

Schlussbericht

zu IFL-Vorhaben Nr. 6304100

Thema

AIRKom - Einsatz autonomer Flugroboter für den vollautomatisierten Materialtransport zwischen Lagerplatz und Kommissionierplätzen

Berichtszeitraum

01.05.2019 – 31.10.2021

Forschungsvereinigung

Forschungsgemeinschaft Intralogistik/Fördertechnik und Logistiksysteme (IFL) e.V.

Forschungseinrichtung(en)

Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik (FAPS)

AIRKom

IFL-Vorhaben Nr. 6304100

Einsatz autonomer Flugroboter für den vollautomatisierten Materialtransport zwischen Lagerplatz und Kommissionierplätzen

Schlussbericht

Kurzzusammenfassung:

Autonome Flugroboter bieten aufgrund ihrer dreidimensionalen Bewegungsmöglichkeit und den im Vergleich zum flurgebundenen Transport kurzen Transportstrecken und den erzielbaren Transportgeschwindigkeiten weitreichende Einsatzmöglichkeiten. Die Verlagerung von Transportprozessen in die Luft ermöglicht zudem eine Reduktion der am Boden für Logistikprozesse benötigten Fläche, sodass die Flächenproduktivität gesteigert wird.

Im Rahmen des Forschungsprojektes AIRKom wurde untersucht, wie der vollautomatisierte Materialtransport mit autonomen Multicoptern konzipiert und umgesetzt werden kann. Es wurde ein Demonstrator realisiert, mit dem das Konzept erfolgreich experimentell validiert wurde. Forschungsschwerpunkte des Projektes waren die Lokalisierung des Flugroboters in Innenbereichen, die Absicherung, Kollisionsvermeidung und Navigation sowie die vollautomatisierte Handhabung von Kleinladungsträgern durch den Flugroboter zum Materialtransport.

Das Ziel des Forschungsprojektes ist erreicht worden.

Umfang des Berichtes: 67 Seiten, 35 Abbildungen, 81 Literaturquellen
Laufzeit: 01.05.2019 bis 31.10.2021
Fördermittelgeber: Forschungsgemeinschaft Intralogistik/Fördertechnik und Logistiksysteme (IFL) e.V.
Forschungsstelle: Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik (FAPS)
Projektleiter: Prof. Dr.-Ing. Jörg Franke

Bearbeiter und Verfasser: Christian Hofmann, M. Sc.
Markus Lieret, M. Sc.
Prof. Dr.-Ing. Jörg Franke

Studentische Arbeiten im Rahmen des Projektes:

U. Ali Analyse und Optimierung von Lokalisierungsalgorithmen für Flugroboter

M. Fichtner Automatisierte Erstellung und Aktualisierung von semantischen Karten mithilfe von autonomen Flugrobotern

P. Gall Entwicklung und Analyse eines Ansatzes zur Navigation von mobilen Robotern mit semantischen Karten

Z. Lin Entwicklung und Analyse eines Ansatzes zur Erkennung und Poseschätzung von Kleinladungsträgern mithilfe von Kameradaten

F. Metzler Optimierung von Multikoptern hinsichtlich Sicherheit und Flugverhalten

Y. Shen Analyse und Erweiterung von Ansätzen zur Umgebungsrepräsentation für mobile Roboter

J. Steinkamp Analyse und Bewertung von Sensoren zur Umgebungserfassung für Flugroboter

J. Zhu Analyse und Erweiterung von Algorithmen zur Objekterkennung und Objektlokalisierung für mobile Roboter

Inhaltsverzeichnis

Inhaltsverzeichnis	4
Abbildungsverzeichnis	6
Vorbemerkungen	8
1 Einleitung	9
1.1 Problemstellung	9
1.2 Arbeitshypothese	9
2 Stand der Forschung und Entwicklung zum professionellen und industriellen Einsatz von Flugrobotern	10
3 Lösung der Arbeitspakete	11
3.1 Rahmenbedingungen und Anforderung an den Flugroboter und das betrachtete Szenario	11
3.2 Gesamtkonzept zum vollautomatisierten innerbetrieblichen Materialtransport mit autonomen Flugrobotern	12
3.3 Grundlegender Aufbau eines Flugroboters zum vollautomatisierten Materialtransport	13
3.3.1 Hardwarekomponenten des Flugroboters und deren grundlegende Funktionalitäten	13
3.3.2 Software zum autonomen Betrieb des Flugroboters	17
3.3.3 Simulationsumgebung	21
3.3.4 3D-Druck des Drohnenrahmens	22
3.3.5 Ladestation zum automatisierten Laden der Drohne	25
3.4 Poseschätzung für den autonomen Flug in Innenbereichen	26
3.4.1 Sensoren, Systeme und Ansätze zur Posebestimmung	26
3.4.2 Posebestimmung mithilfe eines UWB-Funkortungssystems	29
3.4.3 Kamerabasierte Poseschätzung	30
3.4.4 Evaluation und Bewertung der Poseschätzung	32
3.5 Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung	36
3.5.1 Stand von Forschung und Entwicklung der Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung	36
3.5.2 Konzept zur Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung	37
3.5.3 Erstellung und Aktualisierung einer dreidimensionalen semantischen Karte ...	39
3.5.4 Autonome Navigation des Flugroboters	41
3.5.5 Evaluation und Bewertung der Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung	43
3.6 Vollautomatisierte Erkennung und Handhabung von Kleinladungsträgern mit einem Flugroboter	48

3.6.1	Stand von Forschung und Entwicklung der Objekthandhabung mittels Flugrobotern	48
3.6.2	Konzept zur Objekthandhabung	48
3.6.3	Erkennung des Kleinladungsträgers und Posebestimmung relativ zu diesem.....	50
3.6.4	System zur Aufnahme und zum Ablegen des Kleinladungsträgers	51
3.6.5	Evaluation und Bewertung der Handhabung des Kleinladungsträgers	52
3.7	Gesamtbewertung und Fazit	54
4	Verwendung der Zuwendungen	57
5	Notwendigkeit und Angemessenheit der geleisteten Arbeit.....	58
6	Bewertung des wissenschaftlich-technischen und wirtschaftlichen Nutzens der erzielten Ergebnisse.....	59
7	Ergebnistransfer in die Wirtschaft.....	60
7.1	Transfermaßnahmen in der Laufzeit des Forschungsprojektes	60
7.2	Transfermaßnahmen nach Abschluss des Vorhabens	61
8	Literaturverzeichnis	62

Abbildungsverzeichnis

Abbildung 1: Ablaufdiagramm (links) und Skizze (rechts) zum Ablauf des innerbetrieblichen Materialtransports mit einem autonomen Flugroboter	12
Abbildung 2: Erarbeiteter Demonstrator zum vollautomatisierten Materialtransport; 1: Rahmen, 2: Motoren, 3: Propeller, 4: Flugsteuerung, 5: On-Board-Computer, 6: RC-Empfänger, 7: UWB-Tag	13
Abbildung 3: Weitere im Projekt genutzte Flugroboter, insbesondere für Untersuchungen hinsichtlich des autonomen Betriebs	16
Abbildung 4: Softwarekonzept für den vollautomatisierten Materialtransport mit dem Demonstrator .	18
Abbildung 5: Vorgeschlagenes Softwarekonzept für den vollautonomen Betrieb	20
Abbildung 6: Im Rahmen des Projekts implementierte GUI zur Steuerung und Überwachung des Flugroboters	21
Abbildung 7: Beispielhafte Simulationsumgebung; Links: Überblick über eine Lagerumgebung die in der Simulation erstellt wurde; Rechts: Simuliertes Kamerabild eines Flugroboters in der Lagerumgebung mit im Bild erkannten KLTs	22
Abbildung 8: Links: Konstruierter Multicopterrahmen zum 3D-Druck einschließlich weiterer Elemente wie Propellerschutzkäfigen sowie Ultraschallsensoren zur Absicherung; Rechts: 3D-Druck eines Teilstücks des Rahmens und gedruckte Stützstruktur im inneren des Rahmens.....	22
Abbildung 9: Links: CAD-Modell zweier einzelner Teile des 3D-gedruckten Rahmens die verklebt und mit einem Zylinderstift gefügt werden; Rechts: Teilweise aufgebauter Quadrocopter basierend auf dem 3D-gedruckten Rahmen.....	23
Abbildung 10: PX4-Vibrationsmetrik des Demonstrators während mehrerer Flüge, zwischen denen der Multicopter gelandet ist.....	24
Abbildung 12: Quadrocopter basierend auf dem 3D-gedruckten Rahmen im Flug.....	24
Abbildung 11: PX4-Vibrationsmetrik des Quadrocopters basierend auf dem 3D-gedruckten Rahmen	24
Abbildung 13: Links: Quadrocopter beim Anflug zur autonomen Landung in der Ladestation; Rechts: Quadrocopter nach der Landung in der Station, die roten Lichter zeigen an, dass der Akku der Drohne geladen wird	25
Abbildung 14: Links: Bild des Demonstrators, die Positionen der Tags am Demonstrators sind durch orangefarbene Kreis gekennzeichnet Rechts: Skizze zur Berechnung des Gierwinkels des Multicopters basierend auf den gemessenen Tag-Positionen.....	30
Abbildung 15: Prinzip der Posebestimmung mittels Kamerabildern basierend auf Bildmerkmalen	30
Abbildung 16: Multicopter mit D435-Kamera (oben) und T265-Kamera (unten)	32
Abbildung 17: Vergleich der Lokalisierung in der x- und y-Koordinate unterschiedlicher Posebestimmungssysteme anhand einer quadratischen Trajektorie, die durch den Demonstrator dreimal abgeflogen wurde	33

Abbildung 18: Vergleich des Gierwinkels (Yaw-Winkel) unterschiedlicher Posebestimmungssysteme anhand einer quadratischen Trajektorie, die durch den Demonstrator dreimal abgeflogen wurde	34
Abbildung 19: Vergleich der geschätzten Flughöhe durch die T265-Kamera und VINS-Fusion bei einem Flug mit konstanter Flughöhe von 0,5 Metern	35
Abbildung 20: Konzept zur Absicherung des Flugroboters mit einer Stereo-/RGB-D Kamera (Erfassungsfeld in gelb skizziert) sowie Ultraschallsensorik (Erfassungsfelder in blau skizziert) und Propellerschutzkäfigen (rot)	38
Abbildung 21: Überblick über den erarbeiteten Prozess zur Erstellung der semantischen Karte	39
Abbildung 22: Prozessablauf zur Erstellung und Aktualisierung der semantischen Karte	41
Abbildung 23: Überblick über die Pfadplanung mittels mehrschichtiger Kostenkarten basierend auf dem ROS Navigation Stack	42
Abbildung 24: CAD-Modelle der konzipierten und untersuchten Propellerschutzkäfige.....	44
Abbildung 25: Erfolgreicher Kollisionstest mit den Propellerschutzkäfigen	44
Abbildung 26: Links: Flugroboter, in dessen Umgebung sich drei Pappkartonstapel als Hindernisse befinden, während des Fluges; Rechts: Karte, Hindernissdetektion und Pfadplanung zum Erreichen eines Zielpunktes zwischen de Pappkartons des Flugroboters	45
Abbildung 27: Links: Ultraschallsensor, D435 Depth Camera und T265 Tracking Camera montiert an einem Flugroboter; Rechts: Hindernisdetektion einer Glasscheibe durch den Ultraschallsensor (grüner Kreis) sowie durch die D435-Kamera (roter Kreis)	46
Abbildung 28: Hinderniserkennung und semantische Kartierung während des Fluges; Mitte und Rechts: Pfadplanung basierend auf einer Kostenkarte mit semantische Schichten	47
Abbildung 29: Konzept zur vollautomatisierten Aufnahme eine KLTs durch den Flugroboter	49
Abbildung 30: Im linken Bild wird der KLT nur durch den neu erarbeitenden Ansatz erkannt, die Erkennung des mittig im KLT angebrachten AprilTags ist durch die Beleuchtung und Reflektionen im KLT nicht möglich; Im mittleren und im rechten Bild werden sowohl der KLT als auch der AprilTag im Kamerabild erkannt.....	50
Abbildung 31: Links und Mitte: Führungsschienen der Aufnahmevorrichtung und Motor mit Arm zur Fixierung des KLTs; Rechts: Flugroboter beim KLT-Transport.....	52
Abbildung 32: Plot der x-Koordinate bestimmt durch das UWB-System, den im Projekt erarbeiteten Ansatz und mittels eines AprilTags während eines Fluges über einem KLT in variierender Höhe von 1,5 m bis 0,6 m.....	53
Abbildung 33: Plot der y-Koordinate bestimmt durch das UWB-System, den im Projekt erarbeiteten Ansatz und mittels eines AprilTags während eines Fluges über einem KLT in variierender Höhe von 1,5 m bis 0,6 m.....	54
Abbildung 34: Ablauf der Transportflüge; der erste und dritte Transportflug werden durch die grünen Pfeile angedeutet, der zweite Transportflug durch die gelben Pfeile	55
Abbildung 35: Plot des Schubes des Multicopters in Prozent über der Zeit während des Experiments	55

Vorbemerkungen

Das Vorhaben „Einsatz autonomer Flugroboter für den vollautomatisierten Materialtransport zwischen Lagerplatz und Kommissionierplätzen“ wurde im Förderzeitraum vom 01.05.2019 bis zum 31.10.2021 unter der Leitung von Prof. Dr.-Ing. Jörg Franke am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik (FAPS) der Friedrich-Alexander-Universität Erlangen-Nürnberg durchgeführt.

Im Rahmen des Projektes wurden die erarbeiteten Ideen und gewonnenen Erkenntnisse fortlaufend den Mitgliedern des projektbegleitenden Ausschusses kommuniziert und mit ihnen in fünf regulären Treffen des Projektbegleitenden Ausschusses diskutiert.

Durch die Unterstützung des Projektes vonseiten der Unternehmen im Projektbegleitenden Ausschuss und der IFL war es trotz der Einschränkungen durch die COVID-19-Pandemie möglich, die Ziele des Forschungsprojektes erfolgreich zu erreichen. Wir bedanken uns bei den Unternehmen des Projektbegleitenden Ausschusses für die Unterstützung und die zahlreichen Austauschmöglichkeiten.

Die Bearbeitung des Forschungsvorhabens erfolgte gemäß dem Arbeitsplan des Projektantrages in sechs Arbeitspaketen. Aufgrund der COVID-19-Pandemie wurde die Projektlaufzeit kostenneutral um sechs Monate verlängert. Durch die Maßnahmen und gesetzlichen Regelungen zur Eindämmung und Bekämpfung der Pandemie war der Zugang zum Versuchsfeld für experimentelle Untersuchungen und Validierungen zeitweise nicht oder nur eingeschränkt möglich. Diese pandemiebedingten Verzögerungen konnten durch die kostenneutrale Laufzeitverlängerung ausgeglichen werden.

Der Schlussbericht greift die Arbeiten und Ergebnisse der einzelnen Arbeitspakete auf und vernetzt diese zu einer Gesamtbetrachtung der Ergebnisse des Forschungsvorhabens. In Kapitel 1 wird die grundlegende Problemstellung, die Anlass des Forschungsprojektes war, sowie die daraus abgeleitete Arbeitshypothese erläutert. In Kapitel 2 wird der Stand von Forschung und Entwicklung des professionellen und industriellen Einsatzes von Flugrobotern dargestellt. Der erarbeitete Lösungsweg für die im Projekt betrachtete Problemstellung, die erzielten Ergebnisse der Arbeitspakete im Gesamtkontext und die Validierung der Lösungsansätze werden in Kapitel 3 beschrieben. In den Kapiteln 4 bis 6 werden die Mittelverwendung, deren Notwendigkeit und Angemessenheit sowie der wissenschaftlich-technische und wirtschaftliche Nutzen der erzielten Ergebnisse erläutert. Kapitel 8 gibt einen Überblick über die durchgeführten und zukünftigen Maßnahmen bezüglich des Transfers der Ergebnisse in die Wirtschaft.

1 Einleitung

1.1 Problemstellung

Drohnen finden zunehmend Einsatz in gewerblichen und industriellen Bereichen. Für die Anzahl an kommerziell genutzten Drohnen wird ein Wachstum von 45.000 im Jahr 2020 auf 132.000 im Jahr 2025 geschätzt [1]. Nach ihrer Etablierung im Verbrauchermarkt haben Drohnen bzw. Multicopter mittlerweile Einzug in zahlreiche industrielle Anwendungsfelder gefunden und sich in Bereichen wie Vermessung, Inspektion und Inventur etabliert.

Durch technische Entwicklungen und fortschreitende Autonomisierung sind in den vergangenen Jahren Projekte mit autonomen Flugrobotern für die Warenzustellung in Außenbereichen gestartet und durchgeführt worden [2, 3].

Es stellt sich also die Forschungsfrage, ob und wie der intralogistische Materialtransport mithilfe von autonomen Flugrobotern durchgeführt bzw. ergänzt werden kann. Besonders im innerbetrieblichen Einsatz bieten Flugroboter weiterführende Potentiale und Einsatzmöglichkeiten.

Wie auch bei der Zustellung von Lieferungen in Außenbereichen bieten autonome Flugroboter aufgrund ihrer dreidimensionalen Bewegungsmöglichkeit und den daraus resultierenden, im Vergleich zum flurgebundenen Transport, kurzen Transportstrecken, der erzielbaren Transportgeschwindigkeiten und ihrer flexiblen Einsetzbarkeit weitreichende Einsatzmöglichkeiten. Weiterhin ermöglichen sowohl die Verlagerung einzelner als auch zyklischer Liefervorgänge in die Luft eine Reduktion der am Boden für Logistikprozesse benötigten Fläche, sodass die Flächenproduktivität signifikant gesteigert wird.

1.2 Arbeitshypothese

Ziel des Forschungsprojektes AIRKom war die Erforschung des Einsatzes autonomer Flugroboter für den innerbetrieblichen Warentransport, insbesondere im Kontext der vollautonomen Belieferung von Kommissionierplätzen direkt vom Lagerort mittels Flugrobotern.

Zur Lösung der Problemstellung und zur Umsetzung der Forschungsergebnisse im Rahmen eines funktionsfähigen Demonstrators sind technische Innovationen im Bereich Handhabung von Objekten mit Flugrobotern, in der präzisen und robusten Navigation von Flugrobotern im Innenbereich und der sensorbasierten Kollisionsvermeidung sowie Untersuchungen im Bereich der Arbeitssicherheit essentiell. Zur Realisierung der vollautomatisierten Aufnahme und Ablage von Kleinladungsträgern muss ein mechatronisches Aufnahmesystem für den Flugroboter entwickelt und realisiert werden, das den Anforderungen der Intralogistik gerecht wird. Um die während der Objektaufnahme und -abgabe erforderliche Lokalisierungs- und Positionierungsgenauigkeit zu erreichen, müssen die Ladungsträger optisch detektiert werden und die berechnete Relativposition muss anschließend mit den Positionsdaten bestehender innerbetrieblicher Ortungssysteme sowie den Inertialsensordaten des Flugroboters fusioniert werden, um eine Fluggenauigkeit im Subzentimeterbereich zu ermöglichen, die, mit für den industriellen Einsatz geeigneter Technik, aktuell nicht erreichbar ist.

Die entwickelten Technologien und Lösungen sollten weiterhin die Grundlage für universell einsetzbare innerbetriebliche Flugroboter legen.

2 Stand der Forschung und Entwicklung zum professionellen und industriellen Einsatz von Flugrobotern

Aktuell werden Drohnen bereits für unterschiedlichste professionelle und industrielle Anwendungen genutzt. Neben Anwendungen in Außenbereichen wie der Landwirtschaft [4], der Inspektion [5], der Überwachung und dem Transport [2, 3], werden Drohnen auch für Anwendungen in Innenbereichen genutzt. Insbesondere die Inventur in Hochregallagern mittels Drohnen ist bereits am Markt verfügbar [6].

Jedoch sind für den Innenbereich noch keine Flugrobotersysteme am Markt vorhanden, die autonom arbeiten. Hürden hierbei, die aktuell in der Forschung untersucht und mit neuen Lösungsansätzen überwunden werden müssen, sind die präzise, robuste und sichere Navigation und Kollisionsvermeidung in Innenbereichen sowie die Umgebungserfassung und Objektinteraktion.

In [7] wird eine autonome Drohne zu Inventur in Innenbereichen vorgestellt. Der Flugroboter kann die Inventuraufgabe in Kollaboration mit einem bodengebundenen Roboter selbstständig lösen, jedoch werden keine sicherheitstechnischen Aspekte oder eine umfassende Umgebungserfassung betrachtet.

Ein Ansatz zu Navigation für einen autonomen Flugroboter in Innenbereichen wird in [8] präsentiert. Hier ist auch die Kollisionsvermeidung ein wesentlicher Aspekt. Der Aspekt der Objektinteraktion bzw. Objekthandhabung wird hier jedoch nicht betrachtet.

In einer Vorarbeit des Lehrstuhls FAPS wird das Prinzip des innerbetrieblichen Materialtransports mit einem Flugroboter betrachtet [9]. Auch hier wird die Objekthandhabung grundlegend gelöst, jedoch ohne Betrachtung einer umfassenden Umgebungserfassung und Navigation, was für den langfristigen und effizienten Einsatz von autonomen Flugrobotern in der Intralogistik erforderliche Fähigkeiten sind.

Insgesamt ist im Stand von Forschung und Entwicklung kein autonomes Flugrobotersystem vorhanden, welches alle Anforderungen hinsichtlich Autonomie des Roboters sowie der vollautomatisierten Objekthandhabung abdeckt [10].

Auf den Stand von Forschung und Entwicklung zu den spezifischen Themengebieten,

- Lokalisierung und Poseschätzung für Multicopter in Innenbereichen,
- Kollisionsabsicherung, Umgebungskartierung, Kollisionsvermeidung und Navigation und
- vollautomatisierte Handhabung von Objekten,

wird an den entsprechenden Stellen in Kapitel 3 eingegangen, um die Vorgehensweise bei der Lösung der Arbeitspakete aufbauend auf dem Stand von Forschung und Entwicklung darzulegen.

3 Lösung der Arbeitspakete

In diesem Kapitel wird der im Projekt erarbeitete Lösungsweg für den vollautomatisierten Materialtransport mittels autonomer Flugroboter in der Intralogistik vorgestellt. Zunächst werden in Abschnitt 3.1 Rahmenbedingung und Anforderungen an den Flugroboter und das betrachtete Szenario bzw. die Umgebung definiert. In Abschnitt 3.2 wird ein Überblick über das erarbeitete Gesamtkonzept gegeben. In den folgenden vier Abschnitten 3.3 bis 3.6 wird im Detail auf den Flugroboter (Abschnitt 3.3), die Lokalisierung bzw. Poseschätzung für den autonomen Flug im Innenbereich (Abschnitt 3.4), die Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung (Abschnitt 3.5) sowie die vollautomatisierte Erkennung und Handhabung von Kleinladungsträgern durch den autonomen Flugroboter eingegangen. Hierbei wird jeweils ein kurzer themenbezogener Überblick über den Stand von Forschung und Entwicklung geben und die im Projekt erarbeiteten Ansätze und Methoden vorgestellt und bewertet. Abschließend werden in Abschnitt 3.7 die wichtigsten Erkenntnisse zusammengefasst und ein Fazit gezogen.

3.1 Rahmenbedingungen und Anforderung an den Flugroboter und das betrachtete Szenario

Für das im Rahmen des Projektes untersuchte Szenario für den Einsatz von autonomen Flugrobotern zum vollautomatisierten innerbetrieblichen Materialtransport wurden aus Komplexitätsgründen folgende Rahmenbedingungen festgelegt:

- Der Flugroboter soll einen Kleinladungsträger (KLT) der Bauform RL-KLT 3147 (Maße: 300 × 200 × 147 mm, entsprechend VDA-Empfehlung 4500) transportieren.
- Der Transport erfolgt ausschließlich im Innenbereich, es werden somit keine Wetter- bzw. Witterungseffekte auf den Flugroboter wie Wind oder Regen betrachtet.
- Menschen kollaborieren nicht direkt mit dem Flugroboter, die Absicherung von Begegnungen zwischen Mensch und Flugroboter in einem gewissen Arbeitsraum werden aber im Projekt betrachtet.
- Die im Projekt erarbeitete und untersuchte grundlegende Flugroboterbauform entspricht einem Multicopter. Aufgrund des teilweise stark begrenzten verfügbaren Raumvolumens in Innenbereichen für Flugmanöver ist diese Bauform dank der hohen Manövrierfähigkeit und der Möglichkeit senkrecht zu starten und zu landen für das betrachtete Szenario gut geeignet.

Hieraus ergeben sich folgende Anforderungen an den Flugroboter:

- Der Flugroboter muss sich im Innenbereich ausreichend genau und robust lokalisieren, um sicher Hindernissen auszuweichen und den KLT zu handhaben.
- Der Flugroboter muss Hindernissen in der Umgebung selbständig ausweichen, um Kollisionen zu vermeiden.
- Der Flugroboter muss in der Lage sein, den zu transportierenden Kleinladungsträger selbstständig zu erkennen, zu lokalisieren, aufzunehmen und abzulegen.
- Der Flugroboter soll möglichst leicht sein, um eine möglichst hohe Nutzlastaufnahme bzw. Flugzeit zu ermöglichen.

3.2 Gesamtkonzept zum vollautomatisierten innerbetrieblichen Materialtransport mit autonomen Flugrobotern

In Abbildung 1 links ist das im Projekt erarbeitete Gesamtkonzept für den vollautomatisierten innerbetrieblichen Materialtransport mit einem autonomen Flugroboter als Blockdiagramm dargestellt. Der Flugroboter befindet sich zu Beginn des Prozesses an einem definierten Ort, beispielsweise einer Ladestation. Wird der Transport eines Kleinladungsträgers angefordert (1), nimmt der Flugroboter diesen an, soweit keine Gründe dagegensprechen (z. B. Akkustand zu gering) (2). Hierfür und zur weiteren Kommunikation und Überwachung ist der Flugroboter via Funk mit einem anderen Rechner, der sogenannten Bodenstation (engl. Ground Control Station (GCS)), verbunden.

Nach der Annahme des Transportauftrages startet der Flugroboter selbständig und navigiert autonom zum Ziel (3). Dies umfasst die präzise Lokalisierung bzw. Poseschätzung des Flugroboters und eine Kollisionsvermeidung.

Sobald der Flugroboter sich in der Nähe des zu transportierenden Kleinladungsträger befindet, muss der Flugroboter dessen Pose relativ zur eigenen bestimmen. Mit dieser Information kann der Handhabungsprozess begonnen werden. Hierzu landet der Flugroboter auf dem KLT und fixiert diesen mit einer entsprechenden Lastaufnahme an sich selbst (4).

Anschließend fliegt der Flugroboter mit dem KLT autonom zum Zielort, zu dem der KLT transportiert werden soll (5). Dort landet der Roboter, löst die Fixierung des KLT und legt diese so ab (6). Um für einen folgenden Transportauftrag einsatzbereit zu sein, ist es sinnvoll, die Akkus zu laden. Dies kann wiederum in einer entsprechenden Ladestation für Multicopter geschehen (7).

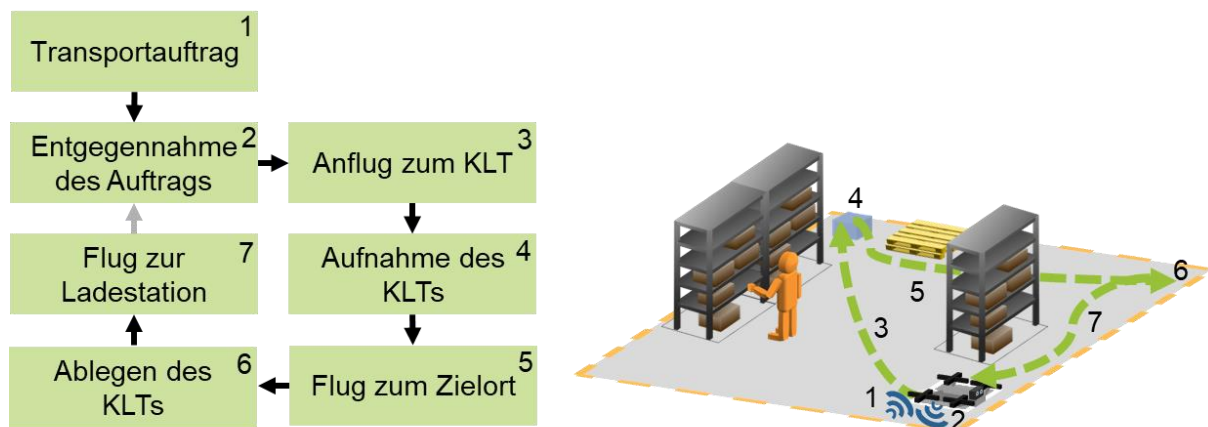


Abbildung 1: Ablaufdiagramm (links) und Skizze (rechts) zum Ablauf des innerbetrieblichen Materialtransports mit einem autonomen Flugroboter

3.3 Grundlegender Aufbau eines Flugroboters zum vollautomatisierten Materialtransport

In diesem Abschnitt wird der grundlegende Aufbau des Demonstrators vorgestellt, mit dem der vollautomatisierte Materialtransport realisiert und validiert wurde. Zunächst wird hierbei auf den Hardwareaufbau eingegangen (Abschnitt 3.3.1) und anschließend das erarbeitete Softwarekonzept sowie die hierfür genutzte Software vorgestellt (Abschnitt 3.3.2). Zudem wird in Abschnitt 3.3.3 eine Simulationsumgebung, die zur Entwicklung und Erprobung der Software genutzt wurden vorgestellt. In Abschnitt 3.3.4 werden die im Projekt untersuchte Fertigung eines 3D-gedruckten Multicopterrahmens und Flugexperimente mit diesem vorgestellt. Abschließend werden in Abschnitt 3.5.4 Experimente mit einer Ladestation zum automatisierten Laden der Akkus des Flugroboters erläutert.

