz der Positionen und deren Standardabweichung.

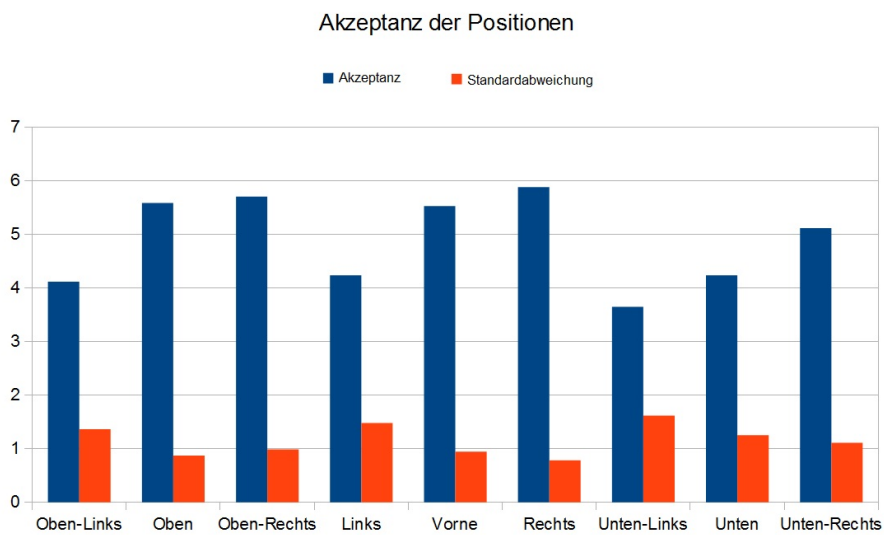


Abbildung 6.8: Akzeptanz der verschiedenen Positionen.

Nutzen wäre. Anschließend wurden die Probanden gefragt, ob sie Verbesserungsvorschläge zu den beiden Modi haben. Hierbei meinten die meisten Probanden, es sollten stärkere und deutlichere Impulse verwenden werden. Ein Proband führte an, ein kurzer, dafür starker Impuls wäre besser als ein langer aber schwächerer Impuls. Ob die Impulse dadurch besser und angenehmer wahrgenommen werden, könnte in späteren Arbeiten untersucht werden. P6, P12, P14 und P15 kritisierten die Lautstärke der Drohne. Zusätzlich wurden die Probanden gefragt, auf welche Art eine Drohne zudem als Output-Gerät dienen könnte. Hierzu meinte ein Proband, dass die Drohne dem Benutzer Informationen zukommen lassen könnte, indem die Drohne in eine bestimmte Richtung wippt oder dem Benutzer durch LEDs Lichtsignale zukommen lässt.

Daraufhin mussten die Probanden dieselben Fragen, zu den Modi "Button press" und "Drohne neu positionieren" beantworten. Der "button press"-Modus wurde durchschnittlich mit einer 3,7, bei einer Standardabweichung von 0,92, bewertet und der letzte Modus mit 4,4, bei einer Standardabweichung von 0,87. Hier fielen die Beurteilungen besser aus als in den Output-Modi, nicht zuletzt deshalb, da einige Probanden der Auffassung waren, sie könnten sich in diesen Bereichen die Drohne als hilfreich vorstellen. Verbesserungsvorschläge gab es in diesen Modi nur wenige. Ein Proband hätte sich bei dem Modus "Button press" ein stärkeres haptisches Feedback gewünscht, durch das die Drohne dem Benutzer einen größeren Widerstand zukommen lässt. Ein anderer Proband beschwerte sich darüber, dass die Drohne zu nah auf ihn zugeflogen sei.