3.3.1 Hardwarekomponenten des Flugroboters und deren grundlegende Funktionalitäten

In Abbildung 2 ist ein Foto des erarbeiteten Demonstrators zu sehen. Wichtige Elemente des Flugroboters sind hierbei durch Pfeile mit entsprechender Nummerierung gekennzeichnet. Diese, sowie weitere nicht im Bild sichtbare Elemente und deren Funktionsweise werden im Folgenden erläutert.

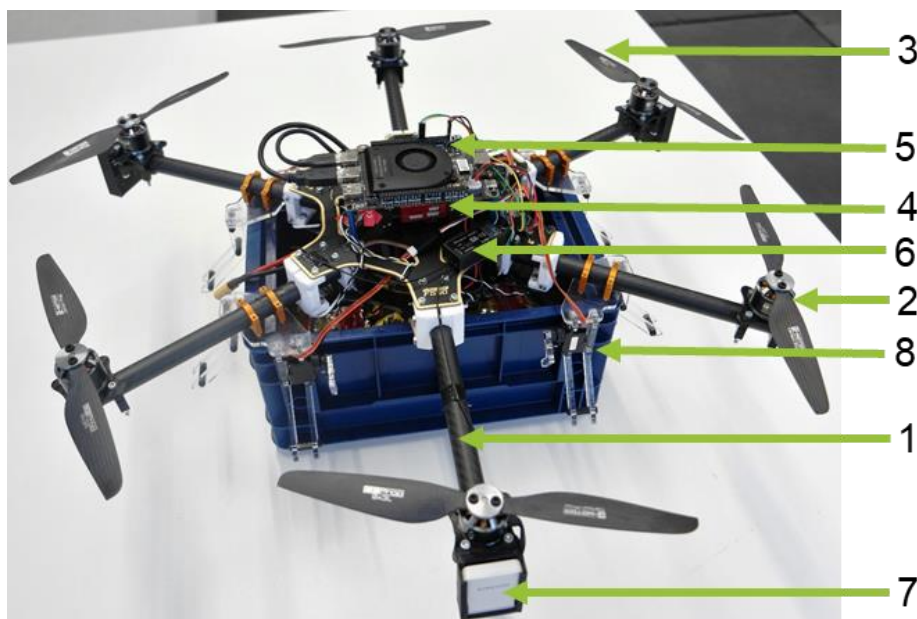


Abbildung 2: Erarbeiteter Demonstrator zum vollautomatisierten Materialtransport; 1: Rahmen, 2: Motoren, 3: Propeller, 4: Flugsteuerung, 5: On-Board-Computer, 6: RC-Empfänger, 7: UWB-Tag

Das grundlegende Element eines Flugroboters ist der Rahmen (1). Dieser gibt die grundsätzliche Form des Roboters vor. Er besteht meist aus einer oder mehreren Grundplatten an denen mehrere Arme zur Anbringung der Motoren (2) symmetrisch angebracht sind. Auf Basis der Anzahl der Arme wird auch die konkrete Bezeichnung für einen Multicopter abgeleitet (z. B. vier Arme: Quadrocopter; sechs Arme: Hexacocter; acht Arme: Octocopter). Für den Rahmen sollten möglichst leichte und steife Bauteile genutzt werden, um einerseits das Eigengewicht des Rahmens zu minimieren und andererseits Vibrationen im Flug zu vermeiden. Vibrati-

onen, hervorgerufen durch die Drehungen der Motoren und der Propeller (3) oder lose Komponenten können über den Rahmen zu Sensoren und weiteren Elementen des Flugroboters geleitet werden und deren Messungen negativ beeinflussen. Dieser Effekt ist insbesondere bei Beschleunigungssensoren auf dem Flugroboter beobachtbar und führt dadurch zu stark verrauschten Messdaten und somit zu einem schlechteren und instabilen Flugverhalten des Roboters. Langfristig können die Vibration auch zu einer Ablösung von weiteren Komponenten, die am Rahmen befestigt sind führen.

Der Rahmen des im Projekt erarbeiteten Demonstrators besteht aus den zwei Grundplatten eines DJI FLAME WHEEL ARF KIT F550 Hexacopters. Zur Optimierung des Flugverhaltens und Anbringung von weiteren Elementen (z. B. der Lastaufnahme) wurden Arme aus Carbonrohren (CFK-Rohre) zwischen den Platten mit eigens konstruierten und 3D-gedruckten Halterungen starr angebracht. Auf der oberen Grundplatte zentriert ist die Flugsteuerung (engl. Flight Control Unit (FCU); 4) angebracht. Mit Abstandshaltern darüber ist ein On-Board-Computer (OBC; 5) befestigt. Zudem ist ein Funk-Empfänger (RC-Receiver; 6) für den Empfang von Steuerbefehlen von einer manuellen Fernsteuerung auf der Grundplatte befestigt. Zwischen die Platten wird der Akku zur Energieversorgung des Flugroboters eingeschoben und befestigt. In der unteren Grundplatte sind Leitungen zu Stromverteilung an die Motoren integriert. Der Akku wird an mittels eines XT60-Steckers mit den Leitungen in der unteren Grundplatte verbunden. An diese Leitungen sind für jeden Motor Electronic Speed Controller (ESCs) angelötet, mit welchen den Stromfluss zu den Motoren und so deren Drehzahl durch die FCU reguliert werden kann. Zusätzlich sind DC/DC-Wandler auf der unteren Platte angebracht und verbunden, um den OBC und weitere elektronische Komponenten mit Energie auf den benötigten Spannungslevel bereitzustellen. Auf der Unterseite der unteren Grundplatte ist eine Farbkamera (Logitech C270) angebracht, die zur Erkennung des KLTs dient (siehe Abschnitt 3.6.3). An den Enden der Arme sind die Motoren befestigt, an welchen wiederum die Propeller fixiert sind. Zusätzlich befinden sich an drei Armen Tags zur Lokalisierung des Flugroboters mittels eines Ultrabreitband-Funklokalisierungssystems (UWB-System; 7). Diese sind mit dafür konstruierten und 3D-gedruckten Halterungen befestigt.

Im Rahmen des Projekts wurde die Grundform eines Hexacopters gewählt, da dieser einerseits ausreichenden Schub erzeugen kann, um einen KLT zu transportieren und andererseits noch ausreichend klein dimensioniert werden kann, um in teilweise engen Innenbereichen sicher zu navigieren. Zudem wäre es technisch möglich, den Ausfall eines Motors durch die verbleibenden fünf Motoren zu kompensieren und so eine sicheres Flugverhalten weiter zu gewährleisten.

Die Flugsteuerung steuert und regelt den Flug und die Aktionen des Flugroboters. Hierfür gibt sie entsprechende Steuersignale an die ESCs, welche den Stromfluss an die Motoren und somit deren Drehzahl einstellen. Diese ist für jeden Motor separat möglich, sodass durch entsprechend unterschiedliche Drehzahlen der einzelnen Motoren die dreidimensionale Bewegung des Multicopters ermöglicht wird. Das Schweben (Halten einer Höhe) des Roboters in der Luft, sowie das Aufsteigen (höherer Schub) bzw. Absinken (geringer Schub) wird durch eine entsprechende Anpassung des Motorstroms und so der Drehzahlen erzielt. Drehungen und Bewegungen im Raum werden analog durch Anpassung der Drehzahlen einzelner Motoren erzielt. Hierdurch wird gesteuert welche Kräfte auf den Roboter durch die Propeller an den einzelnen Armen generiert werden. Werden alle Motoren in einem passenden Muster angesteuert so kann präzise die Gesamtkraft auf den Flugroboter und so dessen Bewegung und Rotation gesteuert werden. Zur Vermeidung von Drehmomenten, die durch die Propeller

auf den Roboter wirken könnten, drehen sich nebeneinanderliegende Motoren jeweils in entgegengesetzte Richtung [11]. D. h. drei Motoren des Hexacopters drehen sich im Uhrzeigersinn, drei entgegen dem Uhrzeigersinn.

Der maximale Schub, den die Drohne erzeugen kann, wird durch die Akkuspannung, die Anzahl und die spezifischen Eigenschaften der Motoren sowie die Bauform der Propeller (Länge und Steigung) bestimmt. Die Dimensionierung dieser Elemente sollte also entsprechend dem gewünschten Einsatzszenario erfolgen. Hierbei ist das gewünschte Gesamtgewicht (Eigengewicht und Nutzlast) sowie die gewünschte Flugzeit zu berücksichtigen. Softwareprogramme, wie beispielsweise „eCalc“ [12], unterstützen bei der Auswahl und Dimensionierung dieser Elemente sowie passender ESCs, welche entsprechende maximale Ströme leiten und regulieren müssen.

Im Stand der Technik werden zur Energieversorgung für Multicopter meist Lithium-Polymer-Akkumulatoren (LiPo-Akkus) genutzt. Diese bieten eine hohe Energiedichte und Ausgangsströme bei einem geeigneten Gewicht. Bei unsachgemäßer Handhabung und Tiefenentladung (z. B. durch einen zu langen Flug) besteht jedoch erhebliche Brandgefahr. [13, 14] Weitere Gefahrenpotentiale ergeben sich durch die sich drehenden Propeller, herabfallende Teile sowie Kollisionen mit dem Flugroboter. Teilweise können diese Gefahren durch eine redundante Auslegung der Komponenten des Roboters gemindert werden. Dies wird z. B. für FCUs in [15] beschrieben. Eine Übersicht über mögliche Gefahrenpotentiale und Maßnahmen zu Vermeidung bzw. Verminderung dieser wird in [13] und [14] gegeben. Im Rahmen des Projekts wurde die Redundanz nicht weiter untersucht, da im Stand von Forschung und Entwicklung bereits Ansätze und Lösungen hierfür verfügbar sind.

Die erarbeiteten Konzepte, Untersuchungen und Experimente zur Vermeidung von Kollisionen und zur Gefahrenminderung bei Kollisionen werden in Abschnitt 3.5 vorgestellt.

Zur Regelung der Motordrehzahlen für das gewünschte Flugverhalten bzw. die gewünschten Flugmanöver muss die Orientierung und Position des Flugroboters im Raum (Pose) bekannt sein bzw. geschätzt werden. Dies übernimmt die Flugsteuerung mit entsprechender Software. Hierfür sind Sensordaten erforderlich. Die entsprechenden Sensoren können in der Flugsteuerung verbaut sein (z. B. Inertialsensorik) oder am Flugroboter angebracht sein (z. B. Kamera) oder auch externe Infrastruktur und Sensoren einbeziehen (z. B. UWB-System, Motion Capture System). Aus den Messdaten schätzt die Software der Flugsteuerung die aktuelle Lage und Position der Drohne und gibt Korrektur- bzw. Steuersignale für gewünschte Flugmanöver an die jeweiligen ESCs.

Zur Sensordatenverarbeitung und autonomen Steuerung und Navigation des Flugroboters dient der On-Board-Computer. Dieser wird auch zur Kommunikation mit der Bodenstation genutzt und kann so z. B. Transportaufträge entgegennehmen und den Status des Roboters übermitteln.

Zusätzlich sollte der Flugroboter über einen entsprechenden Funkempfänger verfügen, der mit einer Fernsteuerung gekoppelt ist. Hierdurch kann der Multicopter komplett oder auch einzelne Funktionen (z. B. Not-Aus, engl. Kill-Switch) manuell gesteuert werden.

Die für den vollautomatisierten Materialtransport erarbeitete Aufnahmevorrichtung wird in Abschnitt 3.6 beschrieben.

Eine tabellarische Auflistung der wichtigsten Komponenten des Demonstrators ist in Tabelle zu finden. Der Demonstrator hat einen Durchmesser von 60 Zentimetern und ein Gewicht (einschließlich Akku, ohne Nutzlast) von ca. 2050 Gramm.

Tabelle 1: Überblick über die Komponenten des Demonstrators

Bauelement	Bezeichnung
Motoren	T-Motor MN2212 KV920-V2.0
Propeller	T-Motor CFProp 9x3A
ESCs	DJI 30A Opto ESC
Flugsteuerung	Pixracer Autopilot V1.0
On-Board-Computer	Latte Panda Alpha 864s
LiPo-Akku	Turnigy 4S1P 4000 mAh
RC-Receiver	Graupner GR-16 HoTT
Fernsteuerung	Graupner mx-20 HoTT

Der gezeigte und beschriebene Demonstrator kann den vollautomatisierten Transport von KLTs selbständig durchführen. Für die vollständige Autonomie sind jedoch weitere Sensoren sowie zusätzliche Software erforderlich. Im Rahmen des Forschungsprojektes war es jedoch nicht möglich, die gesamte Software für den autonomen Betrieb auf den verfügbaren und in den Flugroboter integrierbaren OBC auszuführen. Neben dem Latte Panda Alpha 864s wurde auch ein NVIDIA Jetson Xavier NX (Abbildung 3 links) als OBC genutzt, der über eine GPU für die effiziente Ausführung von Algorithmen basierend auf maschinellen Lernen verfügt. Die Flugroboter in Abbildung 3 sind grundlegend wie der vorgestellte Demonstrator aufgebaut, unterscheiden sich jedoch in wenigen Komponenten und deren Spezifikationen.

Die Integration aller Funktionalitäten (KLT-Transport und vollständig autonomer Betrieb des Flugroboters) scheint aber basierend auf den erzielten Ergebnissen durch Softwareoptimierungen und -erweiterungen möglich. Eine Skalierung des Flugroboters, sodass auch leistungsfähigerer OBC (mit mehr Gewicht und Energiebedarf) genutzt werden kann, wäre ein weiterer Ansatz, um diese Hürde zu überwinden.



Abbildung 3: Weitere im Projekt genutzte Flugroboter, insbesondere für Untersuchungen hinsichtlich des autonomen Betriebs

3.3.2 Software zum autonomen Betrieb des Flugroboters

Zum autonomen Betrieb des Flugroboters sind mehrere Softwaremodule, verteilt auf zwei Rechner, die Flugsteuerung und den OBC, erforderlich.

Die FCU steuert und regelt direkt den Flug des Roboters. Im Rahmen des Projektes wurde mit einer Pixracer Autopilot V1.0 Flugsteuerung gearbeitet, auf der die PX4 Autopilot Software (im folgenden PX4) ausgeführt wird [16, 17]. Die Flugsteuerung kann ohne Einschränkungen durch die meisten anderen von PX4 unterstützten Modelle ersetzt werden [18]. Die Projektergebnisse wurden mit Version 11.1.3 sowie vorherigen Versionen der PX4-Software erzielt.

Die PX4-Software wird im Rahmen eines Open-Source-Projektes entwickelt, der Quellcode ist also frei zugänglich [18]. Der Kern der Software ist die Zustandsschätzung des Multicopters sowie die darauf und den Flugbefehlen aufbauende Regelung der Motordrehzahlen. Die Zustandsschätzung der Flugsteuerung basiert auf einem Extended Kalman Filter (EKF). Das EKF schätzt aus unterschiedlichen Eingangsdaten (Sensordaten) den aktuellen Zustand des Roboters. Dieser umfasst die Lage, Geschwindigkeit und Position des Multicopters sowie weitere für den Multicopter relevante Größen, wie Störgrößen der Sensoren oder die Windgeschwindigkeit. Neben den in der FCU integrierten Sensoren (Inertiale Messeinheit, Magnetometer, Barometer) zur Zustandsschätzung können auch weitere Sensoren bzw. deren Messdaten, zur Stützung und Verbesserung der Zustandsschätzung eingebracht werden. Im Rahmen des Projektes wurde diese Stützung durch weitere, für den Innenbereich geeignete Sensorik untersucht und wird in Abschnitt 3.4 detaillierter beschrieben. Eine Einschränkung ist jedoch, dass ausschließlich eine externe Quelle mit Posedaten zur Stützung in den EKF eingespeist werden kann. Das heißt, falls der EKF durch die Posedaten mehrerer zusätzlicher Systeme gestützt werden soll, müssten diese zuvor fusioniert werden.

Die PX4-Software und somit das Verhalten des Flugroboters kann über eine Vielzahl an Parametern eingestellt werden. Relevant ist hier vor allem die Konfiguration des EKFs, sodass nur die gewünschten Sensordaten mit entsprechenden Kovarianzen zur Zustandsschätzung genutzt werden. Zudem kann die maximale Geschwindigkeit des Roboters mittels der Parameter limitiert werden. Für viele gängige Bauformen von Multicoptern bietet PX4 bereits grundlegend geeignete Parametersets. Bei angepassten und selbst gebauten Modellen kann die Optimierung der Reglerparameter das Flugverhalten optimieren, eine Anleitung hierzu ist verfügbar [19]. Über die Parameter können auch Sicherheitsfunktion aktiviert bzw. deaktiviert werden, beispielsweise, dass auch ein Betrieb und die Kommunikation via angeschlossenem USB-Kabel mit dem OBC möglich ist, sowie das Verhalten des Roboters bei einem fast leeren Akku oder ähnlichen sicherheitsrelevanten Ereignissen.

Die PX4-Software unterstützt mehrere Flugmodi, von denen einige nur für die manuelle Fernsteuerung relevant sind. Für den autonomen Betrieb im Rahmen des hier betrachteten Anwendungsfalles sind vor allem die Modi „TakeOff“, „Land“ und „Offboard“ relevant. Im „TakeOff“ Modus hebt der Multicopter auf eine definierte Höhe ab und hält die Position und Höhe, bis er weitere Kommandos erhält. Beim Wechsel in den „Land“-Modus landet der Multicopter an der Stelle, über der er sich gerade befindet. Der „Offboard“-Modus dient zum autonomen

Betrieb des Multicopters, sodass die Befehle ausgeführt werden, die der OBC an die Flugsteuerung sendet. [20] [17]

Zusätzlich ist jedoch eine manuelle Fernsteuerung empfehlenswert, um im Notfall den Multicopter mittels dieser und dem entsprechenden RC-Receiver, der direkt mit der FCU verbunden ist, ausschalten zu können. Wird der „Kill-Switch“ manuell ausgelöst, so stoppt die Flugsteuerung sofort alle Motoren (auch im Flug), unabhängig von allen zuvor erhaltenen Befehlen. Dies entspricht also einer Notaus-Funktionalität.

Die PX4-Software bietet zudem die Möglichkeit, relevante Daten, wie die Zustandsschätzung oder erhaltene Befehle, zur Auswertung nach dem Flug auf einer SD-Karte zu speichern (Logging).

Auf dem OBC des Flugroboters wurde für dieses Projekt das Betriebssystem Ubuntu 18.04 sowie das Meta-Betriebssystem und Software-Framework Robot Operating System (ROS; Version „Melodic Morenia“) verwendet [21, 22]. Das ROS bietet unterschiedliche Funktionalitäten zur Entwicklung und zum Betrieb von autonomen mobilen Robotern. Dies sind beispielsweise Softwarebibliotheken für die Interprozesskommunikation, Gerätetreiber für Sensoren und Aktoren und Open-Source Softwarepakete, die bereits entwickelte Funktionalitäten für autonome Roboter enthalten.

Für die Kommunikation des OBCs mit der FCU, wurde das ROS-Softwarepaket „mavros“ genutzt [23]. Dieses ermöglicht die Kommunikation von Software basierend auf dem ROS mit der PX4-Software mittels des „mavlink“-Protokolls und somit die autonome Steuerung des Flugroboters. [16, 24]

Ein Überblick über die einzelnen Softwaremodule, welche für den vollautomatisierten Materialtransport mit dem Flugroboter genutzt und erarbeitet wurden, wird in Abbildung 4 gegeben. Hier ist das Softwarekonzept bzw. die Softwarerealisierung skizziert, mit welcher der vollautomatisierte Materialtransport umgesetzt wurde.

Wie bereits beschrieben, sind für eine umfassende Autonomie des Flugroboters eine leistungstärkeren OBC und weitere Softwareoptimierungen erforderlich.

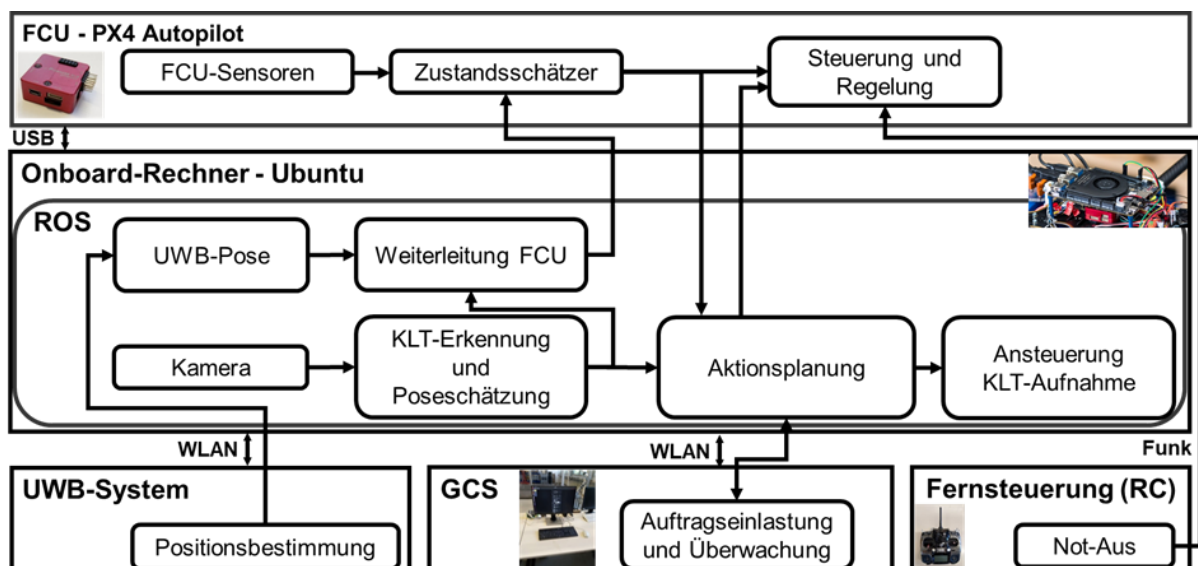


Abbildung 4: Softwarekonzept für den vollautomatisierten Materialtransport mit dem Demonstrator

Das oberste Rechteck in Abbildung 4 stellt die FCU mit der PX4-Software und wichtigen Softwaremodulen da. Das Rechteck darunter umfasst die Software, die auf dem OBC ausgeführt wird. Dies ist das ROS und auf ROS basierende weitere Softwaremodule (ROS Nodes). Die

unteren Rechtecke stellen die GCS, das UWB-System und dessen Anker sowie die manuelle Fernsteuerung dar. Die GCS und das UWB-System sind zur Kommunikation via WLAN mit dem OBC verbunden, die Fernsteuerung kann über eine separate Funkverbindung Befehle (Not-Aus) mittels des RC-Empfängers an die FCU senden.

Neben weiteren Funktionalitäten, die die PX4-Software umfasst und die auch ausgeführt werden, sind für den betrachteten Anwendungsfall insbesondere die Zustandsschätzung und die Steuerung und Regelung des Flugroboters durch die FCU von Bedeutung. Der Zustandsschätzer (EKF) erhält zunächst einerseits Messdaten der in der FCU integrierten Sensoren (Details hierzu in Abschnitt 3.4). Zusätzlich wird die Zustandsschätzung der FCU durch externe Informationen gestützt. Dies ist im normalen Flug, z. B. beim Anflug zum KLT, die Poseschätzung basierend auf einem UWB-Funkortungssystem und sobald der Flugroboter mit der nach unten gerichteten Kamera den zu transportierenden KLT erkannt hat, die Poseschätzung basierend auf dem in den Bildern erkannten KLT (Abschnitt 3.6.3). Im Rahmen des Projektes wurde dies mittels eines Umschaltverhaltens realisiert. D. h. solange der Flugroboter nicht über einem zu transportierenden KLT schwebt, wird die Zustandsschätzung durch die UWB-Posebestimmung gestützt. Sobald eine KLT-basierte Posebestimmung verfügbar ist, wird die Zustandsschätzung durch diese gestützt. Die aktuelle Zustandsschätzung wird einerseits durch die PX4-Software zur Steuerung und Regelung des Flugroboters genutzt, andererseits auch in ein ROS-basiertes selbst erarbeitetes Softwaremodul gespeist, das der Übermittlung von Flugbefehlen sowie der Auftragsverwaltung und Aktionsplanung dient. Wird ein neuer Transportauftrag vom Kontrollrechner an dieses Softwaremodul gesandt, so werden die entsprechende Befehle (z. B. Starten, zu einer Position fliegen, Landen) von diesem Modul passend für den Transportauftrag generiert und an die Flugsteuerung gesandt. Dieses Softwaremodul fängt auch Fehler und Probleme ab, z. B. wenn sich ein KLT nicht an der richtigen Position befindet oder nicht erkannt werden kann. Hierauf reagiert das Modul und entsprechend der Flugroboter mit zuvor definierten Aktionen, z. B. mit einem Rückflug zur Ausgangsposition.

Falls der KLT erkannt wurde und die Landung darauf erfolgreich war, so gibt diese Modul den Befehl zur Fixierung des KLTs am Roboter.

Das erarbeitete Softwarekonzept einschließlich aller erforderlichen Funktionalitäten für den autonomen Betrieb des Flugroboters ist in Abbildung 5 dargestellt. Wesentliche Änderungen ergeben sich hier nur bei der Software, die auf dem OBC ausgeführt wird. Zur Poseschätzung während des Fluges wird hier nun eine Poseschätzung unter zusätzlicher Einbeziehung einer visuellen-inertialen Odometrie vorgeschlagen. Zur Kollisionsvermeidung und Umgebungskartierung werden zusätzlich eine RGBD-Kamera und Ultraschallsensoren genutzt. Die erstellte Karte kann auch zur Poseschätzung genutzt werden. Abhängig von der aktuellen Pose, der Umgebungskarte und dem aktuellen Auftrag kann durch das Aktionsplanungsmodul eine Pfadplanung angestoßen werden. Basierend auf dem so geplanten Pfad, gibt das Aktionsplanungsmodul entsprechende Zielposen an die Flugsteuerung. Sollte durch das Kollisionsvermeidungsmodul eine mögliche Kollision erkannt werden, so kann diese gegebenenfalls durch eine Anpassung der Pfadplanung erreicht werden oder durch einen Eingriff der Aktionsplanung, indem ein neue Zielpose zur Entfernung vom Objekt vorgeben wird.

Das in Abbildung 5 skizzierte Konzept wurde, aufgrund der limitieren Rechenleistung auf dem Roboter, nicht vollständig im Rahmen des Projektes umgesetzt, die Funktionen und Module aber einzeln realisiert, validiert und untersucht. Basierend hierauf scheint es technisch mög-

lich, das Gesamtkonzept für den vollautonomen Betrieb eines Flugroboters mit einem entsprechend leistungsfähigen OBC sowie weiteren Entwicklungsarbeiten für eine effizientere und robustere Softwareimplementierung zu realisieren.

Die Auslagerung von einigen Softwaremodulen, wie z. B. der Pfadplanung, auf die GCS oder in eine Cloud wäre möglich, jedoch müsste hier die Verbindung die entsprechende Bandbreite und Latenz für eine zeitnahe Übertragung aller erforderlichen Daten bieten. Bei einem Verbindungsabbruch würde der Roboter zu dem nicht mehr über alle Fähigkeiten für den autonomen Betrieb verfügen.

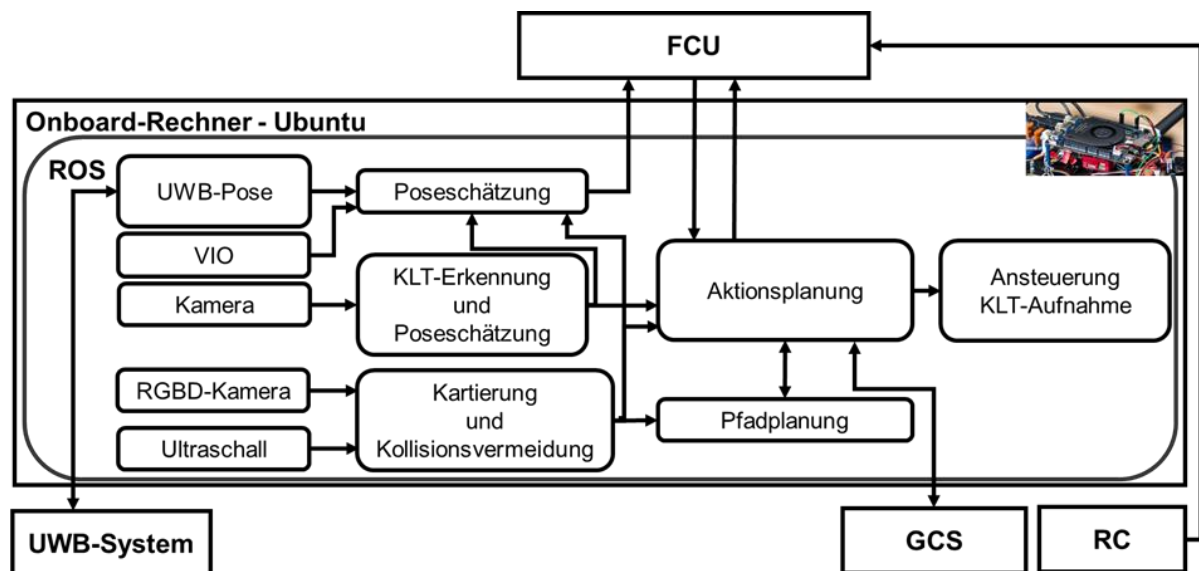


Abbildung 5: Vorgeschlagenes Softwarekonzept für den vollautonomen Betrieb

Als beispielhafte Realisierung der Auftragseinlastung und Überwachung wurde eine graphische Benutzeroberfläche (engl. Graphical User Interface (GUI)) implementiert. Diese ist in Abbildung 6 dargestellt. Links, im Feld „Drone Data“, werden aktuelle Zustandsinformationen des Flugroboters angezeigt. Dies sind der generelle Systemstatus, wie z. B. der aktuelle Modus des Roboters (im Bild „Offboard“), die aktuelle Poseschätzung der FCU, die aktuell vorgegebene Zielpose des Multicopters, sowie die Poseschätzung des externen Systems (z. B. des UWB-Systems), mit dem der EKF der FCU gestützt wird. Im Abschnitt „Drone Control“ können Befehle an den Roboter gesandt werden. Neben einem einfachen Start (Takeoff) und einer Landung (Land), ist es auch möglich eine Sequenz von Aktionen an die Aktionsplanung des Flugroboters zu übermitteln, die dann von diesem nacheinander ausgeführt werden. Dies umfasst neben einfachen Aktionen wie Starten, Landen und dem Anflug von Zielposen auch die Aufnahme und das Ablegen von KLTs an bestimmten Zielorten. Auf der rechten Seite werden alle angeflogenen und vorgegebenen Zielposen des Roboters sowie die aktuelle Pose des Roboters in einer zweidimensionalen Ansicht visualisiert.

Die GUI wurde basierend auf der Programmiersprache Python unter Nutzung der Bibliotheken „tkinter“ [25] und „OpenCV“ [26] implementiert. Die Kommunikation mit dem OBC erfolgt basierend auf dem ROS (Topics / Actions / Services).

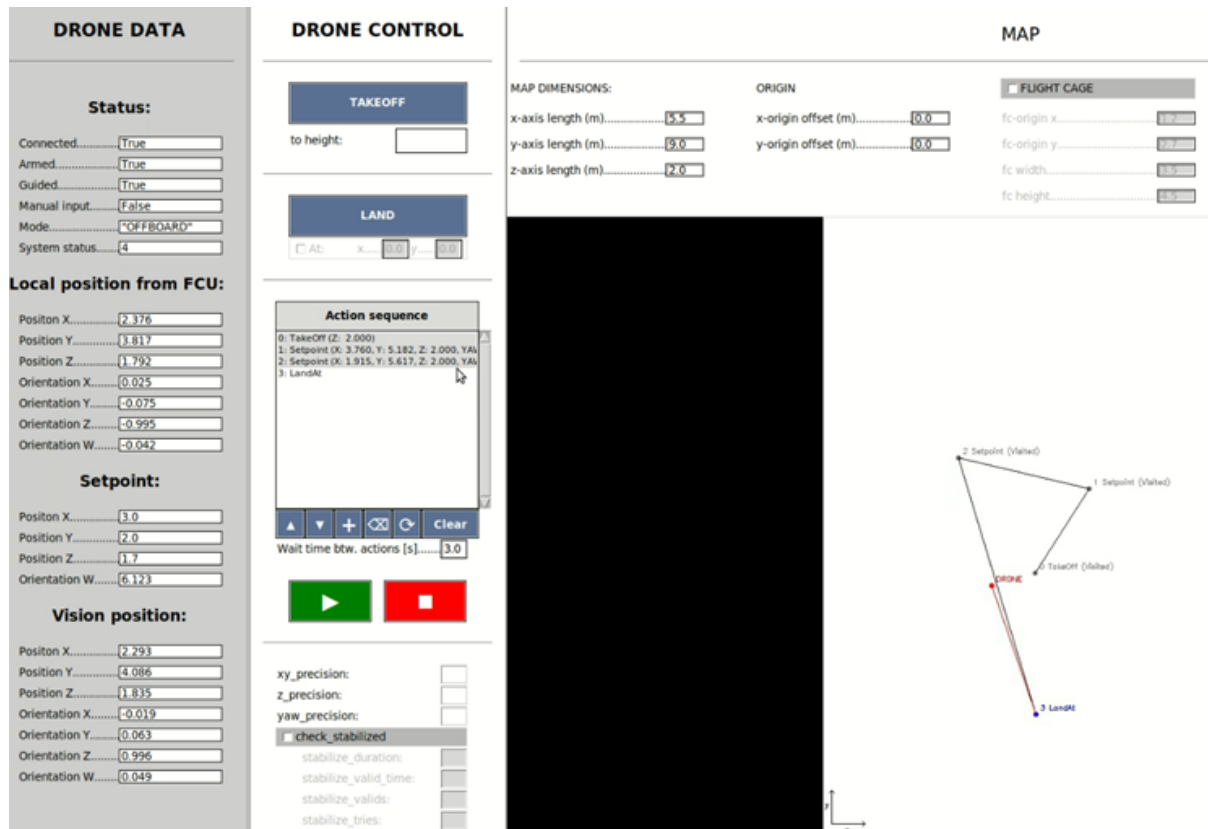


Abbildung 6: Im Rahmen des Projekts implementierte GUI zur Steuerung und Überwachung des Flugroboters

3.3.3 Simulationsumgebung

Zum Testen und Validieren von Software und Funktionalitäten bietet die Nutzung einer Simulation große Vorteile. Insbesondere bei Systemen, die durch Fehler bzw. Fehlverhalten stark beschädigt werden oder Schäden anrichten können, ist die Nutzung einer Simulation hilfreich. Da beim Absturz oder bei Kollisionen eines Flugroboters meist Schäden am Flugroboter entstehen, wurden alle implementierten und genutzten Softwaremodule vor dem Einsatz auf einer realen Drohne in der Robotik-Simulationssoftware „Gazebo“ getestet [27, 20]. PX4 bietet hier eine enge Verzahnung der FCU-Software mit der Simulation durch die Möglichkeit, die FCU-Software als Software-in-the-Loop in Verknüpfung mit der simulierten Drohne auszuführen [20]. Zusätzlich bietet die Simulationsumgebung Gazebo die Möglichkeit Sensoren, wie z. B. Kameras, zu simulieren, sodass sich auch Autonomie-Funktionalitäten, wie die Umgebungskartierung und Kollisionsvermeidung in der Simulation getestet und erprobt werden können. In Abbildung 7 sind beispielhaft zwei Bilder aus Simulationen, die im Rahmen des Projektes durchgeführt wurden dargestellt. Links ist ein Überblick über eine eigens erstellte simulierte Lagerumgebung zu sehen. Rechts ist das Bild einer simulierten Kamera auf dem Flugroboter in der Lagerumgebung zu sehen. In diesem Kamerabild wurden mithilfe eines Objekterkennungsalgorithmus KLTs erkannt.

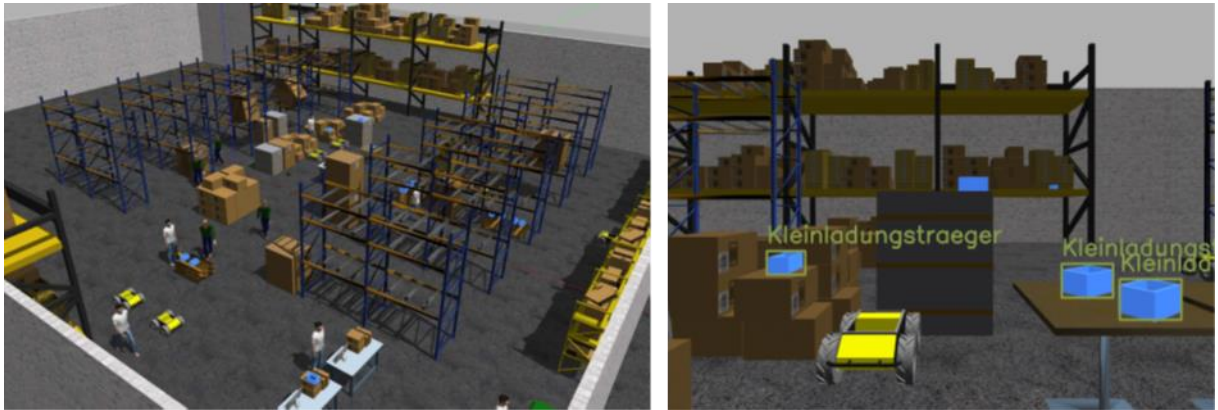


Abbildung 7: Beispielhafte Simulationsumgebung; Links: Überblick über eine Lagerumgebung die in der Simulation erstellt wurde; Rechts: Simuliertes Kamerabild eines Flugroboters in der Lagerumgebung mit im Bild erkannten KLTs

3.3.4 3D-Druck des Drohnenrahmens

Der Rahmen und die Lastaufnahme des erarbeiteten Demonstrators können nicht beliebig an andere Transportobjekte angepasst werden. Die Möglichkeit zur flexibleren Gestaltung und Anpassung des Drohnenrahmens mittels 3D-Druck wurde daher ebenfalls im Projekt analysiert. Hierfür wurde untersucht, wie ein Drohnenrahmen mit einem handelsüblichen 3D-Druckers gefertigt werden kann und welche Flugeigenschaften ein Flugroboter basierend auf dem 3D-gedruckten Rahmen aufweist. Der Rahmen, wie in Abbildung 8 dargestellt, wurde für die Untersuchung so entworfen, dass der Transport eines KLTs (RL-KLT 3147) möglich wäre. Es wurde ein Quadrocopter konstruiert und gefertigt, um das grundlegende Vorgehen und Prinzip für einen 3D-gedruckten Rahmen zu validieren. Für einen Hexacopter, wie dem Demonstrator, wäre es erforderlich gewesen, im Vergleich zum Quadrocopter, wesentlich mehr einzelne Elemente des Rahmens zu konstruieren und zu drucken.

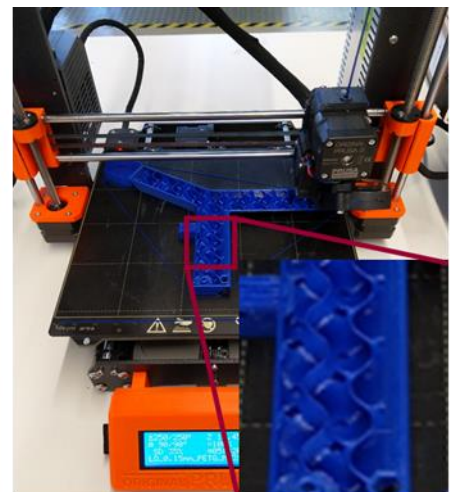
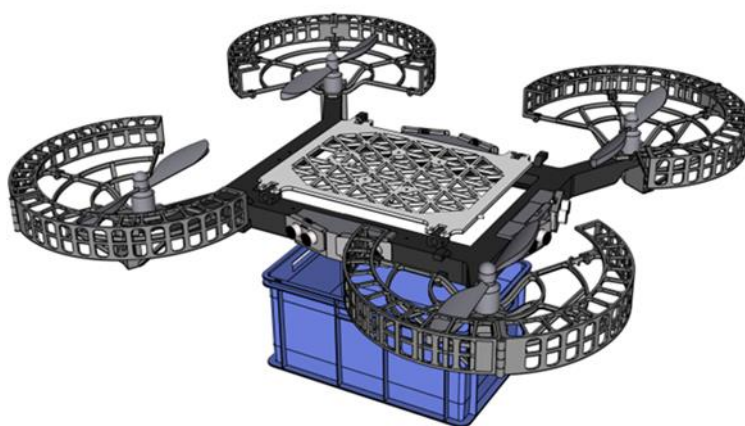


Abbildung 8: Links: Konstruierter Multicopterrahmen zum 3D-Druck einschließlich weiterer Elemente wie Propellerschutzkäfigen sowie Ultraschallsensoren zur Absicherung; Rechts: 3D-Druck eines Teilstücks des Rahmens und gedruckte Stützstruktur im inneren des Rahmens

Der 3D-Druck erfolgte mittels Fused Deposition Modeling (FDM) und dem Material PLA. Der Rahmen wurde hierbei nicht aus Vollmaterial, sondern mit einer Stützstruktur im Inneren gedruckt (Füllung: 25 %).

Aufgrund des begrenzten Druckraums des 3D-Druckers wurde der Rahmen in mehreren Teilen gedruckt, welche anschließen verklebt und mit Zylinderstiften gefügt wurden. Das Prinzip ist in Abbildung 9 links dargestellt. Der verjüngte Abschnitt wird mit geeignetem Klebstoff bestrichen und in die Einbuchtung des zweiten Teils gesteckt. Anschließend werden beide Teile zusätzlich mit einem Zylinderstift fixiert, der in die Löcher gepresst wird. Insgesamt wurde der Rahmen des Quadrocopters aus sechs Einzelstücken gebaut.

Ein fertig gedruckter Rahmen mit einigen Elektronikkomponenten ist in Abbildung 9 rechts zu sehen. Zur Befestigung der für den Flug erforderlichen Elektronik wurde auf dem Rahmen noch eine Acrylplatte angebracht.

Der 3D-gedruckte Rahmen im Flug ist in Abbildung 10 gezeigt. Hierbei sind zusätzlich Komponenten zur Absicherung angebracht. Die Absicherung des autonomen Fluges wird in Abschnitt 3.5, auch basierend auf Versuchen mit diesem Aufbau, beschrieben und bewertet.

Insgesamt hat der 3D-gedruckte Rahmen einschließlich der Elektronik und Akku (LiPo 4S, 4000 mAh) ein Gewicht von ca. 1950 Gramm, wovon der Akku ca. 400 Gramm wiegt.

Das Flugverhalten des Quadrocopters basierend auf dem 3D-gedruckten Rahmen ist als gut zu bewerten. Betrachtet man in Abbildung 11 die Vibrationsmetrik ([28]), die die Auswirkungen von Vibrationen auf die Inertiale Messeinheit der FCU angibt, der PX4-Software aus Fluglogs im Vergleich zum Demonstrator (Abbildung 12), der ein sehr gutes Flugverhalten mit geringen Vibrationen aufweist, so sind die Vibrationen nur geringfügig höher. Je kleiner der Wert dieser Metrik ist, desto stören Vibration beeinflussen die Sensoren der FCU.

Jedoch entspricht das Gewicht des Quadrocopters basierend auf dem 3D-gedruckten Rahmen mit 1950 Gramm fast dem des Demonstrators (2050 Gramm, beide mit Akku). Zudem ist der Demonstrator ein Hexacopter mit vergleichbaren Motoren, Propellern und Größe, hat aber durch die zwei zusätzlichen Motoren wesentlich mehr Schub als der Quadrocopter. Für einen effizienten Einsatz eines Multicopters basierend auf einem 3D-gedruckten Rahmen sollte also dessen Gewicht soweit möglich reduziert und auch entsprechend in die Auslegung der Antriebsleistung des Multicopters einbezogen werden.

Die Untersuchungen zeigen aber, dass 3D-gedruckte Rahmen ein vielversprechender Weg für individuell gefertigte Multicopter mit guten Flugeigenschaften sind.

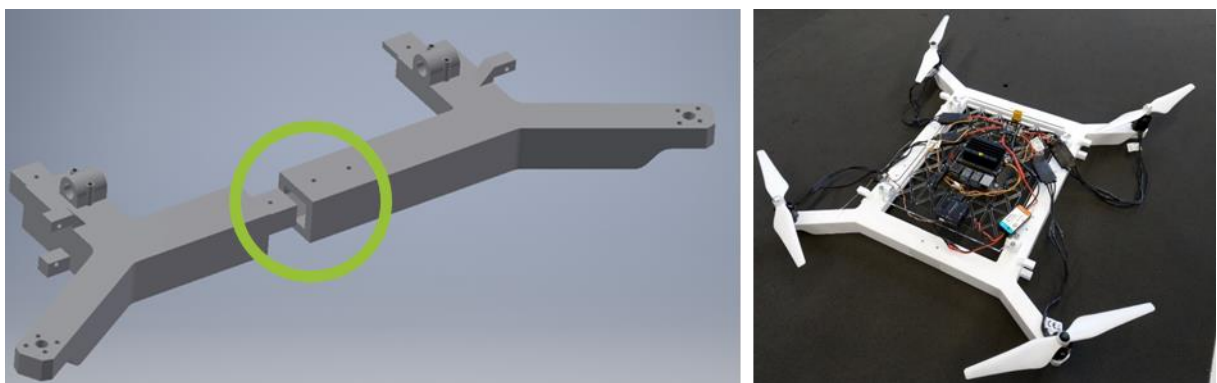


Abbildung 9: Links: CAD-Modell zweier einzelner Teile des 3D-gedruckten Rahmens die verklebt und mit einem Zylinderstift gefügt werden; Rechts: Teilweise aufgebaute Quadrocopter basierend auf dem 3D-gedruckten Rahmen

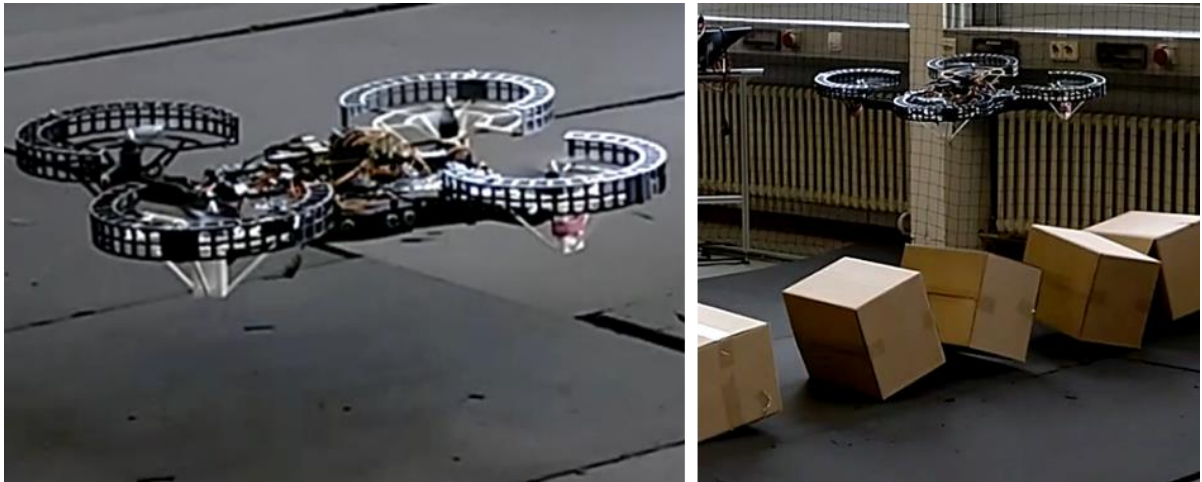


Abbildung 12: Quadrocopter basierend auf dem 3D-gedruckten Rahmen im Flug

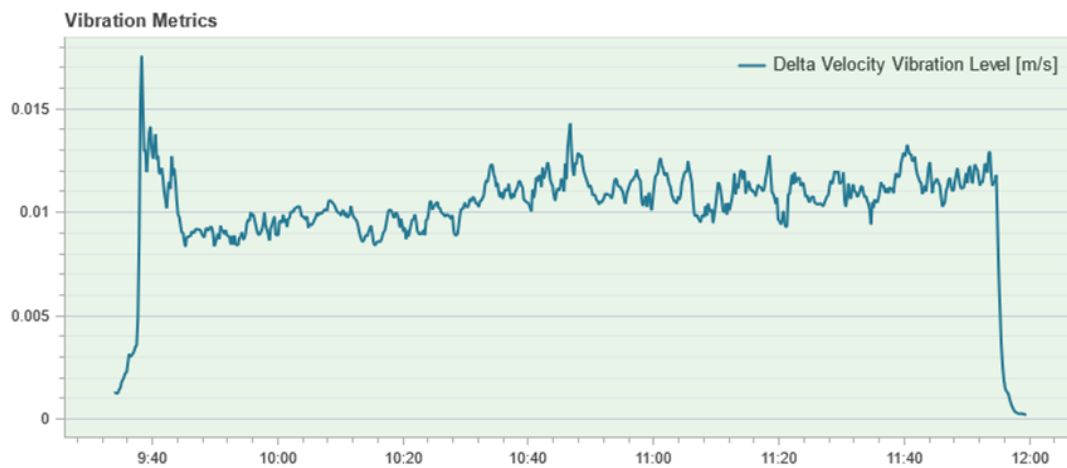


Abbildung 11: PX4-Vibrationsmetrik des Quadrocopters basierend auf dem 3D-gedruckten Rahmen

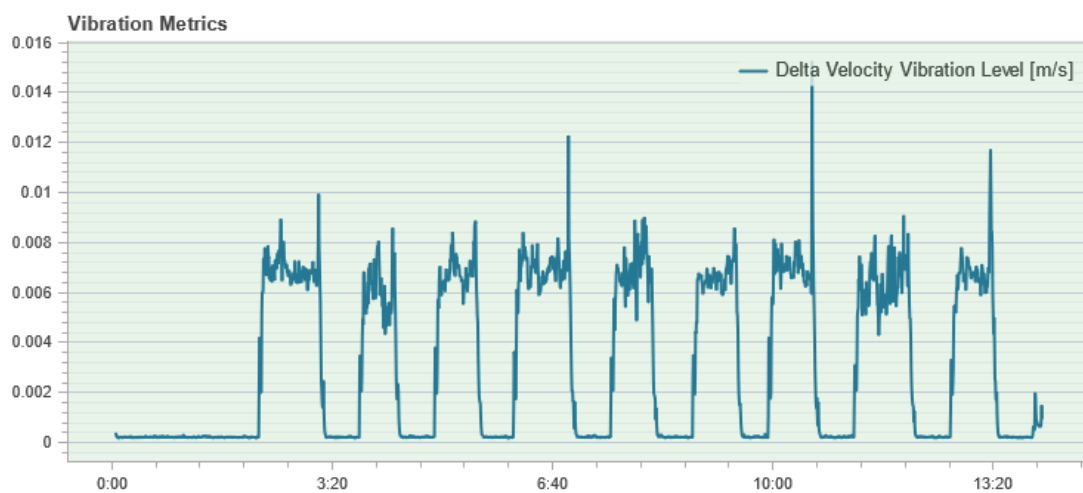


Abbildung 10: PX4-Vibrationsmetrik des Demonstrators während mehrerer Flüge, zwischen denen der Multicopter gelandet ist

3.3.5 Ladestation zum automatisierten Laden der Drohne

Ein weiterer Aspekt für den vollautomatisierten Betrieb erforderlich ist, ist das automatisierte Laden der Akkus.

Im Rahmen des Projekts wurde eine Ladestation für Drohnen durch die Firma Rosenberger Hochfrequenztechnik GmbH & Co. KG (Mitglied im Projektbegleitenden Ausschuss) für Experimente zur Verfügung gestellt. Die Ladestation ist in Abbildung 13 zu sehen. Sobald eine Drohne in der Station landet kann über Kontakte am Landegestell der Drohne und der Ladestation sowie entsprechender Elektronik auf der Drohne der Akku geladen werden. Die Drohne kann somit autonom zur Ladestation fliegen, dort landen und die Akkus werden automatisiert geladen.

Das Prinzip zum sicheren und präzisen Landen in der Station wurde analog zum Landen auf einem KLT umgesetzt (siehe Abschnitt 3.6), wobei hier der Einsatz von optischen Markern wie AprilTags [29, 30] zur präzisen Positionierung beim Landen gut geeignet ist. Die Drohne fliegt also über die Station, erkennt einen Marker in der Station und kann anschließend die Pose relativ zu diesem präzise und robust schätzen. Im Folgenden sinkt die Drohne in zwei Stufen nahe über die Station bzw. den Marker ab und landet schließlich in dieser. Durch die Führungen an der Station wird sichergestellt, dass die Kontaktierung zwischen den Ladekontakten am Landegestell und denen der Station zum erfolgreichen Laden vorhanden ist. Eine Drohne beim Ladevorgang nach einem erfolgreichen autonomen Anflug ist in Abbildung 13 rechts zu sehen.

Da die Ladkontakte unten am Landegestell der Drohne angebracht sind, wären für eine Kombination mit der Lastaufnahme zum Transport von KLTs noch weitere Entwicklungsarbeiten erforderlich, um beide Funktionsprinzipien funktionsfähig zu vereinen.

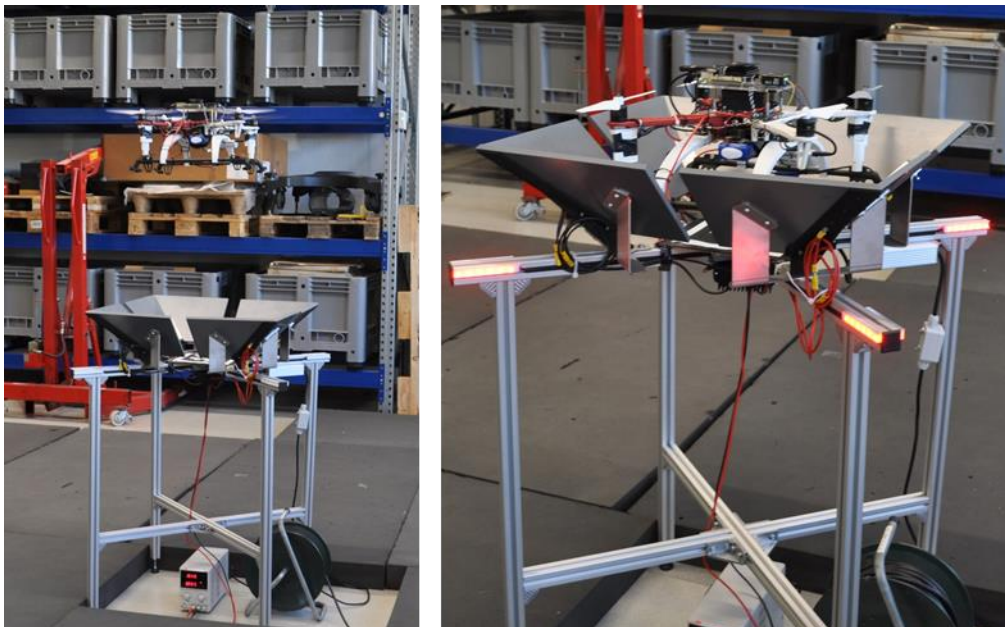


Abbildung 13: Links: Quadrocopter beim Anflug zur autonomen Landung in der Ladestation; Rechts: Quadrocopter nach der Landung in der Station, die roten Lichter zeigen an, dass der Akku der Drohne geladen wird

3.4 Poseschätzung für den autonomen Flug in Innenbereichen

Für einen sicheren und stabilen Flug des Roboters sowie für die Kollisionsvermeidung und Navigation, ist es essentiell, dass dieser seine Pose präzise und robust schätzt. Die präzise Schätzung der Pose ist auch für das Landen auf dem KLT relevant, da die Landung zur erfolgreichen Fixierung des KLTs mit einer Genauigkeit von wenigen Zentimetern erfolgen muss.