Abschließend mussten die Probanden auf einer 7-point Likert Scale angeben, wie sie die Idee, eine Drohne als Tangible User Interface zu nutzen, generell finden. Hier ergab sich im Durchschnitt ein Wert von 5,2, bei einer Standardabweichung von 1,64, was einen sehr guten Wert darstellt. Es konnte sich also der Großteil der Probanden vorstellen, eine Drohne als solches User Interface zu benutzen. Zusätzlich wurden die Probanden nach Vor- und Nachteilen zu dieser Idee gefragt. Viele Probanden sehen einen Vorteil darin, dass die Interaktion mit der Drohne sehr intuitiv ist und sie eine gute Kontrolle über diese haben, da sie mit einem physischen Gegenstand interagieren, den sie selbst bewegen können. Einen großen Vorteil sahen einige Probanden darin, dass Blinde dieses User Interface gut bedienen können, da es zum Benutzer geflogen kommt und dieser nicht nach ihm suchen muss. Nachteile sahen die Probanden bei der Konstruktion des Käfigs. Einigen Probanden war es unangenehm und fiel es schwer, den Käfig zu greifen, da er für kleine Hände zu groß ist. Ein Proband hatte zudem Angst, in die Rotoren zu greifen. Zusätzlich wurden die Geräusche der Drohne als zu laut empfunden. P5, P7 und P14 kritisierten die Akkulaufzeit der Drohne, da es ab und zu vorkam, dass der Akku während der Studie leer wurde. Ein Proband war der Auffassung, dass ein solches User Interface die Unselbstständigkeit des Menschen weiter fördert und die Abhängigkeit gegenüber der Technik dadurch zu groß werden würde. Wie man sieht, bezieht sich die meiste Kritik auf Hardware-Probleme, die im Laufe der Zeit noch verbessert werden können. Insbesondere sollte den Benutzern in Zukunft ein kleiner Käfig oder gar eine andere Konstruktion angeboten werden, die sich besser zum Greifen eignen.

6.3 Zusammenfassung und Diskussion

Aus den Ergebnissen der Studie, lässt sich zeigen, dass die Drohne dem Benutzer haptische Informationen übermitteln kann. Zudem wurde gezeigt, dass die Drohne den Benutzern eine Skalagrenze durch Impulse deutlich machen kann. Bei einer Skala von 30 cm wurde der Impuls nach einer größeren Distanz gespürt, wie bei einer 50 cm Skala. Anschließend konnte gezeigt werden, wie Probanden die Drohne als fliegenden Button benutzen. Dabei wurde festgestellt, dass es am angenehmsten war, die Drohne in die Richtungen, vorne, links und rechts zu drücken. Daraufhin wurde die Drohne auf verschiedene Positionen bewegt, um zu erfahren, welche Positionen für die Benutzer am angenehmsten ist, um dadurch ein bestmögliches Tangible User Interface zu erzeugen. Die Analyse der Daten ergab, dass die Positionen auf der rechten Seite des Probanden als angenehmsten empfunden wurde.

Die Studie zeigt, welche Eigenschaften der Drohne verbessert werden müssen und welche Interaktionen mit der Drohne dem Benutzer am angenehmsten erscheinen. Zum einen könnte die Verzögerungszeit, die die Benutzer brauchen bis sie einen Impuls spüren, durch eine stärkere Drohne verkleinert werden. Da aber einigen Probanden die Impulse bereits bei der benutzten Drohne zu stark waren, müsste auf kürzere aber deutlichere Impulse umgestellt werden. Zudem sollte jeder Benutzer die Impulsstärke selbst einstellen können, um dadurch ein bestmögliches Wohlbefinden zu erreichen. In Hinblick auf die Funktionalität "Button press" sollten für die wichtigsten und meist genutzten Aktionen, die man mit den Drohnen-Klicks ausführen möchte, die Klick-Richtung links, rechts und nach vorne gewählt werden. Die Höhe sollte am besten vom jeweiligen Benutzer manuell eingestellt werden können. In der Studie wurde gezeigt, dass die angenehmsten Positionen, an der die Drohne positioniert werden sollte, auf der rechten Seite lagen. Dies liegt vermutlich daran, dass kein Proband Linkshänder war. Für Linkshänder wären die Positionen auf der linken Seite vermutlich angenehmer. Daher sollte diese Interaktion mit der Drohne davon abhängig gemacht werden, ob der Proband Links- oder Rechtshänder ist, um dadurch ein bestmögliches Wohlbefinden für den Benutzer zu erreichen. Durch die gewonnen Informationen und Daten, lassen sich Programme entwickeln, die die Drohne als ein angenehmes und sinnvolles Tangible User Interface einsetzen.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