Wie im vorherigen Abschnitt beschrieben, erfolgt die Schätzung des Zustands des Flugroboters, einschließlich der Pose, mittels eines EKF durch die PX4-Software auf der FCU. Hierfür werden einerseits die in der FCU integrierten Sensoren genutzt, andererseits besteht die Möglichkeit, Posedaten eines externen Systems (das sich aber dennoch auf dem Roboter befinden kann) in den EKF einzuspeisen.

3.4.1 Sensoren, Systeme und Ansätze zur Posebestimmung

Im Stand von Forschung und Entwicklung gibt es eine Vielzahl von Systemen und Ansätzen zur Lokalisierung bzw. Posebestimmung von mobilen Robotern. In diesem Abschnitt werden potentiell zur Nutzung auf einer Drohne geeignete Sensoren, Systeme und Ansätze vorgestellt und hinsichtlich ihrer Eignung für das im Projekt betrachtete Szenario eingeordnet:

- **Inertiale Messeinheit** (engl. Inertial Measurement Unit (IMU)): Eine IMU setzt sich aus Beschleunigungssensoren (Accelerometer) und Drehratensensoren (Gyroskope) zusammen. Hierbei werden für jede der drei Raumachsen jeweils ein Accelerometer und ein Gyroskop genutzt. So kann die Beschleunigung und die Drehrate bezüglich jeder Raumrichtung gemessen werden. Durch einfache Integration der Drehraten kann die Orientierung im Raum abgeleitet werden, durch zweifache Integration der Beschleunigungen die Position im Raum. Aufgrund der Integration akkumulieren sich Messfehler bedingt durch Rauschen, Bias und weitere Störeffekte sodass eine von IMU-Messdaten abgeleitete Pose nur kurzfristig als genau und stabil betrachtet werden kann. Der sich akkumulierende Fehler wird Drift genannt. Dennoch bieten die Messdaten einen großen Mehrwert im Rahmen einer Sensordatenfusion zur Poseschätzung zusammen mit anderen Messdaten bzw. Sensoren. [31]
Eine IMU hat keinen Bezug zur Umgebung bzw. Infrastruktur, sondern misst direkt die Zustandsgrößen Beschleunigung und Drehrate des Flugroboters. In der verwendeten Pixracer-Flugsteuerung ist bereits eine IMU integriert. Die Messungen werden von der PX4-Software in die Zustandsschätzung des EKF einbezogen.
- **Magnetometer**: Ein Magnetometer kann auf Drohnen in Form eines Kompasses genutzt werden, sodass die Orientierung der Drohne in Bezug zum magnetischen Nordpol gemessen werden kann [31]. Für Einsatzszenarien in Gebäuden ist diese Art der Lagemessung ungeeignet, da durch die Gebäudestrukturen und weitere Störeffekte nur unzureichende Messung erzielt werden [32]. In der Pixracer-Flugsteuerung ist ein Magnetometer integriert, der Zustandsschätzer wurde aber so konfiguriert, dass die Messdaten nicht genutzt werden
- **Barometer**: Ein Barometer misst den Luftdruck und kann so zur Höhenbestimmung genutzt werden. Auch hier sind die Messdaten beim Einsatz in Innenbereichen fehler-

behaftet [32]. In der Pixracer-FCU ist ebenfalls ein Magnetometer integriert, der Zustandsschätzer wurde aber wiederum so konfiguriert, dass die Messdaten nicht genutzt werden.

- **Ultraschallentfernungsmessung:** Bei der Ultraschallentfernungsmessung wird die Laufzeit des Echos eines ausgesendeten Ultraschallsignals gemessen. Hierdurch kann mittels der Schallgeschwindigkeit im Medium Luft die Distanz zum Objekt, an welchem das ausgesendete Signal zurückreflektiert wurde, berechnet werden. Dieses Verfahren funktioniert robust, jedoch kann nur ein Echo und somit meist nur die Distanz zu dem Objekt, welches sich in kürzester Distanz zum Sensor im Erfassungskegel befindet, ermittelt werden. Die Reichweite beträgt im Medium Luft meist wenige Meter. Die Genauigkeit der Distanzmessung liegt im Bereich von Millimetern bis hin zu Zentimetern. [31]

Die Ultraschallentfernungsmessung kann zur Bestimmung der Höhe über dem Boden genutzt werden. Hier ist aber zu beachten, dass insbesondere in Innenbereichen Objekte auf dem Boden stehen können und so die Messung nicht mehr der Höhe über dem Boden entspricht. Ähnliches gilt für eine Distanzmessung zur Decke für eine Höhenbestimmung. Sobald die Deckenstruktur nicht einer Fläche entspricht, ist diese Art der Höhenbestimmung ungenau bzw. führt zu einer falschen Höhenbestimmung.

Zu Lokalisierung und Poseschätzung ist die Ultraschallsensorik für Flugroboter im Innenbereich also nur bedingt geeignet und wurde daher nicht für diesen Zweck im Projekt eingesetzt. Der Einsatz auf dem Roboter für die Hindernisdetektion und -lokalisierung ist jedoch sinnvoll und wird in Abschnitt 3.5. betrachtet. Mit den Sensoren können im Bereich des Schallkegels Hindernisse detektiert werden, während die Sensoren relativ klein sind, ein geringes Gewicht haben und verhältnismäßig wenig Energie benötigen.

- **Laserentfernungsmessung und LIDAR -Sensorik:** Hierbei wird mittels eines Laserstrahls die Distanz zu Objekten in der Umgebung gemessen. Das Funktionsprinzip ist ähnlich zur Ultraschallentfernungsmessung. Jedoch wird statt dem Ultraschall ein Laserstrahl genutzt, wodurch Messung auch für weitere Distanzen (mehrere Meter bis zu hundert Meter) bei wesentlich höherer Genauigkeit (im Bereich von Millimetern) möglich sind. Es gibt unterschiedliche Bauformen, von der Punkt-Laserentfernungsmessung über 2D-LIDAR bis hin zu 3D-LIDAR-Sensoren, die Umgebung rund um den Sensor mit mehreren tausenden 3D-Punkten präzise abbilden. [31]

Punkt-Laserentfernungsmesser bieten verglichen mit der Ultraschallentfernungsmessung eine höhere Messgenauigkeit und Reichweite, decken jedoch nur vergleichsweise kleinen Bereich für eine Hindernisdetektion ab. Senkrecht zur Flugebene ausgerichtet können sie ebenfalls zur Bestimmung der Höhe des Flugroboters genutzt werden, im Innenbereich entstehen hier aber, wie bereits beschrieben, Probleme, sobald der Flugroboter Objekte überfliegt.

Eine präzise Lokalisierung und Poseschätzung für Flugroboter basierend auf den Messungen von 2D- und 3D-LIDAR-Sensoren ist möglich [33, 34]. 3D-Lidar-Sensoren können auf Drohnen eingesetzt werden, jedoch sind diese Sensoren relativ teuer, schwer und die Verarbeitung der 3D-Punktwolke erfordert eine entsprechende Rechenleistung. 2D-Lidar-Sensoren decken aber nur eine Ebene ab, was für die dreidimensionale Bewegungsfähigkeit und Navigation des Flugroboters nicht vollumfänglich geeignet ist.

Sensoren zur Laserentfernungsmessung und LIDAR-Sensorik wurden daher nicht weiter im Projekt betrachtet.

- **Kameras:** Kameras wandeln das von der Umgebung reflektierte Licht mittels eines Bildsensors in ein digitales Bild [31]. Es kann ein relativ großer Bereich der Umgebung mit einem Kamerabild abgedeckt werden. Mittels Stereokameras und RGB-D Kameras kann zu dem die Entfernung von Objekten zur Kamera bestimmt werden und so ein dreidimensionales Abbild der Umgebung generiert werden [31]. Dies ist vor allem für die Hindernisdetektion und Umgebungskartierung vorteilhaft.

Die Lokalisierung mittels einer Kamera bzw. mittels derer Bilder ist möglich, z. B. durch visuelle Odometrie (VO) [35] und visuellen SLAM (V-SLAM) [36] oder mittels bekannter visueller Marker [30]. Bei der VO und beim V-SLAM besteht die Möglichkeit, dass sich Messfehler mit der Zeit akkumulieren, da die Posebestimmung bzw. -schätzung relativ erfolgt, also kontinuierlich in Bezug auf die Ausgangspose.

Kameras sind relativ günstig verglichen mit LIDAR-Sensorik und bieten ein geringes Gewicht bei einer relativ geringen Baugröße sowie eine große Abdeckung der Umgebung für die Umgebungserfassung und Objektdetektion. Zur Verarbeitung der Bilddaten sind jedoch relativ rechenaufwändige Algorithmen erforderlich. Im Rahmen des Projekts wurden unterschiedliche Kameras eingesetzt und untersucht, sowohl zur Lokalisierung und Poseschätzung als auch zur Hindernisdetektion, Umgebungskartierung und Objekterkennung.

- **Globale Navigationssatelliten Systeme (GNSS):** Hierbei wird mittels Laufzeitmessung von Funksignalen, die von Satelliten ausgesandt werden, die Position eines entsprechenden Empfängers auf der Erdoberfläche bestimmt [31]. Die Art der Positionsbestimmung wird im Außenbereich für Drohnen meist genutzt, ist jedoch im Innenbereich nicht verfügbar, da die Funksignale durch die Gebäudestrukturen stark beeinflusst (gedämpft, reflektiert etc.) werden.
- **Ultrabreitband-Funksystem (UWB-System):** Bei dieser Technologie werden in der Umgebung Funkknoten (Anker) installiert, sodass entsprechende mobile Funkknoten (Tags) durch unterschiedliche Messprinzipien relativ präzise (im Bereich von Zentimetern bis Dezimetern) dreidimensional lokalisiert werden können [32, 37]. Metallische Objekte, die die Funksignale beeinflussen, verschlechtern das Lokalisierungsergebnis [38]. Die Lokalisierung der Tags im Raum erfolgt absolut, d. h. in Bezug auf ein festes Koordinatensystem im Raum. Dies ist vorteilhaft, da so die Position im Raum immer bekannt ist und keine Akkumulation von Messfehlern stattfindet. Die Lokalisierung mittels UWB-Systemen ist gut für Innenbereiche geeignet und findet bereits industrielle Anwendung [37]. Im Projekt wurde diese Art der Lokalisierung bzw. eine auf der Lokalisierung aufbauende Posebestimmung genutzt und analysiert.

Zusammengefasst wurden folgende Sensoren und die Fusion der Messdaten dieser im Rahmen der Zustandsschätzung auf der FCU für die Poseschätzung des Flugroboters im Projekt genutzt und untersucht:

- IMU, diese ist bereits fest in der genutzten Pixracer-Flugsteuerung integriert
- UWB-Funkortungssystem (Fa. Kinexon GmbH [37]) mit mehreren Tags auf dem Flugroboter und 12 Anker in der Umgebung
- Intel RealSense Tracking Camera T265 [39], Stereo-Kamera mit V-SLAM-Lösung; der V-SLAM wird auf einem eingebetteten Chip in der Kamera berechnet und die so geschätzte Pose ist zur weiteren Nutzung direkt verfügbar

- Intel RealSense Depth Camera D435 [40], Stereokamera zur 3D-Erfassung der Umgebung; diese wurde einerseits in Kombination mit einem VO-Ansatz aus dem Stand der Forschung eingesetzt, andererseits zur Erstellung einer 3D-Karte der Umgebung und Lokalisierung in dieser
- Nach unten (zum Boden) blickende 2D-Kamera (Logitech C207), zur Erkennung und präzisen Positionierung über dem KLT und zur Posebestimmung mittels optischer Marker

3.4.2 Posebestimmung mithilfe eines UWB-Funkortungssystems

Die Posebestimmung mittels eines UWB-Funkortungssystems wurde durch die Nutzung mehrerer Tags umgesetzt. Diese Tags werden mithilfe von Anker und unterschiedlichen Funklokalisierungsprinzipien im Raum lokalisiert [37]. Im Rahmen des Projektes wurde ein UWB-System der Firma Kinexon GmbH genutzt welches eine Lokalisierungsgenauigkeit der Tags von bis zu 10 Zentimetern bietet [37]. Insgesamt besteht das UWB-System aus 12 Anker die an den Rändern eines rechteckigen Versuchsbereichs mit einer Fläche von ca. 8 x 5,5 Metern und einer Höhe von 4 Metern fest installiert sind. Für eine optimale 3D-Lokalisierung der Tags sollten sich diese innerhalb des Bereiches, den die Anker umschließen befinden und es sollten sich keine weiteren Objekte zwischen den Tags und den Anker befinden.

Die Berechnung der Lokalisierung der Tags erfolgt auf einem externen Rechner und die so bestimmte 3D-Position jedes Tags wird mittels WLAN an den OBC des Flugroboters übertragen. Am Demonstrator wurde die Poseschätzung mit drei Tags umgesetzt und analysiert. Die Tags sollten voneinander möglichst weit entfernt angebracht werden, um die Beeinflussung der Funksignale die von anderen Tags gesendet bzw. empfangen werden zu minimieren. Die Tags sind an definierten Position an den Armen des Flugroboters angebracht, wie es in Abbildung 14 links gezeigt ist

Durch die bekannte und symmetrische Anbringung der Tags am Flugroboter kann die 3D-Position des Flugroboters durch die Mittelwertbildung der drei Tag-Position berechnet werden.

Die Berechnung des Gierwinkels (auch Yaw-Winkel), also die Rotation des Flugroboters um dessen z-Achse, ist für die Stützung der Zustandsschätzung durch den EKF der FCU ausreichend. Die anderen beiden Orientierungskomponenten, Roll- und Nickwinkel, werden ausreichend genau mittels der IMU durch den EKF geschätzt, da hier auch keine wesentlichen Änderungen erfolgen sollten.

Für die Berechnung des Gierwinkels sind minimal zwei Tags erforderlich, eine höhere Anzahl, wie z. B. drei beim Hexacopter, steigert jedoch die Robustheit und Genauigkeit durch die zusätzlichen Messinformationen. Das Prinzip der Bestimmung des Gierwinkels basierend auf den Tag-Positionen ist in Abbildung 14 rechts skizziert. Die Tags sind durch orangefarbene Rechtecke skizziert. Gesucht ist der Winkel α , der die Drehung des Flugroboters um die z-Achse (die senkrecht aus der x-y-Ebene herauszeigt) darstellt. Mittels der bekannten x- und y-Positionen der einzelnen Tags im Raum kann durch Differenzbildung die Strecke zwischen zwei Tags berechnet werden (orange Linie). Mithilfe der Arcustangens-Funktion (atan2) kann so durch die y-Differenz und x-Differenz der Tag-Positionen der Winkel β zwischen der Strecke und der x-Achse (in grau angedeutet) berechnet werden. Der Winkel γ ist bekannt, da die sechs Arme symmetrisch angebracht sind und beträgt 120 Grad. Aufgrund der gleichen

Länge der Arme ergibt sich ein gleichschenkliges Dreieck, sodass der Winkel δ 30 Grad beträgt. Der Winkel ϵ ergibt sich somit aus der Differenz der Winkel β und δ . Der Winkel α lässt sich somit aus dem Winkel ϵ sowie des halben Winkels zwischen zwei Armen η , der 30 Grad beträgt, berechnen. Zusammengefasst kann der Gierwinkel des Flugroboters basierend auf zwei Tag-Positionen, (x_1, y_1) und (x_2, y_2) , durch die Beziehung

$$\alpha = \text{atan2}(y_1 - y_2, x_1 - x_2) - \delta + \eta \quad 3.1$$

berechnet werden. Die Berechnung erfolgt analog für die beiden weiteren Tag-Paarungen. Der Gierwinkel des Flugroboters kann anschließend über die Mittelung der drei berechneten Winkel mittels der Beziehung

$$\bar{\alpha} = \text{atan2}\left(\frac{1}{3} \sum_{i=1}^3 \sin \alpha_i, \frac{1}{3} \sum_{i=1}^3 \cos \alpha_i\right) \quad 3.2$$

erfolgen [41].

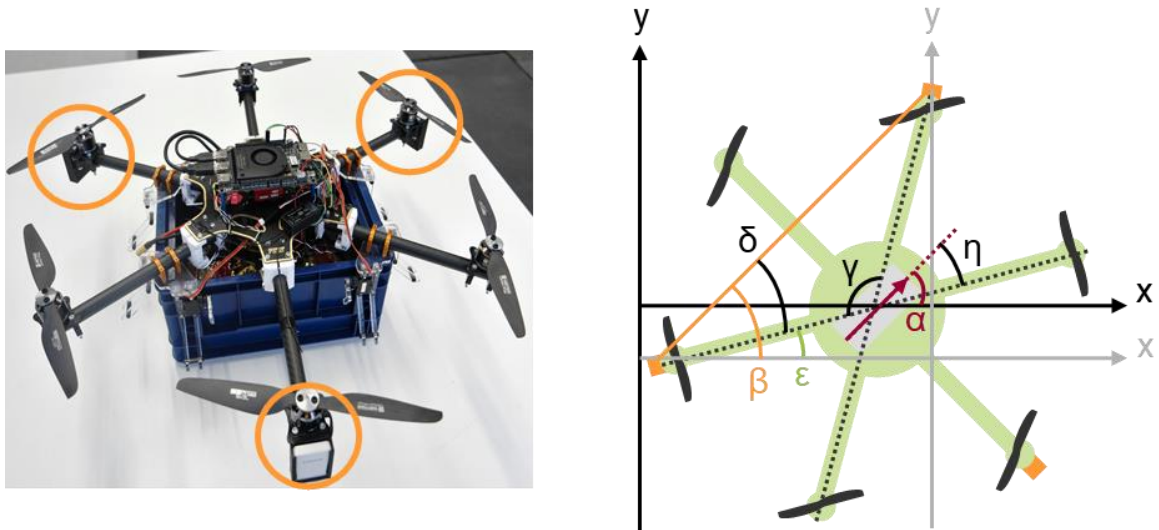


Abbildung 14: Links: Bild des Demonstrators, die Positionen der Tags am Demonstrators sind durch orangefarbene Kreise gekennzeichnet. Rechts: Skizze zur Berechnung des Gierwinkels des Multicopters basierend auf den gemessenen Tag-Positionen.

3.4.3 Kamerabasierte Poseschätzung

Die Poseschätzung von mobilen Robotern mittels Kameras kann durch visuelle Odometrie (VO) bzw. visuelle-inertiale Odometrie (VIO), V-SLAM und mittels optischer Marker realisiert werden.

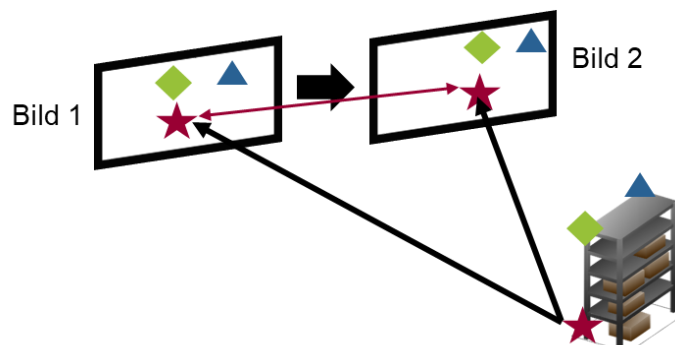


Abbildung 15: Prinzip der Posebestimmung mittels Kamerabildern basierend auf Bildmerkmalen

Bei der VO wird basierend auf Bildmerkmalen, die in den Kamerabildern detektiert werden, die Bewegung der Kamera geschätzt. Dies ist skizzenhaft in Abbildung 15 dargestellt. Zunächst werden in einem Kamerabild Bildmerkmale (Features) erkannt. In einem zweiten Kamerabild aus einer anderen Perspektive, beispielsweise erzielt durch eine Kamerabewegung, werden wieder Bildmerkmale erkannt und es wird eine Merkmalszuordnung der Merkmale aus beiden Bildern durchgeführt. Nun wird die Veränderung der Position im Bild von jeweils zugeordneten Merkmalen ausgewertet. Unter der Annahme, dass die Umgebung statisch ist, und mithilfe der intrinsischen Kameraparameter sowie eines geeigneten Kameramodells für die Kamera, welches modelliert wie die Entstehung eines Bildes erfolgt, kann berechnet werden, aus welchen unterschiedlichen Posen die Bilder aufgenommen wurden, also die Translation und Rotation der Kamera zwischen den Bildern. [42, 35]

Die VO kann mit einer monokularen Kamera sowie mit Stereokameras und RGB-D Kameras durchgeführt werden. Der Nachteil bei der Nutzung von monokularen Kameras ist, dass die Berechnung der Skalierung der Kamerabewegung nicht möglich ist, da keine metrischen Informationen in den Bildern über die Umgebung enthalten sind. Durch die Nutzung und Fusion von Inertialsensorik (IMU) kann die Skalierung berechnet und die Robustheit und Genauigkeit der Poseschätzung, insbesondere bei Drehbewegungen, verbessert werden [35, 43]. Eine weitere Verbesserung kann durch die Nutzung einer Stereokamera erfolgen. Durch eine vorherige Ermittlung der Translation und Rotation der beiden starren Kameras zueinander (Stereokalibrierung, [42]), kann die exakte 3D-Position von korrespondierenden Pixeln in den beiden Bildern berechnet werden. Natürlich ist auch hier die Kombination mit einer IMU möglich, um die Poseschätzung zu optimieren.

Bei einem V-SLAM-Algorithmus werden die erkannten Bildmerkmale nicht nur kurzzeitig lokal für die Poseschätzung gespeichert, sondern langfristig in einer Karte erfasst. Hier ist es das Ziel, nicht nur kontinuierlich die Pose der Kamera zu berechnen, wie es das Ziel der VO ist, sondern auch gleichzeitig eine möglichst exakte Karte der Umgebung zu erstellen (die Karte muss hierbei nicht die Umgebung exakt nachbilden, auch eine exakte Kartierung der erkannten Bildmerkmale kann das Ziel sein). Die Grenze zwischen VO und V-SLAM sind jedoch fließend [36].

Ein Problem, das bei allen Varianten auftritt, ist die Akkumulierung von Fehlern mit der Zeit und somit eine Drift der Poseschätzung [36, 35]. Ein Ansatz diese Drift zu eliminieren bzw. zu vermindern ist die Schleifenschließung [36]. Hierbei werden bereits besuchte bzw. kartierte Orte auf Basis der aktuellen im Bild vorhanden und der kartierten Bildmerkmale wiedererkannt. Es wird geschlussfolgert, dass sich die Kamera also wieder am selben Ort befindet und die Drift der geschätzten Pose kann eliminiert werden. Insbesondere bei Ansätzen, die auf einem Graphen-basierten SLAM aufbauen, kann die gesamte Trajektorie auch rückwirkend und somit auch die Karte optimiert werden [36].

Die Posebestimmung basierend auf optischen Markern, wie z. B. AprilTags [29, 30], die mit einer Kamera erkannt werden, ist nur bedingt für den Einsatz während des Fluges geeignet. Es müssten durchgehend optische Marker im Kamerabild sichtbar sein, sodass eine große Anzahl dieser in der Umgebung angebracht werden müsste und nicht verdeckt werden dürfte. Für die Erkennung von und Posebestimmung relativ zu bestimmten Objekten und Orten eignen sich optische Marker aber durchaus. Dies wird im Kontext der Erkennung und Handhabung der KLTs mit dem Flugroboter in Abschnitt 3.6.3 ausgeführt.

Da VO bzw. V-SLAM nicht abhängig von spezifischen Merkmalen oder Markern in der Umgebung sind und somit flexibel eingesetzt werden können, wurden diese Variante der kamerabasierten Posebestimmung während Fluges im Rahmen des Projektes untersucht.

Im Stand der Technik und Forschung sind unterschiedliche Ansätze und Lösungen für die VO/VIO und V-SLAM vorhanden, ein Vergleich wichtiger Ansätze wird in [36] präsentiert.

Im Rahmen des Projektes wurde einerseits die Intel Tracking Camera T265, welche einen V-SLAM durchführt, eingesetzt und untersucht. Die T265-Kamera besteht aus zwei Fischaugenkameras, einer IMU sowie einem speziellen Chip, der eine V-SLAM-basierte Poseschätzung direkt auf Basis der Sensordaten berechnet [39]. Die Poseschätzung wird also direkt von der Kamera bereitgestellt. Der dahinterliegende V-SLAM-Algorithmus ist aber nicht zugänglich bzw. einsehbar. Intel bietet Softwarepakete zur Einbindung der Kamera mittels des ROS. Eine an einem Flugroboter montierte T265-Kamera für Experimente im Rahmen des Projektes ist in Abbildung 16 zu sehen.

Andererseits wurde der VINS-Fusion-Algorithmus [35, 43–45] untersucht, welcher speziell für Drohnen entwickelt wurde. Der Algorithmus ist also recheneffizient und kann auf dem OBC ausgeführt werden. Zudem liefert er Poseschätzungen in fest definierbaren Zeitintervallen, sodass auch im Flug eine kontinuierliche Aktualisierung der Pose garantiert wird. Der VIO-Algorithmus kann sowohl mit monokularen Kameras also auch mit Stereokameras sowie zusätzlich mit einer IMU eingesetzt werden. In Evaluation und Tests auf frei verfügbaren Datensätzen konnte der Algorithmus die Pose der Kamera bzw. der Drohne mit einer sehr hohen



Abbildung 16: Multicopter mit D435-Kamera (oben) und T265-Kamera (unten)

Genauigkeit mit Fehlern im Zentimeterbereich schätzen [35]. Neben dem VIO-Algorithmus von VINS-Fusion sind auch zugehörige Algorithmen für eine Schleifenschließung und die Fusion von GPS-Daten verfügbar. Die Sensorik (Kamera, IMU) sowie der Rechner, auf dem der Algorithmus die VIO berechnet, können vom Nutzer ausgewählt werden. Dieser Ansatz bietet verglichen mit der T265-Kamera größere Freiheiten und Eingriffsmöglichkeiten, jedoch muss die Parametrisierung und Kalibrierung vom Nutzer selbst durchgeführt werden.

3.4.4 Evaluation und Bewertung der Poseschätzung

In Abbildung 17 sind die Ergebnisse der Lokalisierung für die x- und y-Positionen eines Fluges des Demonstrators basierend auf dem UWB-System, der T265-Kamera sowie die Poseschätzung des EKF der Flugsteuerung, die mit der UWB-Daten gestützt wurde dargestellt. Die wahre Trajektorie des Fluges ist in grün dargestellt. Es wurde dreimal die grüne quadratische Trajektorie abgefolgt und die Messdaten währenddessen gespeichert. Diese grüne, wahre Trajektorie wurde mithilfe eines Motion Tracking System (auch Motion

Capturing System genannt) der Firma Advanced Realtime Tracking GmbH & Co. KG aufgezeichnet und weist eine Genauigkeit im Millimeterbereich auf. Aufgrund des Messprinzips des Motion Tracking Systems, bei dem definierte Marker am Fluroboter mit mehreren Kameras, die in der Umgebung angebracht sind, zur Posebestimmung erfasst werden, ist die Nutzung bei größeren Flächen mit vielen Verdeckungen, wie in einem Lager, aber nicht sinnvoll.

Es ist erkennbar, dass die Poseschätzung der T265-Kamera nicht gut mit der wahren Trajektorie übereinstimmt. Zudem hat die Trajektorie bei jedem der drei aufeinanderfolgenden Flüge einen ähnlichen Verlauf, sodass auf eine Fehlschätzung der Pose, bedingt durch äußere Faktoren wie eine schlechte Bildmerkmalsverteilung oder eine ungünstige Beleuchtung, geschlossen werden kann. Insbesondere für den unteren Abschnitt des Quadrats ($y = 3,5$ m) und den rechten Abschnitt ($x = 4,5$ m) stimmt die Schätzung der T265-Kamera gut mit der wahren Trajektorie überein, was die Hypothese der teilweise ungünstigen Umgebungsbedingungen stützt.

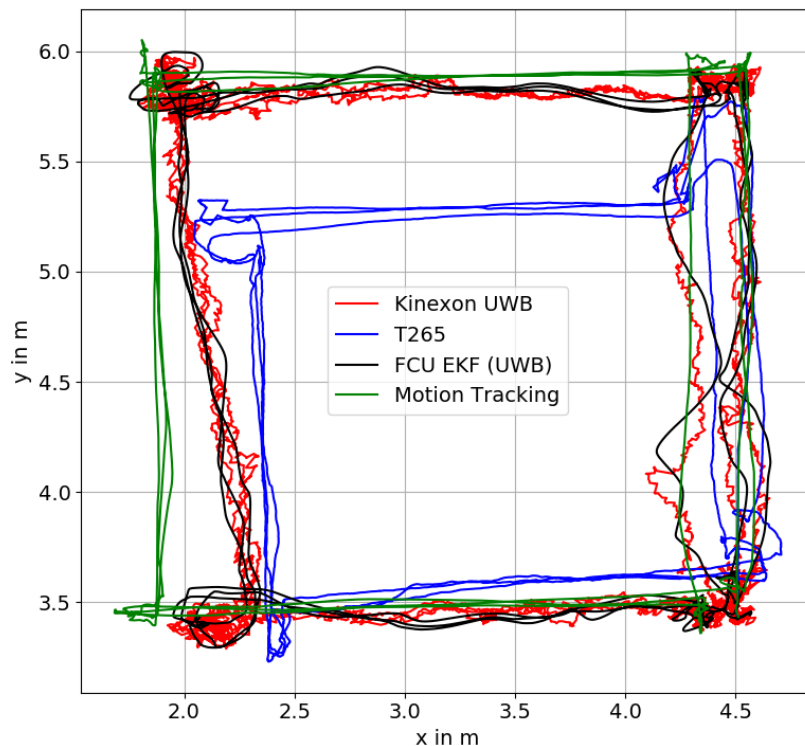


Abbildung 17: Vergleich der Lokalisierung in der x - und y -Koordinate unterschiedlicher Posebestimmungssysteme anhand einer quadratischen Trajektorie, die durch den Demonstrator dreimal abgeflogen wurde

Die Poseschätzung basierend auf dem UWB-System sowie die des EKF der FCU, welche mit den UWB-Daten gestützt wurde, stimmen besser mit der Trajektorie überein, auch wenn hier Abweichung von bis zu 25 Zentimetern vorhanden sind. Das stärkere Rauschen, insbesondere im Bereich von $x = 2,0$ m bis $x = 2,5$ m und $y = 3,5$ m bis $y = 5,0$ m, kann durch geringfügige Verdeckungen, sodass keine direkte Sichtlinie von Tag zu Anker gegeben ist, sowie eine nicht optimale Geometrie der Anker für diesen Raumbereich erklärt werden.

Der EKF bezieht in die Poseschätzung die Messdaten der IMU der FCU ein. Diese Daten sowie der Schätzalgorithmus des EKF glätten so die Poseschätzung; die Stützung durch die UWB-Messungen ist aber klar ersichtlich.

Die Flughöhe von einem Meter bei diesem Flug wurde von allen Ansätzen ausreichend genau geschätzt, wobei hier das stärkste Rauschen mit einer maximalen Abweichung von 40 Zentimetern beim UWB-System vorhanden war, welches sich entsprechend auch auf die Schätzung des EKF's ausgewirkt hat. Die T265-Kamera hatte eine maximale Abweichung von unter zehn Zentimetern in der Höhenschätzung.

In Abbildung 18 ist der bestimmte/geschätzte Gierwinkel der unterschiedlichen Systeme für denselben Messdurchlauf dargestellt. Dieser stimmt bei allen Ansätzen gut mit dem wahren Wert (Motion Tracking System) überein. Am stärksten ist das Rauschen des Gierwinkels des UWB-Systems. Dieses wirkt sich auch auf die Schätzung des EKF's aus. Insgesamt bieten bei dieser Betrachtung sowohl die T265-Kamera als auch der EKF der FCU eine ausreichende Genauigkeit für den autonomen Flug.

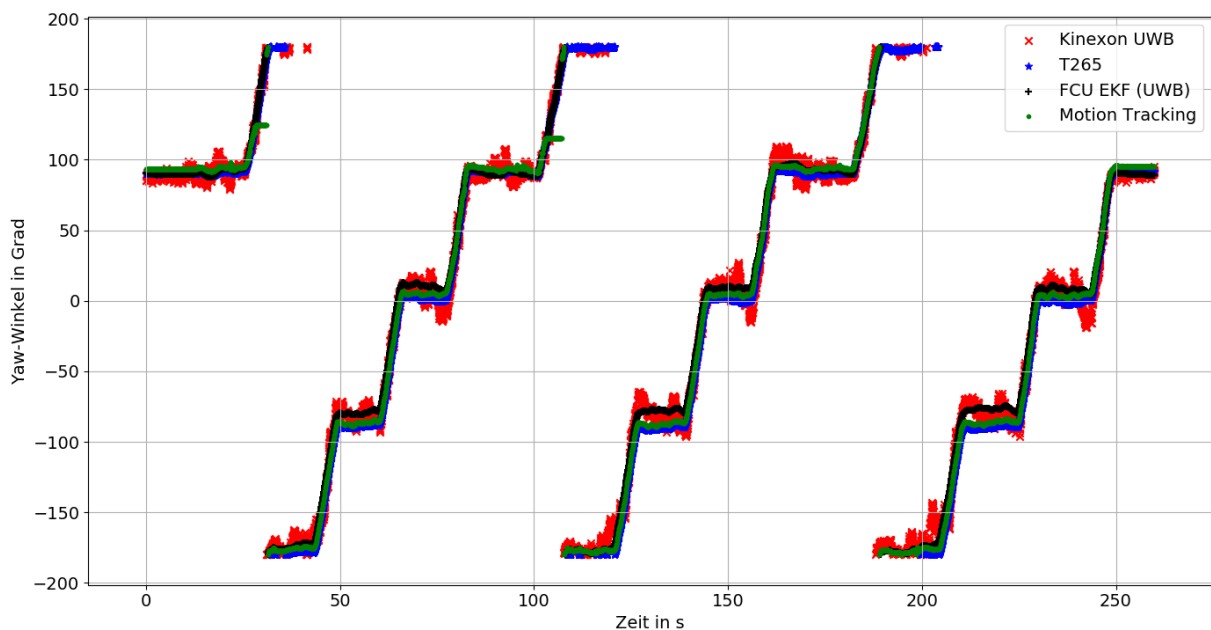


Abbildung 18: Vergleich des Gierwinkels (Yaw-Winkel) unterschiedlicher Posebestimmungssysteme anhand einer quadratischen Trajektorie, die durch den Demonstrator dreimal abgeflogen wurde

Insgesamt wäre die Fusion der Poseschätzungen von UWB-System und der T265-Kamera erstrebenswert, um eine möglichst robuste, präzise sowie stets absolute Poseschätzung für den Flugroboter zu generieren.

Abbildung 19 zeigt die geschätzte Flughöhe von VINS-Fusion und der T265-Kamera bei einem Flug, der ebenfalls eine quadratische Trajektorie in der x-y-Ebene hatte während die Flughöhe konstant 0,5 Meter betrug. VINS-Fusion wurde zur Poseschätzung basierend auf den Stereo-Bildern einer D435-Kamera sowie den IMU-Daten der FCU genutzt.

Die Flughöhe wurde von der T265-Kamera gut geschätzt. Jedoch wurde bei diesem Flug die Poseschätzung der T265-Kamera in den EKF der FCU gespeist wurde somit passte die Flugsteuerung die Flughöhe auch entsprechend der Schätzung an. Die Flughöhe weist jedoch keine Sprünge auf und wurde durchgängig relativ konstant vom Multicopter gehalten.

Jedoch ist klar erkennbar, dass VINS-Fusion die Flughöhe zunächst nur ein wenig überschätzte und anschließend bei Sekunde 30 ein Ansteigen der Flughöhe ca. 40 Zentimeter schätzt, welches jedoch im realen Flug nicht gegeben war. Dieser Anstieg der Höhenschätzung von VINS-Fusion zeigt, dass nicht nur die Umgebungsbedingungen Einfluss auf die VIO / den V-SLAM haben können sondern auch die Parametrisierung und Kalibrierung

eine entscheidende Rolle spielt. Diese ist bei der T265-Kamera bereits vorhanden, muss aber für VINS-Fusion selbst erstellt werden. Eine nicht exakte Kalibrierung und Parametrisierung führt beim VINS-Fusion Algorithmus zu einer ungenauen und instabilen Poseschätzung. In Experimenten in der Simulationsumgebung konnten bei einer exakten Kalibrierung und optimierten Parametrisierung auch mit diesem Algorithmus sehr gute Ergebnisse für die Poseschätzung erzielt werden. Bei Versuchen mit realen Drohnen trat des öfteren eine ungenaue, instabile Poseschätzung auf. Hier müsste in weiteren Entwicklungsarbeiten ein Vorgehen erarbeitet werden, wie die Parametrisierung und Kalibrierung zuverlässig für unterschiedliche Sensoren und Umgebungsbedingungen erfolgen kann.

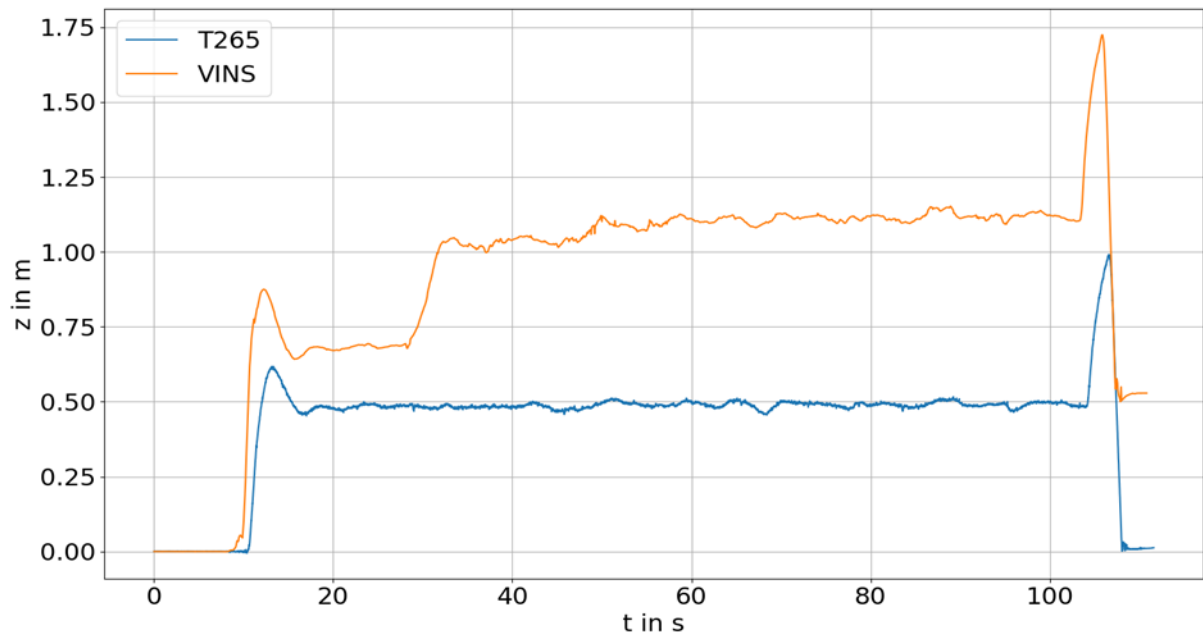


Abbildung 19: Vergleich der geschätzten Flughöhe durch die T265-Kamera und VINS-Fusion bei einem Flug mit konstanter Flughöhe von 0,5 Metern

Zusammengefasst ist die Poseschätzung des EKFs gestützt durch die UWB-Posebestimmung für den autonomen Flug ausreichend präzise. Die UWB-Posebestimmung kann sich jedoch durch Hindernisse in der Umgebung, die die Funksignale beeinflussen, verschlechtern. Für eine möglichst große Robustheit und Genauigkeit wäre also die Fusion von UWB-Posebestimmung und einer VIO / einem V-SLAM ein vielversprechender Ansatz. Hierfür müsste eine geeignete Sensordatenfusion erarbeitet werden, die insbesondere auch Schleifenschließungen bei einer VIO / eine V-SLAM geeignet berücksichtigt.

3.5 Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung

Aufbauen auf der Poseschätzung, kann die Navigation des Flugroboters erfolgen. Hierzu ist eine geeignete Karte und Erfassung der Umgebung sowie die Pfadplanung basierend auf der Karte und erfassten Hindernissen notwendig. Zudem kann es vorteilhaft sein, zusätzliche Informationen über die Umgebung in einer semantischen Karte zu speichern. Jedoch sollte trotzdem auch die Kollision des Flugroboters mit Objekten der Umgebung abgesichert werden, bzw. Maßnahmen getroffen werden, um mögliche Schäden dadurch zu vermindern.

Im Folgenden wird zunächst der Stand von Forschung und Entwicklung der Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung (Abschnitt 3.5.1) vorgestellt. Anschließend wird das erarbeitete Konzept zur Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung erläutert (Abschnitt 3.5.2). Die im Projekt erarbeitete Umsetzung der Umgebungserfassung und semantischen Kartierung sowie die darauf aufbauende Navigation werden in den Abschnitten 3.5.3 und 3.5.4 beschrieben. Die Evaluation und Bewertung der Evaluation und Bewertung der Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung erfolgt in Abschnitt 3.5.5.

3.5.1 Stand von Forschung und Entwicklung der Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung

Im Stand von Forschung und Entwicklung gibt es eine Vielzahl von Ansätze zur Kollisionsvermeidung bzw. zur Gefahrenminderung bei Kollisionen von Drohnen mit der Umgebung.

Gefahrenpotentiale, die beim Betrieb von Flugrobotern in Innenräumen auftreten können, werden umfassend in [13] und [14] erläutert und reichen vom Absturz des Flugroboters über Kollisionen und damit verbundene Schnittverletzungen und Beschädigungen durch die Propeller hinzu Lärmbelastung durch die sich drehenden Propeller.

Infrastrukturseitig kann einigen dieser Gefahrenpotentiale durch die Abtrennung des Arbeitsbereiches des Flugroboters durch Netze oder Einhausungen begegnet werden [13]. Infrastrukturmaßnahmen zur Absicherung des Flugbereiches sind jedoch mit hohen Kosten verbunden und schränken die Flexibilität in diesem Bereich ein.

Daher ist es vorteilhaft, Maßnahmen zur Kollisionsvermeidung bzw. zur Gefahrenminderung bei Kollisionen möglichst direkt auf dem Flugroboter umzusetzen.

Zur Gefahrenminderung existieren unterschiedliche Ansätze, wie Fallschirme [46], das automatische Stoppen der Propeller bei Kollisionen [47] sowie Schutzvorrichtungen, die die Rotoren umschließen [48]. Fallschirme haben im Innenbereich aufgrund der meist geringen Flughöhe aber meist keine ausreichende Auswirkung, um den Flugroboter bei einem Absturz abzubremsen. Beim Stoppen der Propeller wird zwar das Gefahrenpotential durch die Propeller gemindert, jedoch stürzt der Roboter so auch ab. Im Rahmen des Projektes wurden daher Schutzkäfige für die Propeller konzipiert, mittels 3D-Druck gefertigt und evaluiert.

Natürlich sollten aber im Optimalfall Kollisionen durch den Flugroboter aktiv vermieden werden. Hierzu muss der Roboter die Umgebung und Hindernisse in einer 3D-Karte erfassen und um diese herum navigieren und diesen ausweichen.

Sensoren hierzu sind, wie bereits in Abschnitt 3.4.1 vorgestellt, die Ultraschallentfernungsmessung, LIDAR und Kameras, insbesondere Stereo- oder RGB-D Kameras zur 3D-Erfassung der Umwelt.

Im Projekt wurde ein Konzept zur Kollisionsvermeidung und Umgebungserfassung basierend auf Ultraschallsensorik und RGB-D Kameras erarbeitet, 3D-LIDAR-Sensorik wurde aufgrund der hohen Kosten und des hohen Gewichts nicht betrachtet. Die Kartierung mittels 3D-LIDAR und Drohnen ist aber möglich [33].

Es gibt Ansätze zur Umgebungskartierung mittels Ultraschallsensorik [49]. Die damit erzeugten Karten sind jedoch ungenauer im Vergleich zu mit Stereo- bzw. RGB-D Kameras erstellte 3D-Karten. Ansätze zur Umgebungskartierung mittels Stereo- bzw. RGB-D Kameras werden unter anderem in [50] und [51] präsentiert. Die Ultraschallsensoren bieten jedoch den Vorteil, dass sie Hindernisse zuverlässig und ohne komplexe Datenverarbeitung detektieren können. Eine Erweiterung von klassischen Karten stellen semantische Karten dar [31]. Hierbei wird die Karte, welche die Umgebung abbildet, um zusätzliche Informationen angereicht, beispielsweise die Art der Objekte, die in der Karte erfasst sind. Ansätze zur semantischen Kartierung werden, teilweise auch mit Drohnen, werden in [52–56] vorgestellt.

Da keiner dieser Ansätze geeignet ist, um auch auf dem On-Board-Computer eines Flugroboters in dynamischen Umgebung in Innenbereichen ausgeführt zu werden, wurde im Rahmen des Projektes ein neuer Ansatz erarbeitet. Dieser Ansatz wird im Februar 2022 auf der „17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications“ (VISAPP 2022) unter dem Titel „Efficient Semantic Mapping in Dynamic Environments“ präsentiert und anschließend als Fachartikel in den Proceedings der Konferenz veröffentlicht (Stand 16.11.2021: der Beitrag wurde zur Publikation angenommen) [57]. In Abschnitt 3.5.3 wird ein Überblick über diesen Ansatz gegeben.

Im Folgenden wird das Konzept zur Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung basierend auf Propellerschutzkäfigen, einer RGB-D Kamera und Ultraschallsensorik für den Flugroboter vorgestellt.

3.5.2 Konzept zur Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung

Für die Absicherung und Kollisionsvermeidung ausgehend vom Flugroboter wird ein Ansatz basierend auf einer Kamera, Ultraschallsensorik und Propellerschutzkäfigen vorgeschlagen, wie in Abbildung 20 dargestellt.

Die Propeller werden durch Schutzkäfige eingehaust, um das Risiko von Schnittverletzungen bzw. Schäden bei Kollisionen zu minimieren. Die Schutzkäfige wurden im Rahmen des Projektes entworfen und mittels 3D-Druck gefertigt.

Eine in Vorwärtsrichtung des Flugroboters blickende Stereo- bzw. RGB-D Kamera (Blickfeld angedeutet durch die gelben gestrichelten Linien) ermöglicht einen umfassenden und detaillierten 3D-Überblick über die Umgebung. Basierend auf den Sensordaten der Kamera können Hindernisse detektiert und erkannt werden sowie eine 3D-Karte der Umgebung aufgebaut und aktualisiert werden. Zur Erstellung der 3D-Karte wurde RTAB-Map genutzt, da der Ansatz sehr gute Ergebnisse liefert, mit unterschiedlichen Kameras und weiterer Sensorik genutzt werden kann und eine Open-Source-Implementierung einschließlich einer Einbindung ins ROS verfügbar ist [50].

Parallel zu RTAB-Map kann mittels des in Abschnitt 3.5.3 beschriebenen Ansatzes eine semantische Karte erstellt werden, die nicht wie RTAB-Map die gesamte Umgebung abbildet, sondern nur relevante Objekte (z. B. Menschen oder KLTs). Somit kann mit RTAB-Map eine 3D-Karte der Umgebung erstellt werden (ohne semantische Informationen), die alle Hindernisse

in der Umgebung enthält, und mit dem neu erarbeiteten Ansatz eine semantische Karte, die spezifische prozess- oder sicherheitsrelevante Objekte enthält.

Da nur eine Kamera genutzt wird, die in Flugrichtung des Flugroboters ausgerichtet ist, können keine Hindernisse erfasst werden, die sich seitlich oder hinter diesem befinden. Hierzu werden Ultraschallsensoren genutzt, sodass eine Hindernisdetektion im gesamten Umfeld um den Multicopter im Bereich von einigen Metern möglich ist. Mit den Sensoren kann so erkannt werden, wenn sich der Roboter einem nicht durch die Kamera erfassten Objekt nähert oder sich ein anderes dynamisches Objekt dem Roboter seitlich oder von hinten nähert. Somit können entsprechende Ausweichaktionen durch den Flugroboter eingeleitet werden. Das Blickfeld der Kamera sollte auch durch einen Ultraschallsensor abgedeckt werden, um Hindernisse, die durch die Kamera nicht erfasst werden können, wie z. B. Glas, zuverlässig zu erfassen.

Die Kollisionsvermeidung erfolgt anschließend durch eine geeignete Pfadplanung, die auf den detektierten Hindernissen und er erstellten Karte basiert. Sollte es dennoch zu einer Kollision kommen, so wird die Gefahr von Verletzungen und Beschädigungen durch die drehenden Propeller durch Propellerschutzkäfige gemindert. Diese schützen auch die Propeller selbst vor Beschädigungen.

Bei der Anbringung der Sensoren müssen die Schutzkäfige sowie gegebenenfalls weitere Komponenten des Roboters berücksichtigt werden. Diese sollten nicht im Erfassungsfeld der Sensoren sein, um Fehlmessungen zu vermeiden.

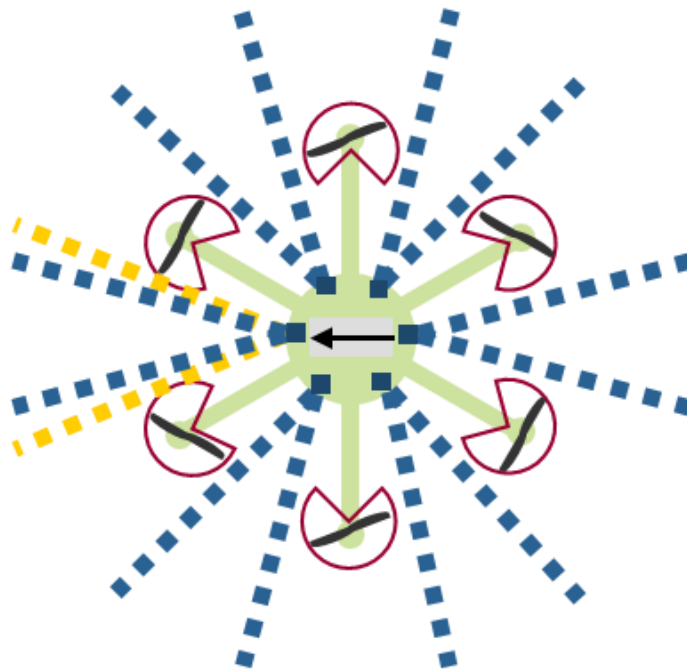


Abbildung 20: Konzept zur Absicherung des Flugroboters mit einer Stereo-/RGB-D Kamera (Erfassungsfeld in gelb skizziert) sowie Ultraschallsensorik (Erfassungsfelder in blau skizziert) und Propellerschutzkäfigen (rot)

3.5.3 Erstellung und Aktualisierung einer dreidimensionalen semantischen Karte

Eine dreidimensionale Karte der Umgebung ist grundlegend für die Navigation geeignet, enthält aber keine weiteren Informationen, um z. B. Kleinladungsträger erkennen und anfliegen zu können. Zudem sind auch keine sicherheitsrelevanten Informationen enthalten, wie die Art eines Hindernisses. Für einen effizienten und gleichzeitig sicheren Betrieb wäre es wünschenswert, dass der Flugroboter Hindernisse in der Umgebung nicht nur detektieren, sondern diese und weitere Objekte auch klassifizieren kann. Hierdurch kann einerseits effizienter mit Objekten interagiert werden, andererseits aber auch besser auf Hindernisse reagiert werden, indem beispielsweise ein größerer Sicherheitsabstand zu Personen eingehalten wird.

Die Verarbeitungskette zur Erstellung und Aktualisierung der semantischen Karte, welche im Rahmen des Projektes erarbeitet wurde, ist in Abbildung 21 dargestellt.

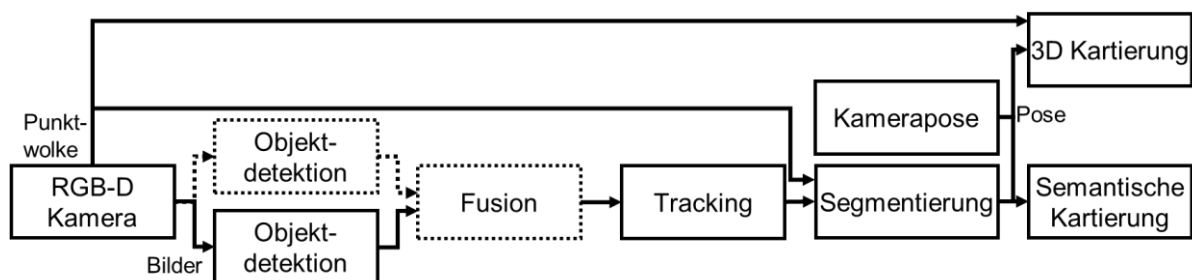


Abbildung 21: Überblick über den erarbeiteten Prozess zur Erstellung der semantischen Karte

Die Erstellung und Aktualisierung der semantischen Karte basiert auf den Sensordaten einer RGB-D Kamera. Für die Entwicklung und Evaluation wurde hierfür die Intel D435 Depth Camera genutzt. Die Farbbilder werden von einem oder mehreren Objektdetektionsalgorithmen ausgewertet. Dies können beliebige Algorithmen sein, die für erkannte Objekte eine Bounding Box, also ein Rechteck, welches das erkannte Objekt im Bild möglichst exakt umschließt, generieren. Zusätzlich sollte ein Konfidenzwert für die Erkennung durch den Algorithmus angegeben werden. Die Objektklasse kann hier die konkrete Art des Objektes sein, also z. B. Mensch, aber auch ein spezifisches Attribut, beispielsweise ob sich ein Objekt aktuell bewegt, also dynamisch ist [58].

Die Verarbeitungskette ist so gestaltet, dass, je nach Anforderung, einer oder mehrere Objektdetektionsalgorithmen verwendet werden können. Für den Fall, dass mehrere dieser Algorithmen verwendet werden sollen, werden die Objektdetektionen der unterschiedlichen Detektoren anschließend fusioniert, mit dem Ziel, dass je realem Objekt auch nur eine Detektion vorliegt. Beispielsweise könnte ein laufender Mensch von einem Detektor im Bild als Mensch erkannt werden, von einem weiteren Detektor als dynamisches Objekt. Durch die Fusion der Detektionen der unterschiedlichen Detektoren soll hieraus entsprechend die Detektion „dynamischer Mensch“ generiert werden. Hierzu werden die Detektionen der unterschiedlichen Detektoren basierend auf der Überlappung ihrer Bounding Boxes fusioniert und die semantischen Informationen konsolidiert. Die grundlegenden Ansätze für diese Fusion sind in [58] und [59] beschrieben.

Anschließend erfolgt ein Tracking der detektierten Objekte im 2D-Bildraum, also die Zuordnung, ob ein neu detektiertes Objekt auch in vorherigen Bildern zu sehen war, es sich somit um dasselbe reale Objekt handelt. Die ist z. B. mittels der Bounding Boxes der Detektion möglich, wie in [60] beschrieben.

Um eine dreidimensionale semantische Karte erstellen zu können, müssen die Detektionen vom 2D-Bildraum in den 3D-Raum überführt werden. Dies erfolgt auf Basis eines selbst erarbeiteten Segmentierungsalgorithmus. Hierbei werden zunächst aus dem zum Farbbild, in welchem die Detektionen gemacht wurden, korrespondierende Punktwolke für jede Detektion extrahiert, welche die jeweilige Bounding Box umschließt. Anschließend werden die Punkte in den einzelnen Teilpunktwolken mittels Euclidean Clustering geclustert [61]. Das Cluster, welches sowohl möglichst viele Punkte umfasst als auch möglichst zentriert in der Teilpunktwolke liegt, wird als die das reale Objekt repräsentierende Punktwolke bestimmt. Alle anderen Punkte werden dem Hintergrund zugerechnet. Zur effizienteren Berechnung wird nicht jede Punktwolke einer Objektdetektion im Folgenden verarbeitet, sondern eine 3D-Bounding Box, welche diese Punktwolke umschließt.