In dieser Bachelorarbeit wurde ein System entwickelt, das eine Drohne als Tangible User Interface benutzbar macht. Es wurde eine Regelung implementiert, welche die Koordinaten der Drohne über das Optitrack erhält und anschließend eine Ausgabe berechnet, damit sich die Drohne an bestimmten Punkten im Raum halten oder zu ihnen fliegen kann. Durch eine Selbststudie wurde ermittelt, mit welchen Abweichungen die Drohne an einer Position hovert. Dieser Wert beträgt etwa 10,3 cm, was zwar kleine Schwankungen bedeutet, aber für die vorgestellten Use-Cases völlig ausreichend ist. Anschließend wurden vier verschiedene Interaktionsfunktionalitäten vorgestellt, welche als Voraussetzung eine Konstruktion um die Drohne herum bedürfen, damit diese gut gegriffen werden kann. In der Studie wurden die vier implementierten Funktionalitäten durch Probanden getestet und ausgewertet. Die Ergebnisse zeigen, dass eine Drohne als Tangible User Interface sehr gut einsetzbar ist. Sie konnte dem Benutzer ein haptisches Feedback geben, das durchschnittlich nach 1,45s bis 2,06s wahrgenommen wurde. Zusätzlich konnte gezeigt werden, wie die Probanden am besten mit der Drohne interagierten. Durch eine Befragung der Probanden konnte ermittelt werden, dass sich die meisten Probanden eine Drohne als Tangible User Interface in Zukunft vorstellen könnten. Im Ergebnis lässt sich daher festhalten, dass Drohnen, insbesondere Quadrocopter, als Tangible User Interface ein sehr großes Potential haben.

7.2 Herausforderungen

Während der Entwicklung des Systems gab es einige Probleme und Herausforderungen. Zu Beginn war es problematisch und sehr zeitintensiv, die richtigen Konfigurationen der PID-Regler zu finden. Es mussten viele Testflüge gemacht werden, bis die Flugstabilität ausreichend war. Anschließend gab es Komplikationen bei der Konstruktion des Käfigs. Das Hauptproblem hierbei war, ein Material zu finden, das leicht genug war, von der Drohne getragen zu werden und dennoch stabil genug, um von den Griffen der Benutzer nicht verformt zu werden. Im Rahmen der vorliegenden Arbeit wurde Pappe benutzt, allerdings verformte sich die Konstruktion leicht, nachdem sie von den Benutzern gegriffen wurde. Dadurch verschoben sich die Distanzen zwischen den Markern, weshalb das Optitrack-System ab und zu Probleme hatte, die Drohne zu tracken. Durch das Gewicht des Kastens musste die benutzte Drohne die

Propeller fast mit voller Leistung ansteuern, um sich auf einer Höhe halten zu können. Dadurch konnte die angegebene Flugdauer von 5 bis 7 Minuten nicht erreicht werden. Stattdessen war der Akku innerhalb von 2 bis 3 Minuten leer, was in der durchgeführten Studie teilweise problematisch war. Deshalb mussten die Aufgaben der Studie so aufgeteilt und aufgebaut sein, dass die benötigten Zeiten innerhalb der Akkulaufzeit lagen. Gesamt betrachtet, konnte zu jedem Problem eine Lösung gefunden werden, die jedoch in manchen Fällen noch verbessert werden könnte.