Zusammen mit der aktuellen Pose der Kamera bzw. des Flugroboters, kann mittels der 3D-Bounding Box, der Klasse und der Trackinginformationen die semantische Karte erstellt bzw. aktualisiert werden. Dieser Prozess ist in Abbildung 22 skizziert und in folgende Schritte strukturiert:

1. Transformation der detektierten Objekte ins Kartenkoordinatensystem: Zunächst werden die 3D-Positionen bzw. 3D-Bounding Boxes der Detektionen vom Kamerakoordinatensystem, in dem sie aus Sicht der Kamera (Bild/Punktwolke) aufgenommen wurden in ein übergeordnetes Koordinatensystem, z. B. ein Kartenkoordinatensystem transformiert. Hierfür wird die geschätzte Pose des Roboters bzw. die der Kamera im übergeordneten Koordinatensystem genutzt. Somit sind am Ende dieses Schrittes alle neuen Detektionen transformiert ins übergeordnete (Karten)-Koordinatensystem verfügbar.
2. Bestimmung „erwarteter Objekte“: Im zweiten Schritt wird berechnet, welche Objekte, die bereits in der semantischen Karte erfasst sind, sich aktuell im Blickfeld der Kamera befinden und somit erkannt werden sollten. Hierfür werden die in der Karte vor der Kamera befindlichen Objekte mittels ihrer 3D-Positionen, der intrinsischen Kameraparameter und des entsprechenden Kameramodells in die Bildebene der Kamera projiziert. So kann bestimmt werden, welche bereits in der Karte erfassten Objekte aktuell im Kamerabild wieder sichtbar sein sollten und somit auch in den neuen Detektionen enthalten sein sollten.
3. Datenassoziation: In diesem Schritt wird überprüft, ob neue Messungen bereits in der semantischen Karte erfassten Objekten zugeordnet werden können. Dies erfolgt zunächst über die Tracking-ID basierend auf dem Tracking im 2D-Bild. Stimmt diese mit der Tracking-ID eines Objektes in der Karte überein, so werden diese assoziiert, d. h. es wird angenommen, dass die neue Detektion auf demselben realen Objekt basiert, wie das bereits in der Karte erfasste. Für Detektionen, die nicht bereits erfassten Objekten zugeordnet werden konnten, wird überprüft, ob sich deren 3D-Bounding Box mit einem Objekt in der Karte ausreichend überschneidet (mittels einer Intersection over Union-Berechnung) und ob deren Klasse identisch ist. Ist dies der Fall so werden die entsprechenden Detektionen und bereits kartierten Objekte assoziiert.
4. Kartenaktualisierung: In diesem Schritt werden zunächst die Objekte, zu denen eine Detektion assoziiert werden konnten, aktualisiert, in dem die neue Information über Position und die Größe der 3D-Bounding Box in die bereits vorhandenen Objektinformationen gewichtet miteinbezogen wird. Neue Detektionen, welche nicht mit einem

Objekt in der Karte assoziiert wurden, werden nun als neue Objekte in die Karte übernommen. Zusätzlich wird überprüft, ob Objekte aus der Karte gelöscht werden sollten. Hierfür wird mithilfe eines Persistence-Filters geschätzt, wie wahrscheinlich es ist, ob das Objekt noch präsent ist [62]. Wird ein Objekt erneuert detektiert, so nimmt diese Wahrscheinlichkeit zu bzw. bleibt konstant. Wird ein erwartetes Objekt aber nicht detektiert, so verringert sich diese Wahrscheinlichkeit. Unterschreitet sie einen Schwellwert, wird das Objekt aus der Karte gelöscht.

Die Einbeziehung von weiteren Objektattributen, wie der Dynamik eines Objektes in diesen Prozess ist möglich, wie in [57] dargestellt wird.

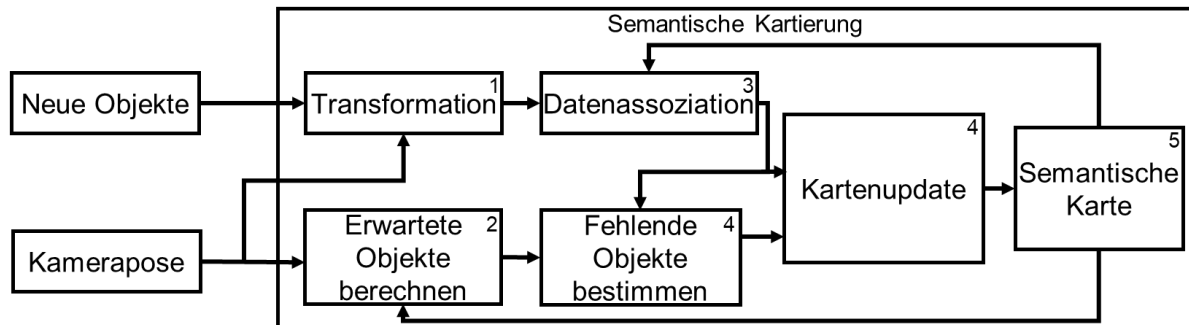


Abbildung 22: Prozessablauf zur Erstellung und Aktualisierung der semantischen Karte

Mit dem Ansatz wird eine dreidimensionale semantische Karte erzeugt, die spezifische Objekte enthält. Die gesamte Umgebung und Hindernisse darin werden aber nicht abgebildet. Hierzu sollte deswegen parallel ein entsprechender 3D-Kartierungsansatz wie RTAB-Map genutzt werden.

3.5.4 Autonome Navigation des Flugroboters

Die einfachste mögliche Art der Navigation des Flugroboters ist der direkte Flugweg. Dem Flugroboter bzw. der Flugsteuerung wird eine Zielpose übergeben, der angefliegen werden soll. Die Flugsteuerung regelt die Motoren daraufhin entsprechend so, dass der Flugroboter auf direktem Weg – unter Berücksichtigung der voreingestellten Parameter wie der maximalen Geschwindigkeit – dorthin fliegt. Soweit keine Hindernisse im Flugbereich des Flugroboters vorhanden sind oder dem Flugroboter entsprechende Zielpunkte unter Berücksichtigung aller Hindernisse von einem externen System gesandt werden, stellt dies eine einfache und effiziente Art der Navigation dar.

Im realen Betrieb sind aber die Informationen über Hindernisse meist nicht genau und die Hindernisse im Arbeitsraum verändern sich. Zudem können neue Hindernisse, wie ein Mensch, den Flugbereich betreten. Durch das Kollisionsvermeidungskonzept bestehend aus Ultraschallsensorik und einer Kamera sollten Hindernisse frühzeitig detektiert werden und gegebenenfalls ein Not-Stopp eingeleitet werden können. Dennoch ist es effizienter und sicherer, Hindernisse bereits frühzeitig zu detektieren und zu kartieren sowie diese anschließend zu umfliegen.

Für die zweidimensionale Navigation kann dies mithilfe des ROS Navigation-Stacks umgesetzt werden [63]. Im Rahmen des Projekts wurde hiermit eine 2D-Pfadplanung und darauf aufbauende Navigation des Flugroboters realisiert.

Die Pfadplanung basiert hierbei zunächst auf einer globalen Pfadplanung, die mittels meist statischen globalen Kostenkarte einen ersten Pfad zum Ziel plant. Die statische Karte der

Umgebung wird im Vorfeld, z. B. mithilfe von RTAB-Map erstellt. Da in der statischen Karte keine dynamischen und neu erschienenen Hindernisse erfasst sind, wird während des Fluges mittels einer lokalen Kostenkarte, des globalen Pfades und eines lokalen Pfadplaners ein lokaler Pfad geplant, der auch dynamische und neue Hindernisse berücksichtigt, sich aber am globalen Pfad orientiert. Dieser Prozess ist in Abbildung 23 skizziert. Die Kostenkarten können hierbei aus mehreren einzelnen Schichten aufgebaut werden. In Abbildung 23 sind dies für die lokale Kostenkarte beispielsweise die Hindernisschicht und die Inflationsschicht. Die Hindernisschicht beinhaltet Hindernisse, die mittels aktueller Sensormessung erkannt wurden. Die Inflationsschicht wiederum vergrößert die Hindernisse, sodass auch die Dimensionen des Roboters bei der Pfadplanung um die Hindernisse berücksichtigt werden.

Eine mehrschichtige Kostenkarte bietet den Vorteil, dass unterschiedliche Informationen mit jeweils eigenen Regeln zur Kostenbestimmung genutzt werden können. Die einzelnen untergeordneten Schichten können somit unabhängig voneinander gestaltet werden. Neue Kostenkarten bzw. Schichten können flexibel je nach Bedarf eingefügt werden und nicht mehr benötigte entfernt werden. [64]

Aufbauend auf der semantischen Karte ist entsprechend auch eine semantische Navigation möglich, also die Integration von semantischen Informationen in die Pfadplanung und Navigation. Hierfür wurde die zuvor beschriebene Lösung zur Navigation basierend auf dem ROS um die Semantik erweitert. Hierbei werden zusätzliche Kostenschichten mit den gewünschten Regeln für die Pfadplanungskosten der darin enthaltenen Objekte definiert. In der übergeordneten Gesamtkostenkarte werden wiederum alle Kostenkarten aggregiert. Für Personen ist dies beispielsweise bereits durch den „Social Layer“ im ROS Navigation Stack implementiert [65].

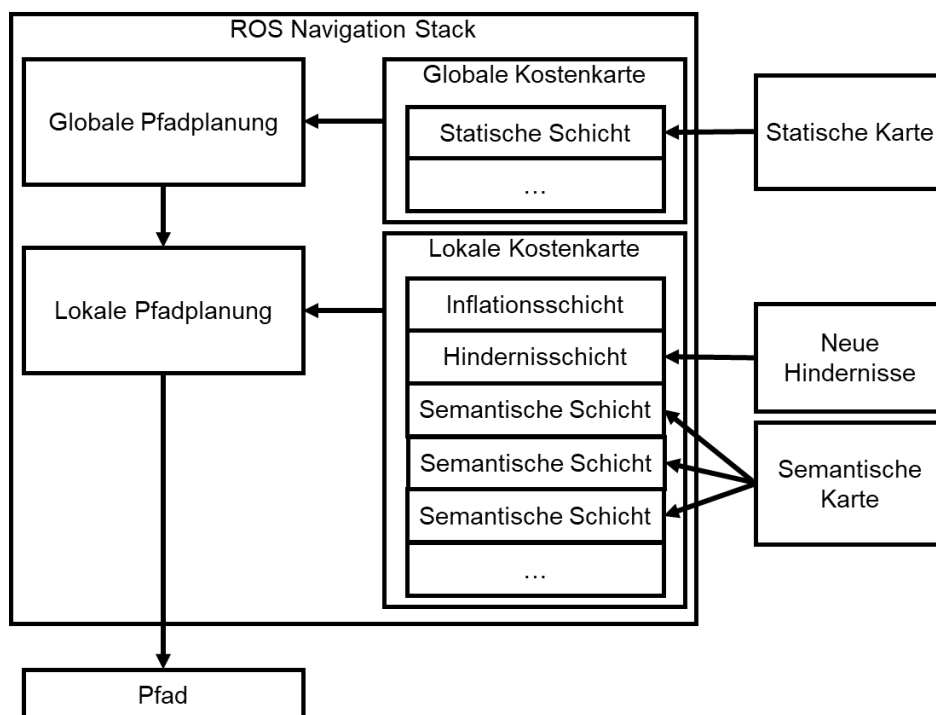


Abbildung 23: Überblick über die Pfadplanung mittels mehrschichtiger Kostenkarten basierend auf dem ROS Navigation Stack

Der „Social Layer“ bildet Personen als Hindernisse mit spezifischen Kosten in einer Kostenkarte ab. Hierfür müssen der Schicht die Positionen aller aktuell erfassten, z. B. mittels des

Ansatzes zur semantischen Kartierung, Personen sowie optional deren Geschwindigkeiten übergeben werden. Zwei Möglichkeiten für die Kostenfestlegung von Personen sind durch den „Proxemic Layer“ und den „Passing Layer“ implementiert. Der „Proxemic Layer“ übernimmt statische Personen in eine Kostenkarte und trägt erhöhte Kosten in einem gewissen Bereich um die Position der Person basierend auf einer konfigurierbaren Normalverteilung ein. Im Falle einer laufenden, also dynamischen Person werden die Kosten in Bewegungsrichtung erhöht, um zu vermeiden, dass der Roboter ebenfalls in diese Richtung einen Pfad plant. Der „Passing Layer“ erhöht die Kosten, zusätzlich zu den grundlegenden Kosten durch die Person, auf der rechten Seite der Person, sodass der Roboter dazu bewegt wird, Personen normalerweise auf der linken Seite zu umfahren. [65]

Im Rahmen des Projektes wurden zwei weitere Möglichkeiten für semantische Kostenkarten umgesetzt. Einerseits eine höhenabhängige Ausblendung von Objekten, andererseits eine Ausblendung von definierten Objekten in den nahen Umgebungen. Durch die höhenabhängige Ausblendung von bestimmten Objekten ist es möglich, diese Objekte (z. B. KLTs) auch mit einer 2D-dimensionalen Pfadplanung zu überfliegen, während um andere Hindernisse immer herumnavigiert wird. Die Ausblendung nahe gelegener Objekte ermöglicht die Annäherung an diese, z. B. an ein Regal um eine Kennzeichnung bei einer Inventur genau zu erfassen. Diese Umsetzung von zwei weiteren Arten von Kostenkarten stellt nur beispielhaft die Möglichkeiten zur Realisierung von neuen semantischen Navigationsregeln mittels der semantischen Schichten dar. Grundlage ist jedoch immer, dass die Objekte in der Umgebung entsprechend semantisch erfasst werden können.

Die 2D-Pfadplanung und Navigation ist grundlegend für Flugroboter geeignet, dennoch muss die Flughöhe immer definiert festgelegt werden. 3D-Pfadplaner sind bereits im Stand der Forschung und Entwicklung vorhanden und können in künftigen Forschungs- und Entwicklungsarbeiten basierend auf den bereits umgesetzten Ansätzen zur 3D-Kartierung integriert werden [66, 67].

Hierbei sollte aber, sofern sich auch Menschen im Arbeitsraum aufhalten könnten, berücksichtigt werden, dass das (Flug-)Verhalten für den Menschen auch möglichst vorhersehbar und planbar sein sollte, um Fehleinschätzung des Roboterverhaltens und daraus resultierende Gefahrensituationen zu vermeiden. Ein geeigneter Ansatz hierfür ist die 2,5D-Pfadplanung und -Navigation. Hier steht nicht der gesamte Raum zur Verfügung, sondern der Raum wird in unterschiedliche Flugebenen unterteilt. Die erleichtert die Pfadplanung wesentlich und auch Vorhersehbarkeit der Bewegungen des Flugroboters [68, 69].

3.5.5 Evaluation und Bewertung der Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung

Zunächst wird die Kollisionsabsicherung durch die Propellerschutzkäfige betrachtet, welche mit dem Quadrocopter basierend auf dem 3D-gedruckten Rahmen in Experimenten evaluiert wurden. Drei Varianten hiervon sind in Abbildung 24 dargestellt, die von links nach rechts jeweils eine optimierte Version des Propellerschutzkäfigs darstellen. Alle Varianten wurden mittels 3D-Druck aus PLA gefertigt. Die erste Version (links) hatte aufgrund der vollständigen Umschließung des Propellers ein relativ hohes Gewicht. Daher wurde die zweite Variante (Mitte) konzipiert, die ein geringeres Gewicht hat, den Propeller aber nicht mehr auf der dem Multicopter zugewandten Seite umschließt. Dieser Anpassung liegt die Annahme zu Grunde,

dass eine Kollision von dieser Seite aus sehr unwahrscheinlich ist. Die Änderungen am Käfig haben aber im Flug zu starken Vibrationen hervorgerufen durch den die Schutzkäfige geführt, welche sich auch negativ auf das Flugverhalten ausgewirkt haben. Diese Vibrationen konnten durch die Einbringung von Streben eliminiert werden (siehe Abbildung 24 rechts).



Abbildung 24: CAD-Modelle der konzipierten und untersuchten Propellerschutzkäfige

Mit den Käfigen wurden Kollisionstests mit einem Stapel Pappkartons durchgeführt. Wie in den Bildern in Abbildung 25 zu sehen ist, konnte der Multicopter die Kollisionen ohne Absturz und Beschädigungen an sich selbst oder den Pappkartons überstehen. Die Propellerschutzkäfige sind daher eine sinnvolle Maßnahme zur Kollisionsabsicherung. In weiteren Arbeiten sollte aber das Material und die Struktur der Käfige noch detailliert analysiert werden, da die Käfige bei Kollisionen bei hohen Fluggeschwindigkeit wahrscheinlich keinen ausreichenden Schutz bieten. Zudem haben die Käfige ein relativ hohes Gewicht von ca. 430 Gramm (vier Käfige). Das Gewicht der Schutzkäfige sollte soweit möglich weiter reduziert werden oder entsprechend in die Dimensionierung des Multicopters einbezogen werden.

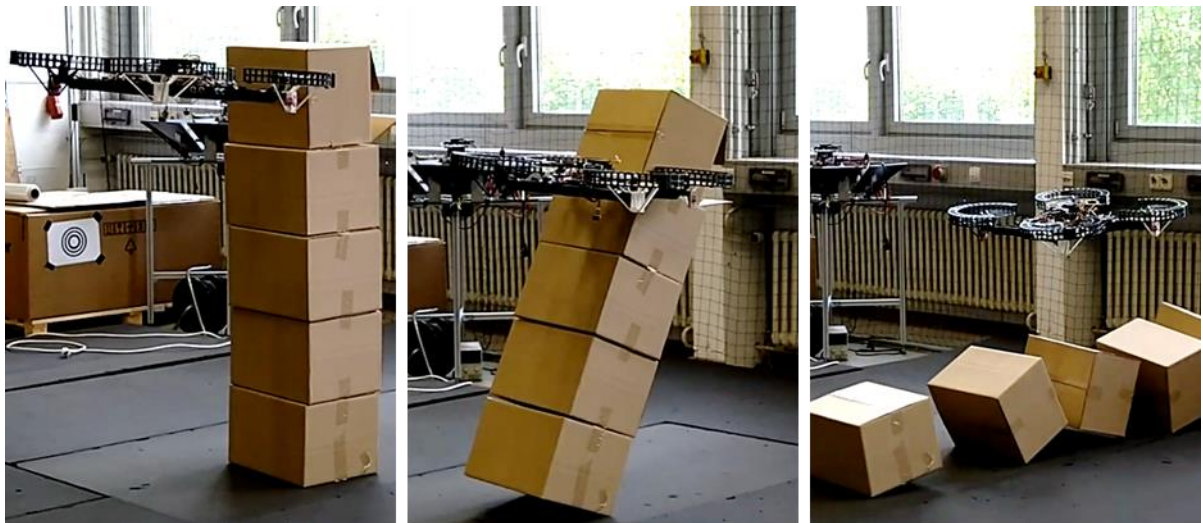


Abbildung 25: Erfolgreicher Kollisionstest mit den Propellerschutzkäfigen

In Abbildung 26 links ist ein Flugroboter während des Fluges im Versuchsfeld des Lehrstuhls zu sehen, um den Roboter herum befinden sich drei Stapel mit Pappkartons, sowie weitere Hindernisse im Hintergrund (Regal, Kisten, Wand, etc.). Im Bild rechts ist die vom Flugroboter mittels einer D435-Kamera und RTAB-Map erstellte Karte sowie überlagert die lokale Kostenkarte zur Navigation zu sehen. Zudem sind der globale Pfad (blau) und der lokale Pfad (rote Pfeile, Planung mittels Time-Elastic-Band Pfadplaner [70]) zu einer Zielpose dargestellt. Die aktuelle Pose des Flugroboters wird durch die Koordinatensysteme in der Mitte der Karte angezeigt. Der Flugroboter kann die Hindernisse in der Umgebung, insbesondere die Pappkartons sehr gut erfassen und korrekt in die Karte eintragen, was in der Karte den drei großen

roten Voxelansammlungen entspricht. Um die so als Hindernisse eingetragenen Kartonstapel wird auch durch die lokale Kostenkarte ein Sicherheitsbereich (in der Karte cyan) entsprechend der Dimensionen des Flugroboters eingetragen, sodass dieser den Stapeln bei der autonomen Navigation nicht zu nahekommt. Zusätzlich sind freie Bereiche, also Bereiche in denen navigiert werden kann in grau und durch Hindernisse belegte Bereiche in schwarz dargestellt.

Bei der Hinderniskartierung sind aber auch einige Fehler bzw. eine ungenaue Kartierung der Hindernisse zu sehen, beispielweise im unteren Kartenbereich sind einige einzelne rote Voxel eingetragen. Diese Fehler und Ungenauigkeiten entstehen durch das Rauschen des Tiefenbildes und so der Punktwolke der Kamera. Dies ist besonders bei größeren Entfernungen zur Kamera (ab ca. 5 m) der Fall. Die Karte kann korrigiert bzw. optimiert werden, sobald der Flugroboter entsprechende Bereiche in geringer Distanz dazu erfasst.

Die lokale Pfadplanung, dargestellt durch rote Pfeile, leitet den Flugroboter sicher um die Hindernisse herum zur Zielpose. Hierbei wird der Flugroboter stets so ausgerichtet, dass er in Vorwärtsrichtung, also in die Richtung in die Kamera blickt, fliegt.

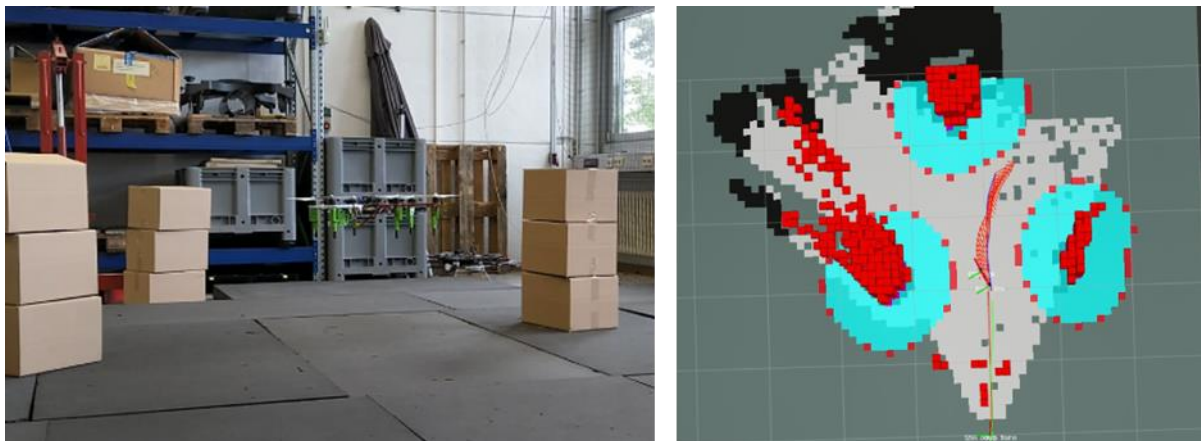


Abbildung 26: Links: Flugroboter, in dessen Umgebung sich drei Pappkartonstapel als Hindernisse befinden, während des Fluges; Rechts: Karte, Hindernissdetektion und Pfadplanung zum Erreichen eines Zielpunktes zwischen de Pappkartons des Flugroboters

In Abbildung 27 rechts ist die Sensorik zur Umgebungserfassung angebracht an einen Multi-copter zu sehen. Oben ist ein Ultraschallsensor (MB1043 HRLV-MaxSonar-EZ4) befestigt, darunter eine D435 RGB-D Kamera, unter dieser ist eine T265-Kamera angebracht (diese ist jedoch nicht relevant für die Umgebungserfassung und -kartierung). Im Bild rechts ist die Hinderniserkennung des Flugroboters vor einer Glasscheibe visualisiert, d. h. Kamera und Ultraschallsensor waren in Richtung der Glasscheibe ausgerichtet. Die Hinderniserkennung durch den Ultraschallsensor ist durch den grünen Kreis im Bild gekennzeichnet, die der Kamera durch den roten Kreis. Die Position der Drohne entspricht dem mittleren Koordinatensystem im Bild. Hier wird deutlich, dass die Kamera durch die Glasscheibe sieht, also nur die Hindernisse im Hintergrund erfasst. Der Ultraschallsensor dagegen detektiert die Glasscheibe als Hindernis sodass der Flugroboter entsprechend davor stoppen kann.

Im Projekt wurden auch weitere Untersuchung bezüglich der Hindernisdetektion mittels Ultraschallsensoren (HC-SR04) durchgeführt. Die Detektion von Objekten mit einer kleinen Reflektionsfläche für das Ultraschallsignal ist dabei nicht immer zuverlässig und wird mit steigender Distanz schlechter. Beispielsweise wurden Pappkartons, wie sie in Abbildung 26 zu sehen sind, schon ab einer Distanz 3 Metern immer zuverlässig detektiert, ein Aluminiumprofil

mit einer Kantenlänge von 45 mm erst in einer Distanz von 2,5 m. Falls sich also sehr dünne Objekte, wie Rohre, Profile oder Kabel, im Flugbereich des Multicopters befinden, besteht die Gefahr, dass diese nicht vom Ultraschallsensor und auch nicht von der Kamera detektiert werden. Hier müssten weitere Maßnahmen zur Kollisionsvermeidung getroffen werden, wie z. B. die Festlegung von Bereichen in der Karte, in denen der Multicopter nicht fliegen darf, wie z. B. in [68] vorgestellt wird.

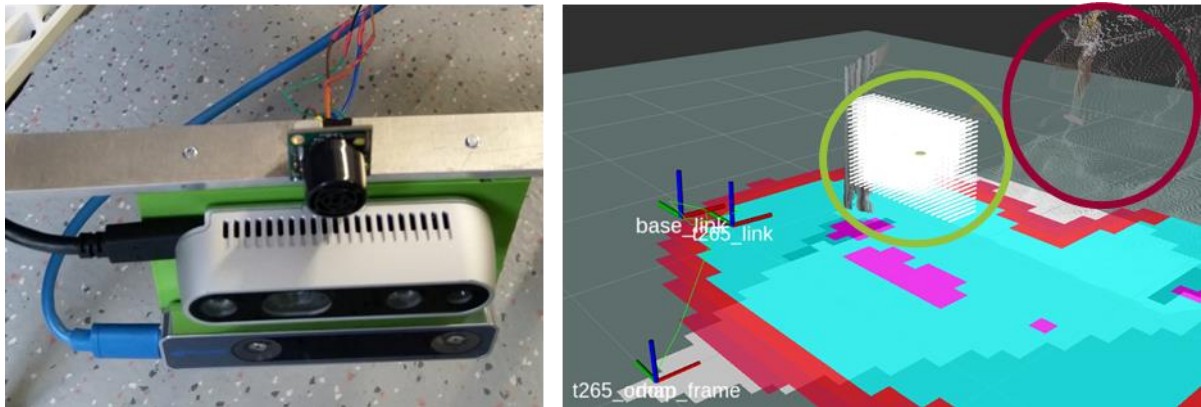


Abbildung 27: Links: Ultraschallsensor, D435 Depth Camera und T265 Tracking Camera montiert an einem Flugroboter; Rechts: Hindernisdetektion einer Glasscheibe durch den Ultraschallsensor (grüner Kreis) sowie durch die D435-Kamera (roter Kreis)

Eine semantische Karte, die mit dem beschriebenen Ansatz und mit einem NVIDIA Jetson Xavier NX als OBC des Flugroboters erstellt wurde ist in Abbildung 28 links zu sehen. Hierbei wurde der Ansatz zur semantischen Kartierung mit dem Objektdetektionsalgorithmus Tiny-YOLOv4 [71] mit Gewichten basierend auf dem COCO-Datensatz, welcher auf der GPU des Jetson ausgeführt wurde, genutzt. Im Bild ist einerseits eine vor dem Flug aufgezeichnete statische Karte der Umgebung (graue (freie) und schwarze (belegte) Bereiche) sowie die aktuelle Hindernisdetektion und semantische Kartierung zu sehen. Die detektierten Hindernisse sind wiederum durch rote Voxel dargestellt. Die semantische Karte wird durch die 3D-Bounding-Boxes sowie die Klasse des kartierten Objekts darüber dargestellt. Unten im Bild ist noch das aktuelle Farbbild der D435-Kamera zu sehen. Hier sind die Objekte, welche vom Objektdetektionsalgorithmus erkannt wurden und so in der semantischen Karte erfasst sind, mit Pfeilen gekennzeichnet. Dies sind der Mensch („person“), ein Stuhl („chair“), sowie ein Koffer („suitcase“) im Hintergrund. Rechts ist zudem ein Hindernis zu sehen, das nicht semantisch erfasst wurde, aber durch die allgemeine Hindernisdetektion (RTAB-Map) in die Karte eingetragen wurde.

Der Mensch und der Stuhl, die nicht in der statischen Karte erfasst sind, wurden sowohl von der Hindernisdetektion mittels RTAB-Map also auch von der semantischen Kartierung korrekt erfasst und kartiert. Durch die Nutzung des Jetsons (6 CPU Kerne bei 15 Watt Leistung) und dessen GPU erreicht der Gesamtansatz (Hinderniserkennung und semantische Kartierung) eine Aktualisierungsrate von 14 – 15 Hz (entsprechend der Bildrate der Kamera, d.h. es werden alle Messungen der Kamera verarbeitet), sodass auch dynamische Hindernisse, wie der laufende Mensch gut erfasst und in der Karte abgebildet werden. Mit diesem Softwaresetup auf dem OBC – RTAB-Map zur Erfassung von Hindernissen, semantische Kartierung sowie

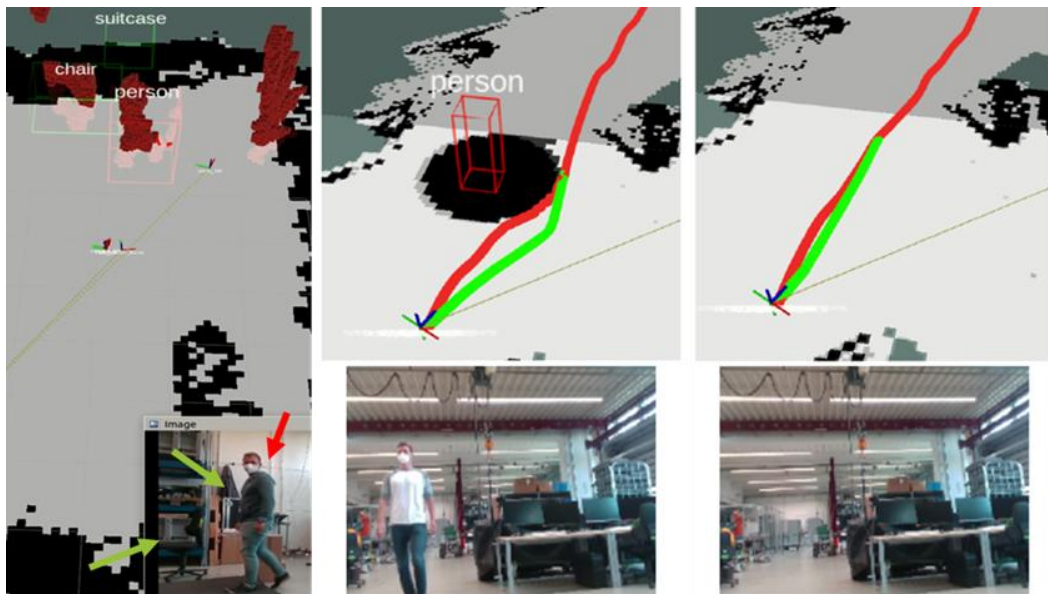


Abbildung 28: Hinderniserkennung und semantische Kartierung während des Fluges; Mitte und Rechts: Pfadplanung basierend auf einer Kostenkarte mit semantische Schichten

Poseschätzung basierend auf dem UWB-System und Aktionsplanung – ist die CPU bereits zu 40 bis 70 Prozent ausgelastet.

Bei der parallelen Nutzung von mehreren Objektdetektoren, z. B. zur zusätzlichen Erkennung von dynamischen Objekten, wie beispielsweise in [72] beschrieben, können nicht mehr alle Eingangsdaten aufgrund nicht ausreichender Rechenkapazität verarbeitet werden, zusätzliche Software, wie die Aktionsplanung ist somit nicht mehr zuverlässig für den autonomen Flug ausführbar.

In den beiden Bildern links daneben sind Ergebnisse der semantischen Navigation aufbauend auf der semantischen Kartierung und Hinderniserkennung zu sehen (mit Tiny-YOLOv4 als Objektdetektor). Die schwarzen und grauen Bereiche zeigen wieder belegte und freie Bereich in der statischen Karte. Wird eine Person erkannt und kartiert, wie im mittleren Bild, so wird durch den hier genutzten „Social Layer“ (Proxemic Layer mit deaktivierter Verzerrung) die Person in die Kostenkarte eingetragen. Im Beispiel ist diese Schicht sowohl in die globale Kostenkarte als auch in die lokale Kostenkarte eingebunden. Die globale (im Bild rot) und die lokale Pfadplanung (grün) planen einen Pfad um die Person herum. Bewegt sich die Person aus dem Bereich heraus, so wird sie aufgrund der nun nicht mehr vorhandenen Detektionen aus der semantischen Karte gelöscht und somit auch aus den Kostenkarten. Die Pfadplanung plant folglich auch wieder einen direkten Pfad. Dies zeigt, dass der Ansatz der semantischen Kartierung und der Übertragung der semantischen Informationen in die mehrschichtigen Kostenkarten gut funktioniert und einen Mehrwert bietet. So können für spezifische Objekt spezielle Navigationsregeln aufgestellt und Verhaltensweisen des Roboters bestimmt werden. Zu Menschen kann beispielsweise ein größerer Sicherheitsabstand eingehalten werden und deren Bewegung berücksichtigt werden, während die Kosten für den Anflug an ein Regal, z. B. zur Inventur, minimiert werden und der Roboter sich diesem so gut nähern kann.

Insgesamt funktioniert die Kollisionsabsicherung, Umgebungskartierung und Kollisionsvermeidung sehr gut, jedoch sind weitere Entwicklungsarbeiten zur Optimierung der Software oder ein leistungsfähigerer Rechner auf dem Flugroboter erforderlich. Der Multicopter müsste

dann hinsichtlich des größeren Gewichts und des höheren Leistungsbedarfs des OBCs skaliert werden.

Eine 3D- oder 2,5D-Pfadplanung sollte in fortführenden Arbeiten ebenfalls integriert werden. Hier müssten noch Entwicklungsarbeiten zur Verknüpfung mit der semantischen Karte erfolgen. Zudem kann die Definition von bestimmten Flugzonen, wie in [68] beschrieben, helfen, kritische Bereiche abzusichern.

3.6 Vollautomatisierte Erkennung und Handhabung von Kleinladungsträgern mit einem Flugroboter

Im Folgenden wird zunächst das grundlegende Konzept zur vollautomatisierten Handhabung von KLTs mit dem Flugroboter vorgestellt. Anschließend wird die Erkennung des Kleinladungsträgers und Posebestimmung relativ zu diesem sowie das System zur Aufnahme und zum Ablegen des Kleinladungsträgers detailliert beschrieben. Abschließend wird der erarbeitete Ansatz zu Handhabung des KLTs evaluiert und bewertet.

3.6.1 Stand von Forschung und Entwicklung der Objekthandhabung mittels Flugrobotern

Im Stand von Forschung und Entwicklung existieren unterschiedliche Ansätze zur Handhabung und insbesondere zum Aufnehmen von Objekten mit einem Flugroboter. In [73] wird ein Ansatz zum Greifen von Objekten mithilfe von Greiffingern, die an der Unterseite eines UAVs angebracht sind vorgestellt. Ein ähnliches Prinzip wird in [74] und [75] präsentiert. Die Aufnahme und Fixierung von geeigneten Objekten mittels eines Magneten wird in [76] erläutert. Das in [77] vorgestellte Prinzip zur Aufnahme von KLTs mit einem Multicopter zur Objekthandhabung wurde im Rahmen des Projektes weiterentwickelt und erweitert.

Bei allen Arbeiten gibt es Beschränkungen hinsichtlich der Art und Größe der Objekte, die aufgenommen werden können. Ein weiterer Faktor ist das Gewicht der Aufnahmevorrichtung, das im Vergleich zur Vorrichtung [77] im Rahmen des Projekts reduziert werden konnte.

3.6.2 Konzept zur Objekthandhabung

Das erarbeitete und realisierte Konzept zur Aufnahme der KLTs ist in Abbildung 29 dargestellt. Der Flugroboter erhält im ersten Schritt den Auftrag, einen KLT zu transportieren. Hierbei ist die Information, wo sich der Kleinladungsträger ungefähr befindet (wenige Meter Genauigkeit), erforderlich. Dies kann z. B. mithilfe von weiteren UWB-Tags realisiert werden, die regelmäßig die Position der zu transportierenden KLTs melden. Zusätzlich wäre es möglich, die Position des KLTs während des Anfluges in der semantischen Karte zu erfassen.

Nach dem autonomen Anflug positioniert sich der Flugroboter in geeigneter Höhe (ca. 1,5 m) über dem KLT. Um den KLT aufnehmen zu können muss der Flugroboter präzise auf diesem landen. Hierzu muss zunächst der KLT erkannt und die Pose des Flugroboters relativ zu diesem kontinuierlich und präzise (Genauigkeit im Zentimeterbereich) geschätzt werden. Für die initiale Erkennung des KLTs mit der nach unten gerichteten Kamera ist diese Flughöhe erforderlich, damit der Kleinladungsträger auch dann erkannt werden kann, wenn der Flugroboter nicht exakt darüber schwebt. So ist der Erfassungsbereich der zum Boden gerichteten Kamera noch groß genug, sodass der KLT auch mit einer gewissen Abweichung der Position

des Flugroboters zu diesem erfasst werden kann. Ist der Flugroboter jedoch zu weit entfernt bzw. zu nah am KLT, so kann dieser nicht im Kamerabild erkannt werden und so auch keine relative Poseschätzung des Flugroboters zum KLT für die präzise Landung und Aufnahme erfolgen.

Sobald der zu transportierende KLT erkannt wurde und eine initiale relative Poseschätzung erfolgt ist, werden in die Flugsteuerung nicht mehr die Poseinformationen basierend auf dem UWB-Systems übergeben, sondern die Poseschätzung relativ zum KLT. Um einen Sprung der Pose (absolute Pose im Raum vom UWB-System gegenüber relativer Pose zum KLT) zu vermeiden, wird initial vor der Einspeisung die Pose des KLTs im Raum bestimmt. Hierzu wird die Pose des KLTs mittels der relativ bestimmten Pose vom Flugroboter zu diesem und der gleichzeitig geschätzten Pose des Flugroboters im Raum (EKF gestützt mit UWB-Pose) in das globale Koordinatensystem des UWB-Systems transformiert. Beim Übergang der Einspeisung zur KLT-Posebestimmung entsteht somit kein großer Sprung.

Nachdem die Poseschätzung des Flugroboters durch die Poseschätzung relativ zum KLT gestützt und somit präziser im Vergleich zur UWB-System-Poseschätzung ist, kann der Aufnahmeprozess gestartet werden. Hierfür sinkt der Roboter in zwei Stufen nahe über den KLT ab. Es werden hierfür zwei Stufen genutzt, da durch die Bewegung des Roboters die Erkennung und Posebestimmung mittels des Kamerabildes ungenauer werden kann. Durch den zweistufigen Absinkvorgang hält der Roboter zweimal die Höhe, bewegt sich also nicht mehr stark, und kann sich so wieder über dem KLT zentrieren und möglich Ungenauigkeiten in der Positionierung korrigieren. Sobald sich der Roboter ca. 20 Zentimeter über dem KLT befindet, wird der Flugsteuerung der Befehl gegeben, in den Landemodus zu wechseln. Der Flugroboter landet anschließend auf dem KLT. Um während diese letzten Schritte eine mögliche Abweichung der Zentrierung über dem KLT und so ein fehlerhaftes Landen auf dem KLT zu vermeiden, sind entsprechende Führungsschienen am Flugroboter angebracht. Nach der Landung auf dem KLT wird dieser durch die erarbeitete Aufnahmevorrichtung am Flugroboter fixiert und kann so von diesem transportiert werden.

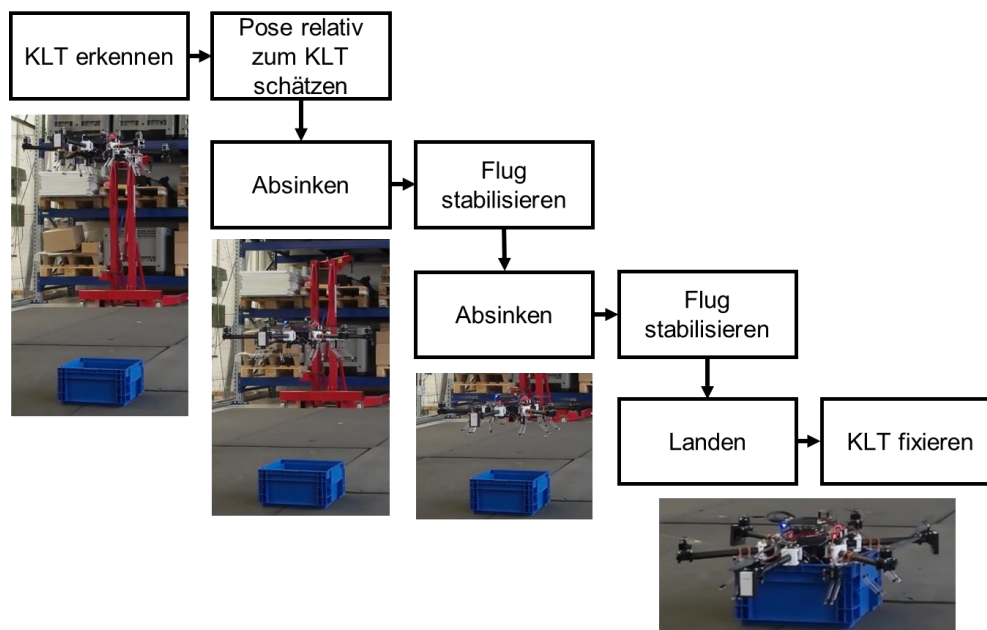


Abbildung 29: Konzept zur vollautomatisierten Aufnahme eines KLTs durch den Flugroboter

Nach Fixierung des KLTs wird die für die Poseschätzung der Flugsteuerung wieder durch das UWB-System gestützt. Der Flugroboter kann den KLT zur gewünschten Zielposition fliegen, dort mit dem KLT landen, die Fixierung lösen und wieder ohne den KLT starten. Eine Parameteranpassung für den Flug mit KLT ist nicht erforderlich, da das veränderte Flugverhalten dynamisch durch die Flugsteuerung ausgeglichen werden kann.

3.6.3 Erkennung des Kleinladungsträgers und Posebestimmung relativ zu diesem

Im Rahmen des Projektes wurden zwei Ansätze zur Erkennung und relativen Poseschätzung mittels Kamerabildern erarbeitet bzw. untersucht. Der erste basiert auf optischen Markern, AprilTags [29, 30], die am oder im KLT befestigt werden. Der zweite erweitert den in [77] vorgestellten Ansatz und basiert auf der Farbe und Form des KLTs zu dessen Erkennung im Kamerabild, sowie auf den bekannten Maßen des KLTs zur Posebestimmung.

Optische Marker basieren auf definierten Mustern und können relativ einfach in der Umgebung angebracht werden. Für die Erkennung und Posebestimmung basierend auf dem optischen Marker ist es ausreichend, das Muster in einer festgelegten Größe auf Papier zu drucken. Die Größe sollte hierbei so gewählt werden, dass das Muster in der gewünschten Kameradistanz dazu noch ausreichend groß und gut im Kamerabild sichtbar ist und so erkannt werden kann. Das definierte Muster kann dann mithilfe von Bildverarbeitungsalgorithmen in den aufgenommenen Kamerabildern erkannt werden. Soweit die reale Größe des Markers sowie die intrinsischen Kameraparameter bekannt sind, kann auch mit dem passenden Kameramodell die Pose der Kamera relativ zum Marker berechnet werden. [30]

Schwierigkeiten bei der Erkennung können sich durch schlechte Lichtverhältnis (siehe Abbildung 30) oder durch eine falsche Größe des Markers ergeben. Zudem könnte der Marker verdeckt und so nicht mehr sichtbar sein. Dies ist insbesondere im hier betrachteten Anwendungsfall relevant, da für den Materialtransport der KLT befüllt sein soll, eine Anbringung des Markers, wie sie in Abbildung 30 zu sehen ist, also nur bedingt geeignet ist. Dennoch wurde zur Validierung der optische Marker im Rahmen des Projektes integriert und untersucht. Einerseits ist der Vergleich zur im Folgenden beschrieben, selbst konzeptionierten und erarbeiteten Methode sinnvoll, da die Erkennung und Poseschätzung basierend auf optischen Mar-

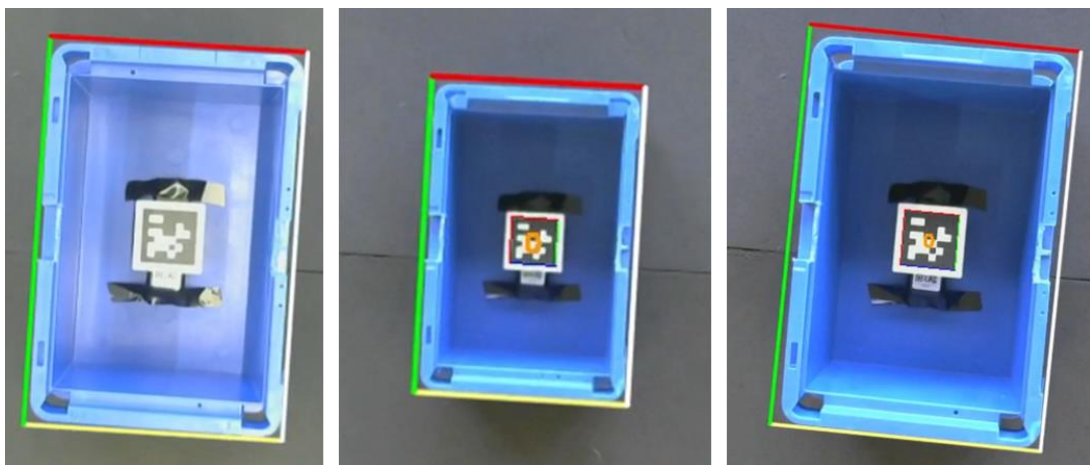


Abbildung 30: Im linken Bild wird der KLT nur durch den neu erarbeitenden Ansatz erkannt, die Erkennung des mittig im KLT angebrachten AprilTags ist durch die Beleuchtung und Reflektionen im KLT nicht möglich; Im mittleren und im rechten Bild werden sowohl der KLT als auch der AprilTag im Kamerabild erkannt

kern sehr robust und präzise ist [29]. Andererseits ist die Nutzung optischer Marker im Anwendungskontext für bestimmte Szenarien durchaus sinnvoll, z. B. für das präzise Landen an einem definierten Ort wie einer Ladestation.

Um den KLT auch im Bild der nach unten gerichteten Kamera unabhängig von einem optischen Marker zu erkennen und die Pose des Roboters relativ dazu schätzen zu können, wurde ein Ansatz mittels der OpenCV-Bibliothek ([26]) erarbeitet, der die Methode zur KLT Erkennung aus [77] erweitert und optimiert.

Im ersten Schritt wird das Kamerabild vom RGB-Farbraum in den HSV-Farbraum (Hue / Farbton, Saturation / Sättigung, Value / Hellwert) überführt. Hierdurch ist die folgende Segmentierung des Farbbildes und Erkennung des Kleinladungsträger basierend auf dessen blauer Farbe einfacher zu realisieren.

Im segmentierten Bild werden anschließend Kanten, in diesem Fall also Übergänge von vorwiegend blauen Bildbereichen zu nicht vorwiegend blauen Bereichen detektiert. Aus den detektierten Kanten werden anschließend Konturen abgeleitet. Um diese Konturen zu glätten und möglichst die Rechteckform des realen KLTs zu erhalten, können die Konturen mithilfe des Ramer-Douglas-Peucker-Algorithmus geglättet werden [78]. Aufgrund der unterschiedlichen Perspektiven auf den KLT ist es jedoch möglich, dass die im Bild erkannte Kontur nicht immer einem Rechteck mit vier Eckpunkten entspricht, sondern durch ein Polygon mit mehr als vier Ecken angenähert wird. Der nächste Verarbeitungsschritt besteht darin, zu bestimmen, ob eine erkannte Kontur tatsächlich einen Kleinladungsträger abbildet, oder durch ein anderes Objekt hervorgerufen wurde. Hierzu wird für jede erkannte Kontur das minimale umschließende Rechteck berechnet. Falls das Flächenverhältnis der Fläche, die die erkannte Kontur umschließt, zur Fläche, die das minimale umschließende Rechteck dieser Kontur beinhaltet, größer als ein Schwellwert (in der folgenden Evaluation 0,85) ist, wird diese Kontur als einen KLT abbildend definiert. Weitere Entscheidungskriterien wie eine minimale bzw. maximale Anzahl an Pixeln, die die Kontur umschließen sollte, oder die Winkel zwischen den Konturpunkten können die Erkennungsqualität steigern.

Nach der Erkennung eines KLTs und der Bestimmung von dessen Eckpunkten im Bild wird mithilfe der intrinsischen Kameraparameter sowie den Eckpunkten des realen KLTs und deren Lage zueinander die relative Pose des Flugroboters in Bezug auf den KLT bestimmt. Hierzu kann der Perspective-N-Point-Algorithmus (PnP-Algorithmus) genutzt werden [42, 79]. Eine robustere Bestimmung der Pose kann durch die Einbeziehung des RANSAC-Algorithmus [79] erzielt werden. Der Ansatz basierend auf dem PnP-Algorithmus ist hier anwendbar, da die Größe des realen KLTs bekannt und stets konstant ist.

3.6.4 System zur Aufnahme und zum Ablegen des Kleinladungsträgers

Das System, das zum Aufnehmen und Ablegen des KLTs konzipiert, optimiert und gebaut wurde ist in Abbildung 31 zu sehen ist. Sie besteht aus acht Führungsschienen zur Zentrierung des Flugroboters beim Landen auf dem KLT sowie aus vier Modellbauservomotoren (Modelcraft Mini Servo MC1811), an denen Arme zur Fixierung und Sicherung des KLTs am Flugroboter angebracht ist. Nach dem Landen auf dem KLT drehen die Servo-Motoren die Arme unter herausstehende Kanten am KLT und halten sie dort. Beim Abheben des Flugroboters liegen die Kanten des KLTs entsprechend auf den vier Armen auf, sodass der KLT vom Flugroboter angehoben wird. Die Ansteuerung der Servomotoren erfolgt

durch eine Mikrocontrollerplattform (PCA9685). Die Führungsschienen sorgen auch dafür, dass der KLT im Flug nicht verrutschen kann.

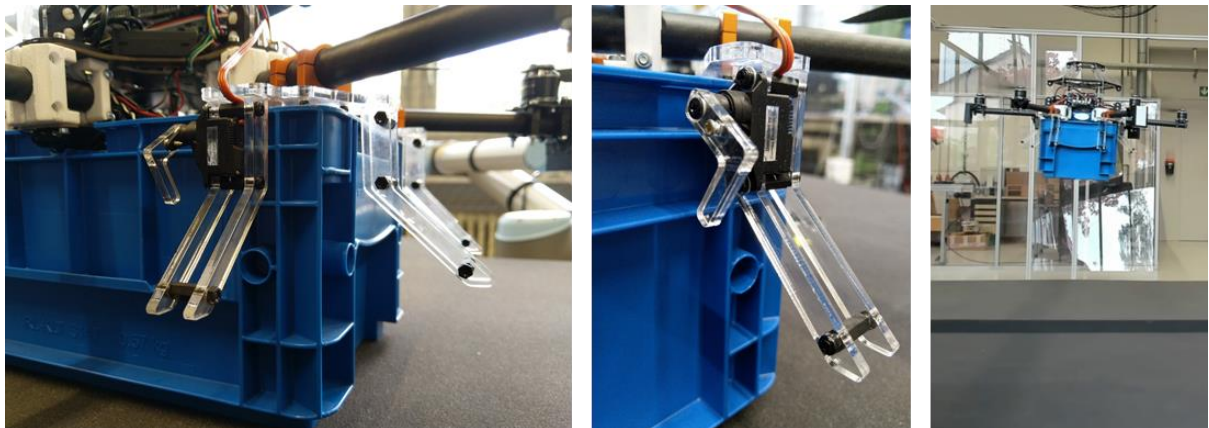


Abbildung 31: Links und Mitte: Führungsschienen der Aufnahmevorrichtung und Motor mit Arm zur Fixierung des KLTs; Rechts: Flugroboter beim KLT-Transport

Die Fixierung an den vier Punkten ist ausreichend, sodass der KLT sicher durch die Arme gehalten wird. Zur Redundanz und Absicherung gegen das Herabfallen des KLTs bei Defekten des Systems wäre es möglich, weitere Arme, die den KLT halten, zu integrieren. Hierdurch nimmt aber das Gewicht der Aufnahmevorrichtung zu, was die Flugzeit und die mögliche Nutzlastzuladung verringert. Aus diesem Grund wurden auch die Führungsschienen mit möglichst wenig Material bei dennoch ausreichender Stabilität und Robustheit aus Acryl gefertigt. Es sind jeweils zwei Führungsschienen für jede KLT-Seite am Flugroboter angebracht, damit der Flugroboter zentriert und nicht verdreht auf dem KLT landet, sodass die Fixierung des KLTs durch die Arme in jedem Fall sichergestellt wird. Das Lösen des KLTs vom Flugroboter nach dem Transport erfolgt durch das Herausdrehen der Arme mittels der Servomotoren, sodass diese nicht mehr unter einer KLT-Kante liegen.

3.6.5 Evaluation und Bewertung der Handhabung des Kleinladungsträgers

Ein Vergleich der Lokalisierung in x- und y-Koordinate beim Flug über einem KLT basierend auf dem UWB-System, dem im Projekt erarbeiteten Ansatz und einem AprilTag ist in Abbildung 32 und 33 dargestellt. Der Flugroboter schwebte für die Erstellung der Plots in unterschiedlichen Höhen zwischen 1,5 m und 0,6 m über dem KLT. Die UWB-Lokalisierung war nicht durch Objekte zwischen Tags und Ankern beeinflusst.

Der im Projekt erarbeitete Algorithmus erkennt den KLT während des gesamten Flugs in allen Distanzen zum KLT, wobei häufig Lücken in der Positionsbestimmung auffallen. Diese Lücken entstehen dadurch, dass der KLT hier in den Kamerabildern nicht erkannt wird. Die Genauigkeit bei der Poseschätzung entspricht nahezu der des AprilTags.

Der AprilTag wird erst nach circa zwanzig Sekunden erkannt. An den Positionen zu Beginn des Fluges waren ungünstige Reflektion und eine Überbeleuchtung im KLT vorhanden (siehe Abbildung 30 links). Nach einer kleinen Veränderung der y-Position und der Flughöhe ist die Beleuchtung des KLTs besser, sodass nun auch der AprilTag sehr gut erkannt wird.

Das UWB-System lokalisiert den Flugroboter durchgängig, jedoch weicht die Lokalisierung insbesondere gegen Ende des Fluges stärker von den beiden anderen Ansätzen ab. In diesem Fall sind diese Abweichung noch sehr gering, jedoch muss bedacht werden, dass die Lokalisierung des UWB-Systems durch Objekte in der Umgebung verschlechtert werden

kann. Aber auch die geringe Abweichung von 5 bis 6 Zentimetern bei der Positionierung über dem KLT bei einer Flughöhe von 0,6 m ist nicht ausreichend für eine erfolgreiche Landung auf dem KLT, trotz der Führungsschienen.

Insgesamt ist der erarbeitete Algorithmus geeignet für eine sichere Landung auf dem KLT und kann den KLT auch bei schlechter Beleuchtung erkennen und die Pose relativ zu diesem bestimmen. Dennoch könnte der Algorithmus noch robuster gestaltet werden, da während des gesamten Fluges insgesamt 1531 Posebestimmungen mittels des AprilTags erfolgt sind, während nur 864 Posebestimmung mittels des erarbeiteten Algorithmus erfolgten. Eine optimierte Parameterwahl sowie die dynamische Einbeziehung der Beleuchtung in den Algorithmus könnten zu einem robusteren Ergebnis führen.

Der Aufnahme- und Ablageprozess wurde in zahlreichen Experimenten validiert. Die erarbeitete Aufnahmevorrichtung hat den KLT stets zuverlässig fixiert und vom Flugroboter gelöst. Voraussetzung für die Aufnahme und auch die Ablage des KLTs ist jedoch, dass ausreichend Platz (ca. 50 cm) um den KLT verfügbar ist. Der KLT muss also vereinzelt auf einer ebenen Fläche stehen.