7.3 Ausblick

Herkömmliche Tangible User Interfaces haben den Nachteil, dass sie passive physische Objekte sind, die meistens auf eine Ebene oder Achse beschränkt sind. Durch die Fortentwicklung und die verbesserten Flugeigenschaften der Quadrocopter bietet es sich an, solche als Tangible User Interfaces zu benutzen. Dies liegt nicht nur daran, dass sie dreidimensional im Raum frei positionierbar sind, sondern auch, dass sie sich selbstständig bewegen und dem Benutzer haptische Feedbacks geben können. Deshalb wurde in dieser Arbeit ein System entwickelt, um dies durch Benutzer zu testen. Im Rahmen der durchgeführten Studie wurden einige Ideen gesammelt, in welchen Situationen dieses System später benutzt werden könnte. Ein sehr gutes Beispiel ist etwa die Nutzung der Drohne als Tangible User Interface als Ersatz für einen Blindenhund. Blindenhunde müssen lange ausgebildet werden und können den blinden Benutzer lediglich an Stellen bringen, zu denen er die Route kennt. Dass Menschen Richtungsimpulse durch einer Drohne spüren können, wurde in dieser Arbeit aufgezeigt. Dadurch könnte der Blinde von der Drohne über haptische Informationen geführt werden. Ein Vorteil hierbei wäre, dass die Drohne die Route nicht kennen muss, durch GPS könnte die Drohne den Blinden an jede beliebige Stelle bringen. Zusätzlich hätte dieses System einen Vorteil für Gehbehinderte. Da die Drohne autonom zu einem Benutzer fliegen kann, müsste sich dieser nicht mühselig zum Interaktionsgerät bewegen und könnte damit diverse Haushaltsgegenstände steuern.

In späteren Forschungsarbeiten könnten die in dieser Arbeit angesprochenen Kritikpunkte verbessert werden. Zum einen könnte versucht werden, die Abweichung beim hovern weiter zu verringern, indem die PID-Regler noch besser konfiguriert werden. Zum anderen sollte das Gerüst aus einem weniger verformbaren Material konstruiert werden und gegebenenfalls eine stärkere Drohne benutzt werden, damit diese schwereres Material tragen kann. Zusätzlich könnte eine stärkere Drohne auch deutlichere Impulse geben, damit die Probanden einen Impuls früher und öfter spüren. Anschließend sollte das System mit richtigen Szenarien in Verbindung gebracht und getestet werden. Dadurch kann untersucht werden, wie die Drohne als Tangible User Interface bei Alltagsproblemen benutzt werden kann. Abschließend lässt sich als Ergebnisse dieser Arbeit festhalten, dass Drohnen und dabei insbesondere Quadrocopter als Tangible User Interfaces großes Potential haben und in vielerlei Hinsicht noch weiter erforscht werden sollten.

Literaturverzeichnis

- [1] —, „Computer-augmented environments: Back to the real world“, *Communications of the ACM*, Bd. 36, Nr. 7, S. 24–27, 1993.
- [2] —, „Tangible user interfaces: Past, present, and future directions“, *Found. Trends Hum.-Comput. Interact.*, Bd. 3, Nr. 1–2, S. 1–137, Jan. 2010. Adresse: <http://dx.doi.org/10.1561/11000000026>.
- [3] —, „Tangible bits: Towards seamless interfaces between people, bits and atoms“, in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Ser. CHI '97, Atlanta, Georgia, USA: ACM, 1997, S. 234–241. Adresse: <http://doi.acm.org/10.1145/258549.258715>.
- [4] —, „Inform: Dynamic physical affordances and constraints through shape and object actuation.“, in *UIST*, Bd. 13, 2013, S. 417–426.
- [5] —, „Diri - the actuated helium balloon: A study of autonomous behaviour in interfaces“, in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Ser. UbiComp '15, Osaka, Japan: ACM, 2015, S. 349–360. Adresse: <http://doi.acm.org/10.1145/2750858.2805825>.
- [6] —, *Designing sociable robots*. MIT press, 2004.
- [7] —, „Drone & me: An exploration into natural human-drone interaction“, in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Ser. UbiComp '15, Osaka, Japan: ACM, 2015, S. 361–365. Adresse: <http://doi.acm.org/10.1145/2750858.2805823>.
- [8] —, „The metadesk: Models and prototypes for tangible user interfaces“, in *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, Ser. UIST '97, Banff, Alberta, Canada: ACM, 1997, S. 223–232. Adresse: <http://doi.acm.org/10.1145/263407.263551>.
- [9] —, „Tangible bits: Beyond pixels“, in *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction*, Ser. TEI '08, Bonn, Germany: ACM, 2008, S. xv–xxv. Adresse: <http://doi.acm.org/10.1145/1347390.1347392>.
- [10] —, „Tangible interfaces for remote collaboration and communication“, in *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, Ser. CSCW '98, Seattle, Washington, USA: ACM, 1998, S. 169–178. Adresse: <http://doi.acm.org/10.1145/289444.289491>.