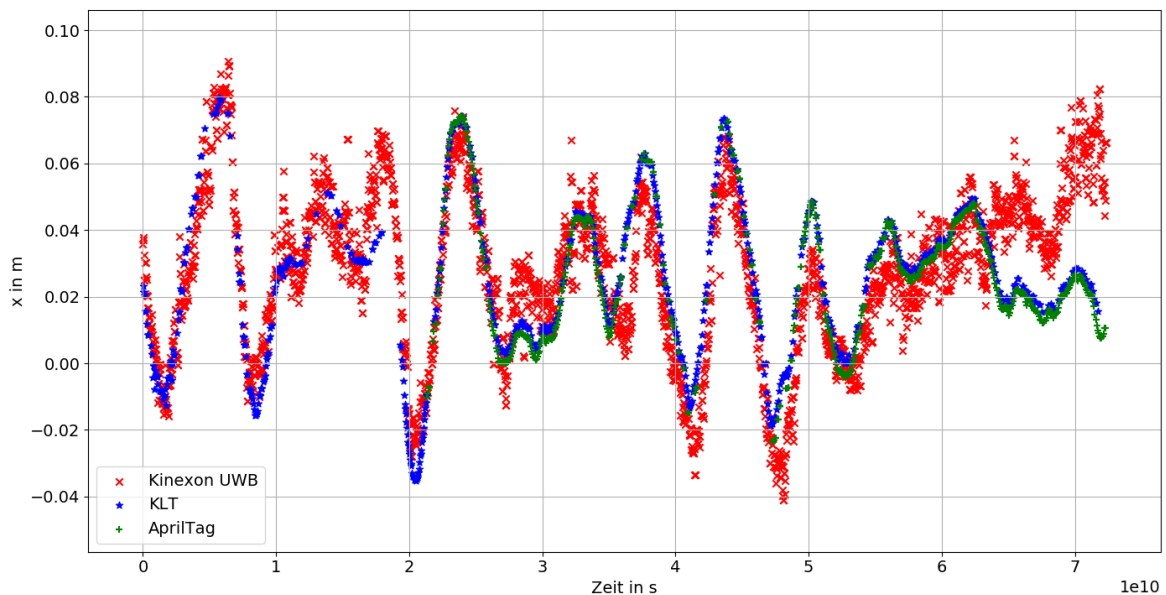


Abbildung 32: Plot der x-Koordinate bestimmt durch das UWB-System, den im Projekt erarbeiteten Ansatz und mittels eines AprilTags während eines Fluges über einem KLT in variierender Höhe von 1,5 m bis 0,6 m

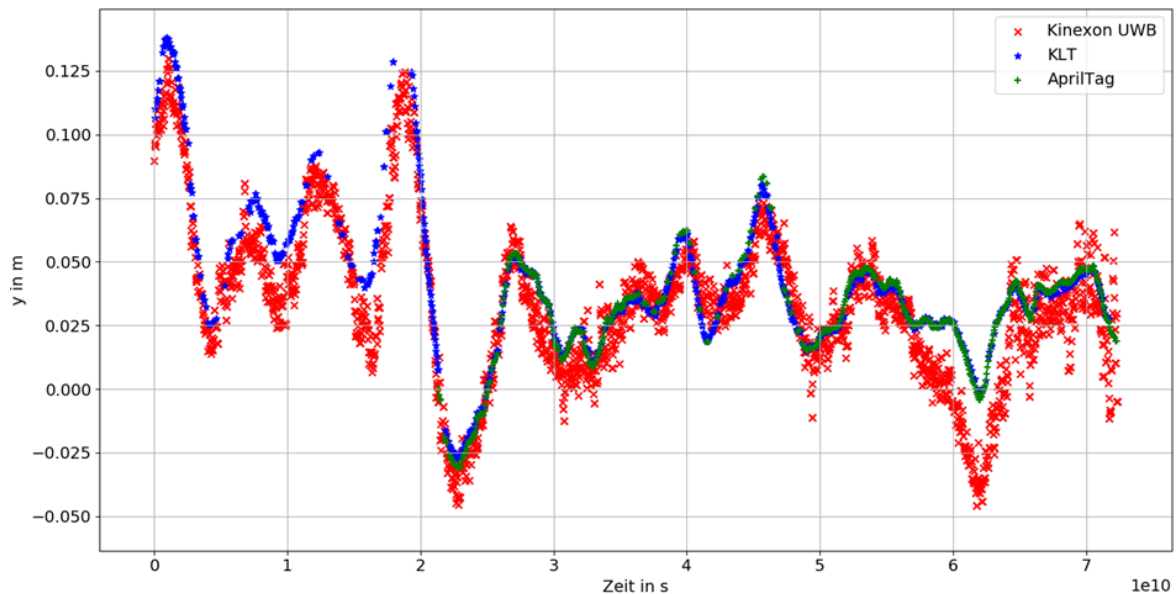


Abbildung 33: Plot der y-Koordinate bestimmt durch das UWB-System, den im Projekt erarbeiteten Ansatz und mittels eines AprilTags während eines Fluges über einem KLT in variierender Höhe von 1,5 m bis 0,6 m

3.7 Gesamtbewertung und Fazit

Der vollautonome Materialtransport mit dem Flugroboter basierend auf dem vorgestellten Konzept wurde erfolgreich realisiert und in zahlreichen Versuchen validiert. Exemplarisch wird hier ein Versuchsszenario vorgestellt, bei dem ein am Boden stehender KLT vom Flugroboter auf einen Tisch transportiert und dort abgelegt wurde. Der Flugroboter startete in einer Dockingstation ([80]) und kehrte dorthin nach dem Transport zurück. Von dort startete er wieder und transportierte den KLT vom Tisch zurück zur Ausgangsposition am Boden und kehrte in die Station zurück. In einem dritten Durchlauf, wurde der KLT analog zum ersten Durchlauf wieder vom Boden zum Tisch transportiert. Der Ablauf ist skizzenhaft in Abbildung 35 eingezeichnet. Die grünen Pfeile stellen den Flugweg beim ersten und dritten Transportflug da, die gelben Pfeile den Weg beim zweiten Transportflug.

Der KLT konnte während des Versuchs dreimal ohne Probleme und Schwierigkeiten aufgenommen, transportiert und wieder abgelegt werden. Der Poseschätzung mittels UWB-System hat hier eine ausreichende Genauigkeit und Robustheit für den autonomen Flug gezeigt.

Der Schubverlauf während des Versuchs ist in Abbildung 35 zu sehen. Hierbei sind die Flugphasen, in denen der KLT transportiert wurde klar erkennbar. Ohne KLT schwebt der Multicopter mit ca. 50% des maximalen Schubs, mit KLT mit ca. 60%. Der KLT hat hierbei ein Zusatzgewicht von 570 Gramm. Zusätzlich war während des gesamten Fluges eine Vorrichtung zum Andocken in der Dockingstation mit einem Gewicht von ca. 200 Gramm angebracht. Somit kann der Flugroboter problemlos eine Nutzlast von 770 g transportieren. Da mit dieser Nutzlast nur ein maximaler Schub von 60% erreicht wurde, ist eine geringe Erhöhung der Nutzlast möglich, ohne das Flugverhalten stark zu beeinflussen.

Wie bereits beschrieben, müsste für die vollständige Autonomie des Flugroboters jedoch noch weitere Sensorik zur Umgebungswahrnehmung und Kollisionsvermeidung angebracht werden und die entsprechenden Softwaremodule ausgeführt werden. Dies ist mit dem im Experiment genutzten Multicopter aufgrund der (aktuell noch) begrenzten Rechenleistung des

OBCs nicht möglich. Zudem sind für die vollständige Autonomie noch weitere Entwicklungsarbeiten für eine robustere, effizientere und umfassendere Softwareimplementierung der im Projekt erarbeiteten und untersuchten Aspekte und Ansätze erforderlich.

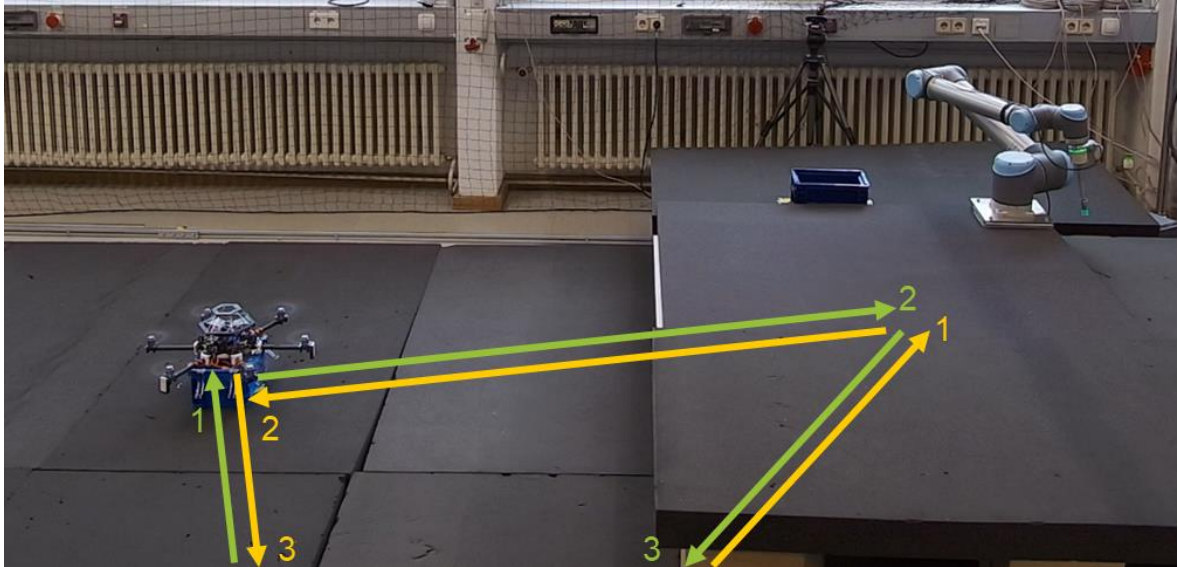


Abbildung 34: Ablauf der Transportflüge; der erste und dritte Transportflug werden durch die grünen Pfeile angedeutet, der zweite Transportflug durch die gelben Pfeile

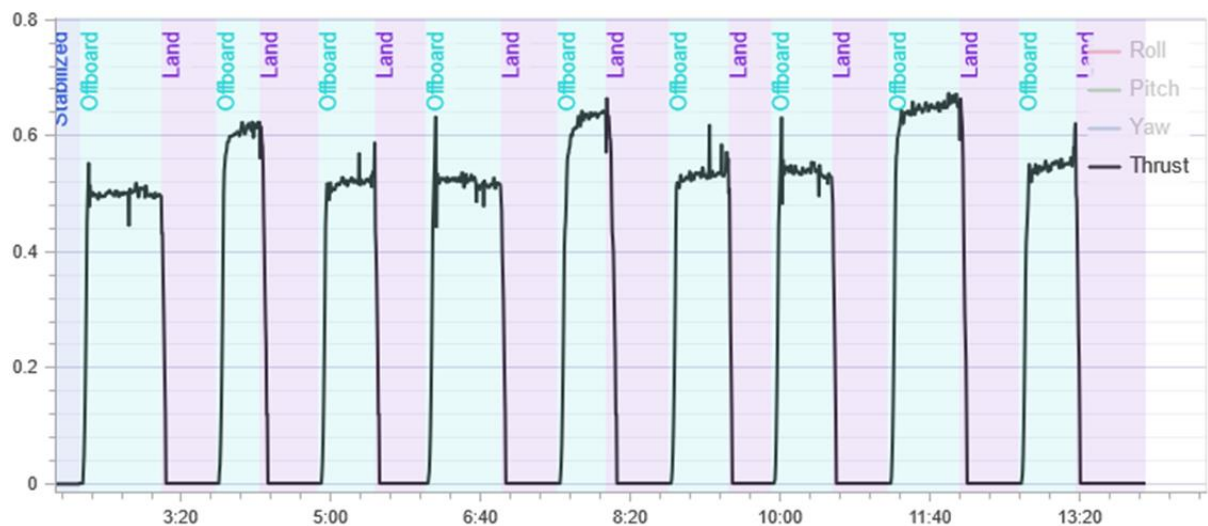


Abbildung 35: Plot des Schubes des Multicopters in Prozent über der Zeit während des Experiments

Die wichtigsten Erkenntnisse des Forschungsprojektes werden im Folgenden nochmals zusammengefasst:

- Der vollautonome Materialtransport mit autonomen Flugrobotern in der Intralogistik ist realisierbar. Die Umsetzbarkeit konnte im Rahmen des Projektes gezeigt und validiert werden.
- Die individuelle Konstruktion und Fertigung des Flugroboters und zusätzlicher Komponenten (z. B. der KLT-Aufnahme) ist erforderlich, da hierfür noch keine Gesamtsysteme am Markt verfügbar sind. Erste Ansätze (z. B. [81]), jedoch noch ohne die vollautomatisierte Nutzlasthandhabung und vollständige Autonomie des Flugroboters, sind bereits in Entwicklung und wahrscheinlich kurz- bis mittelfristig am Markt verfügbar.
- Die Gesamtkonstruktion des Flugroboters (u.a. Anzahl der Motoren, Größe der Propeller, Akkukapazität und -spannung) muss auf das Transportgut und den Anwendungsfall abgestimmt werden. Eine gewisse Flexibilisierung hinsichtlich des Multicopter-aufbaus, insbesondere des Rahmens und der Nutzlastaufnahme, kann mittels 3D-Druck erreicht werden.
- Das Sicherheitskonzept für den Betrieb des Flugroboters muss individuell abhängig von den Umgebungsbedingungen (z. B. Glas oder Kabel in der Umgebung) und Einsatzszenarien (halten sich Menschen im Flugbereich auf) erstellt werden. Im Rahmen des Projektes konnte gezeigt werden, dass die Einhausung der Propeller sowie die erarbeiteten und untersuchten Hard- und Softwareansätze zur Kollisionsvermeidung und semantischen Kartierung für die Absicherung grundlegend geeignet sind und einen Mehrwert bieten. Hier sind jedoch noch weiterführende Entwicklungsarbeiten für effizientere und robustere Software erforderlich.
- Die Lokalisierungsgenauigkeit des UWB-Systems in Kombination mit der optischen Erkennung des Kleinladungsträgers und Poseschätzung relativ zu diesem ist zum autonomen Materialtransport grundlegend ausreichend. Eine weiterentwickelte Poseschätzung unter Einbeziehung einer VIO / eines V-SLAM kann jedoch die Genauigkeit und Robustheit erhöhen.
- Für den industriellen Einsatz ist die redundante Auslegung der Komponenten des Multicopters (FCU, OBC, Sensoren) empfehlenswert.

Folgende Aspekte sind aktuell noch Hürden für den Einsatz von autonomen Flugroboter für den vollautomatisierten Materialtransport:

- Zum heutigen Zeitpunkt sind noch keine Gesetze und Regelungen zum innerbetrieblichen Einsatz von Drohnen vorhanden. So kann beispielsweise nicht genau definiert werden, welche Maßnahmen zur Absicherung getroffen werden müssen. Mit entsprechenden Gesetzen und Regelung ist jedoch in den nächsten Jahren zu rechnen.
- Für den Einsatz von autonomen Flugrobotern sind Erweiterungen und Änderung an der Infrastruktur erforderlich, wie z. B. die Absicherung von gewissen Bereichen, die Installation eines Funkortungssystems sowie eine Ladeinfrastruktur bzw. Ladestation für den Flugroboter
- Die Flexibilität hinsichtlich der Art der Nutzlast ist begrenzt.
- Die Wirtschaftlichkeit muss im Einzelfall geprüft werden, zudem wäre eine geeignete Einbindung von Flugrobotern in ein Intralogistikgesamt-konzept erforderlich.

4 Verwendung der Zuwendungen

Die Projektbearbeitung erfolgte im Zeitraum von Mai 2019 bis Oktober 2021, welcher eine sechsmonatige kostenneutrale Laufzeitverlängerung beinhaltet.

Gemäß dem Antrag wurden 24 Personenmonate für wissenschaftlich-technisches Personal (A1) und 24 Personenmonate für eine studentische Hilfskraft (A3) aufgewendet.

Ausgaben für Gerätebeschaffung (B) sowie Leistungen Dritter (C) waren nicht vorgesehen und sind auch im Laufe des Projektes nicht angefallen.

5 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die durchgeführten Arbeiten sowie die dafür aufgewendeten Ressourcen waren notwendig und angemessen. Sie entsprachen den im Antrag beschriebenen Aufgaben und deckten die Vorgaben des Projektantrages ab.

Die Ergebnisse des Forschungsprojektes wurden den Mitgliedern des Projektbegleitenden Ausschusses in vollem Umfang im Rahmen von Projektbegleitenden Ausschusstreffen und regelmäßigen Berichten kommuniziert und mit ihnen diskutiert.

Durch die Arbeiten und Forschungsergebnisse, die im Projekt erzielt wurden, wird der Weg zur Entwicklung von Flugrobotern für den innerbetrieblichen Einsatz, insbesondere für den Materialtransport, klar aufgezeigt. Hierbei wurden alle relevanten Aspekte, die für den autonomen Betrieb von Flugrobotern in der Intralogistik erforderlich sind, gesamtheitlich betrachtet. Die hierbei identifizierten Hürden und Potentiale helfen den beteiligten Unternehmen, die Technologie und künftigen Einsatz von Flugrobotern in der Intralogistik einzuschätzen und entsprechend zu handeln.

Der zur Veröffentlichung und zur Vorstellung auf einer internationalen Fachkonferenz angenommene Ansatz zur semantischen Kartierung für autonome Flugroboter zeigt, dass die Thematik des Projektes aktuell in der Forschung von Interesse ist und die Arbeiten des Projektes offene Forschungsfragen adressierten.

6 Bewertung des wissenschaftlich-technischen und wirtschaftlichen Nutzens der erzielten Ergebnisse

Durch die Arbeiten und erzielten Ergebnisse wurden Hürden und Potentiale des innerbetrieblichen Einsatzes von Flugroboter, insbesondere beim Materialtransport, aufgezeigt. Die beteiligten Unternehmen können hierdurch einerseits einschätzen, welche weiteren Entwicklungsarbeiten zur Überführung der Technologie in ein Produkt erforderlich sind. Andererseits sind den Unternehmen auch Möglichkeiten des Einsatzes von Drohnen im eigenen Betrieb aufgezeigt worden und können so in deren Planung einbezogen werden.

Da aktuell Drohnen für den innerbetrieblichen Einsatz und damit verbundenen Dienstleistungen vor allem von KMU angeboten werden, können diese von den Projektergebnissen profitieren, indem die im Projekt gewonnenen Erkenntnisse in die Optimierung und Weiterentwicklung ihrer Produkte und Dienstleistungen einfließen lassen.

Des Weiteren sind einige betrachtete Aspekte und Erkenntnisse, wie die semantische Umgebungskartierung, auch auf andere mobile Roboter übertragbar. Hiervon profitieren neben den Herstellern von Intralogistikrobotern auch Unternehmen aus dem Bereich der Servicerobotik, welche immer mehr an Bedeutung gewinnt und deren Einsatzfelder wachsen. Langfristig können somit auch KMU und andere Unternehmen einen Nutzen aus optimierten und intelligenteren mobilen Robotern durch deren effizienteren Einsatz auch bei komplexen Aufgaben und Arbeiten ziehen.

7 Ergebnistransfer in die Wirtschaft

Der Transfer der im Projekt erzielten Ergebnisse erfolgte bereits teilweise zur Laufzeit des Forschungsprojektes. Die Transferstrategie fokussierte sich dabei auf regelmäßige Zwischenberichte über den Fortschritt des Projektes im Rahmen der Sitzungen des projektbegleitenden Ausschusses sowie bilaterale Treffen mit den beteiligten Unternehmen. Bei den fünf Sitzungen des projektbegleitenden Ausschusses wurden die unterschiedlichen Themenschwerpunkte detailliert vorgestellt und diskutiert. Der folgende Abschnitt 7.1 beinhaltet eine detaillierte Darstellung der Transfermaßnahmen, die bereits zur Projektlaufzeit durchgeführt wurden, während der Abschnitt 7.2 die aktuell in Umsetzung befindlichen Maßnahmen aufführt.

7.1 Transfermaßnahmen in der Laufzeit des Forschungsprojektes

Während der Projektlaufzeit wurden bereits die in Tabelle 2 aufgeführten Maßnahmen für den Transfer und die Diskussion der Ergebnisse durchgeführt.

Tabelle 2: Durchgeführte Transfermaßnahmen während der Laufzeit des Projektes

Maßnahmen	Ziele	Rahmen / Inhalt	Zeitraum
Projektbegleitender Ausschuss	Fortlaufende Information des PA über die Forschungsergebnisse und Lenkung des weiteren Projektfortschritts	Vorstellung der Untersuchungsergebnisse und Abgleich mit den Belangen der Praxis	<ul style="list-style-type: none"> • Juli 2019 • Januar 2020 • Juli 2020 • Februar 2021 • Oktober 2021
Veröffentlichung des Projektes auf der Website der Forschungseinrichtung	Information interessierter Unternehmen	Während der Laufzeit wurden die Inhalte und Ziele des Projektes auf der Homepage der Forschungseinrichtung	Mai 2019
Beratungsgespräche	Unternehmensspezifische Informationen für alle interessierten Unternehmen, insbesondere KMU	Persönliche detaillierte Beratung von Intralogistik-Unternehmen zum aktuellen Stand des Forschungsprojektes	fortlaufend auf Anfrage und initiativ
Lehrtätigkeit an Hochschulen	Ergebnistransfer in die Lehre	Diverse Vorlesungen des FAPS	seit SS 2020 in Lehre aufgenommen
Veröffentlichung des Projektes auf der Website der IFL	Information der Öffentlichkeit und interessierter Unternehmen	Interview zum aktuellen Projektstatus und den erreichten Zielen	Januar 2021
Sitzung des Wissenschaftlichen Beirats der IFL	Ergebnisdiskussion und Evaluation im Expertenkreis der IFL	Statusbericht und Ergebnisdiskussion mit weiteren forschenden Stellen	September 2021

Maßnahmen	Ziele	Rahmen / Inhalt	Zeitraum
Artikel in Fachzeitschriften	Information der Öffentlichkeit über das Forschungsprojekt und die Inhalte	Informationen zum Projekt und Einordnung der Ergebnisse durch Unternehmen des Projektbegleitenden Ausschusses im VDMA Magazin	Oktober 2021

7.2 Transfermaßnahmen nach Abschluss des Vorhabens

Nach dem Abschluss des Vorhabens sind weitere Transfermaßnahmen geplant. Diese sind in der folgenden Tabelle aufgeführt.

Tabelle 3: Geplante Transfermaßnahmen nach Abschluss des Projektes

Maßnahmen	Ziele	Rahmen / Inhalt	Zeitraum
Ausführlicher praxisnaher Forschungsbericht	Detaillierte Information der Intralogistikbranche sowie der gesamten Fachöffentlichkeit	Forschungsbericht für alle interessierten Kreise	aktuell
Beratung der Industrie einschließlich Demonstration der Forschungsergebnisse	Maßgeschneiderter Transfer der Forschungsergebnisse	Persönliche, detaillierte Beratung von Intralogistik-Unternehmen	aktuell
Wissenschaftliche Veröffentlichungen	Informationen für die Fachöffentlichkeit	Beitrag über die Forschungsergebnisse auf der „17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications“ (VISAPP 2022)	Februar 2022

8 Literaturverzeichnis

- [1] VERBAND UNBEMANNTE LUFTFAHRT. *Analyse des deutschen Drohnenmarktes* [online], März 2021 [Zugriff am: 16. Dezember 2021]. Verfügbar unter: https://www.verband-unbemannte-luftfahrt.de/wp-content/uploads/2021/03/VUL-Marktstudie_2021_DE_f.pdf
- [2] DEUTSCHE POST DHL GROUP. *DHL Express startet Drohnen-Lieferservice in China* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://www.dpdhl.com/de/presse/pressemitteilungen/2019/dhl-express-startet-drohnen-lieferservice-in-china.html>
- [3] AMAZON. *Prime Air* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>
- [4] DJI. *Landwirtschaftsdrohnen* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://ag.dji.com/de?site=brandsite&from=nav>
- [5] SPECTAIR. *SPECTAIR* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://www.spectair.com/drohneninspektion/>
- [6] DOKS. INNOVATION GMBH. *doks. inventairy* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://doks-innovation.com/de/solutions/inventairy-xl>
- [7] KALINOV, I., A. PETROVSKY, V. ILIN, E. PRISTANSKIY, M. KURENKOV, V. RAMZHAEV, I. IDRISOV und D. TSETSERUKOU. WareVision: CNN Barcode Detection-Based UAV Trajectory Optimization for Autonomous Warehouse Stocktaking [online]. *IEEE Robotics and Automation Letters*, 2020, **5**(4), S. 6647-6653. Verfügbar unter: doi:10.1109/LRA.2020.3010733
- [8] SANDINO, J., F. VANEGAS, F. MAIRE, P. CACCETTA, C. SANDERSON und F. GONZALEZ. UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments [online]. *Remote Sensing*, 2020, **12**(20), S. 3386. Verfügbar unter: doi:10.3390/rs12203386
- [9] LIERET, M., V. KOGAN, S. DOLL und J. FRANKE. Automated in-house transportation of small load carriers with autonomous unmanned aerial vehicles. In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*: IEEE, 22. August 2019 - 26. August 2019, S. 1010-1015. ISBN 978-1-7281-0356-3
- [10] WAWRLA, L., O. MAGHAZEI und T. NETLAND. Applications of drones in warehouse operations. *Whitepaper. ETH Zurich, D-MTEC*, 2019
- [11] MAHONY, R., V. KUMAR und P. CORKE. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor [online]. *IEEE Robotics & Automation Magazine*, 2012, **19**(3), S. 20-32. ISSN 1070-9932. Verfügbar unter: doi:10.1109/MRA.2012.2206474
- [12] MARKUS MÜLLER. <https://www.ecalc.ch/xcoptercalc.php> [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://www.ecalc.ch/xcoptercalc.php>
- [13] FRIEDRICH, J., M. LIERET, L. REICH-ROHRWIG, A. SEEL und W. USLAR. UAS im Indoorbereich - Rechtliche und technische Einführung in die Nutzung von Indoor-UAS, 2021. Verfügbar unter: <https://ids.vdma.org/downloads>
- [14] DEUTSCHE GESETZLICHE. *Sicherer Umgang mit Multikoptern (Drohnen)*. DGUV *Information 208-058*, Juli 2020

- [15] LIERET, M., J. FERTSCH und J. FRANKE. Fault detection for autonomous multirotors using a redundant flight control architecture. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*: IEEE, 20. August 2020 - 21. August 2020, S. 29-34. ISBN 978-1-7281-6904-0
- [16] MEIER, L., D. HONEGGER und M. POLLEFEYS. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*: IEEE, 26. Mai 2015 - 30. Mai 2015, S. 6235-6240. ISBN 978-1-4799-6923-4
- [17] *PX4 Autopilot* [online] [Zugriff am: 16. Dezember 2022]. Verfügbar unter: <https://px4.io/>
- [18] *PX4 Drone Autopilot* [online] [Zugriff am: 16. Dezember 2022]. Verfügbar unter: <https://github.com/PX4/PX4-Autopilot>
- [19] *PX4 Multicopter PID Tuning Guide* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: https://docs.px4.io/master/en/config_mc/pid_tuning_guide_multicopter.html
- [20] *PX4 Autopilot User Guide* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://docs.px4.io/master/en/>
- [21] OPEN SOURCE ROBOTICS FOUNDATION, INC. *Robot Operating System* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://www.ros.org/>
- [22] QUIGLEY, M., K. CONLEY, B. GERKEY, J. FAUST, T. FOOTE, J. LEIBS, R. WHEELER, A.Y. NG und OTHERS. ROS: an open-source Robot Operating System. In: *ICRA workshop on open source software*, 2009, S. 5
- [23] ERMAKOV, V. *MAVROS* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <http://wiki.ros.org/mavros>
- [24] *MAVLink Developer Guide* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://mavlink.io/en/>
- [25] *tkinter* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://docs.python.org/3/library/tkinter.html>
- [26] *Open Source Computer Vision Library* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://opencv.org/>
- [27] *Gazebo* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <http://gazebosim.org/>
- [28] *Log Analysis using Flight Review, Vibration* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: https://docs.px4.io/v1.12/en/log/flight_review.html#vibration
- [29] WANG, J. und E. OLSON. AprilTag 2: Efficient and robust fiducial detection. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: IEEE, 9. Oktober 2016 - 14. Oktober 2016, S. 4193-4198. ISBN 978-1-5090-3762-9
- [30] KROGIUS, M., A. HAGGENMILLER und E. OLSON. Flexible Layouts for