- [11] —, „Groupware: Some issues and experiences“, *Commun. ACM*, Bd. 34, Nr. 1, S. 39–58, Jan. 1991. Adresse: <http://doi.acm.org/10.1145/99977.99987>.
- [12] —, „A design space for ephemeral user interfaces“, in *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, Ser. TEI '13, Barcelona, Spain: ACM, 2013, S. 75–82. Adresse: <http://doi.acm.org/10.1145/2460625.2460637>.
- [13] —, „The reactable*: A collaborative musical instrument“, in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on*, IEEE, 2006, S. 406–411.
- [14] —, „Midair displays: Exploring the concept of free-floating public displays“, in *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, Ser. CHI EA '14, Toronto, Ontario, Canada: ACM, 2014, S. 2035–2040. Adresse: <http://doi.acm.org/10.1145/2559206.2581190>.
- [15] —, „A flying pantograph: Interleaving expressivity of human and machine“, in *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, Ser. TEI '16, Eindhoven, Netherlands: ACM, 2016, S. 653–657. Adresse: <http://doi.acm.org/10.1145/2839462.2856347>.
- [16] —, „L'evolved: Autonomous and ubiquitous utilities as smart agents“, in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Ser. UbiComp '15, Osaka, Japan: ACM, 2015, S. 487–491. Adresse: <http://doi.acm.org/10.1145/2750858.2805850>.
- [17] —, „Dronenavigator: Using drones for navigating visually impaired persons“, in *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, Ser. ASSETS '15, Lisbon, Portugal: ACM, 2015, S. 327–328. Adresse: <http://doi.acm.org/10.1145/2700648.2811362>.
- [18] —, „Bitdrones: Towards levitating programmable matter using interactive 3d quadcopter displays“, in *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, Ser. UIST '15 Adjunct, Daegu, Kyungpook, Republic of Korea: ACM, 2015, S. 57–58. Adresse: <http://doi.acm.org/10.1145/2815585.2817810>.
- [19] *Drohnen werden immer beliebter*. Adresse: <http://www1.wdr.de/nachrichten/ruhrgebiet/drohnen-werden-immer-beliebter-100.html>.
- [20] *Quadcopter*. Adresse: <https://de.wikipedia.org/wiki/Quadrocopter#/media/File:Quadrocopter-X-H-Konfiguration.gif>.
- [21] Adresse: https://upload.wikimedia.org/wikipedia/commons/b/b3/Einfacher_Regelkreis_n.svg.
- [22] *Control theory terminology*. Adresse: <http://www.newport.com/Control-Theory-Terminology/178319/1033/content.aspx>.
- [23] *Pid controller design*. Adresse: <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlPID#7>.

[24] Adresse: <https://www.optitrack.com/about/>.

Alle URLs wurden zuletzt am 22.05.2016 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift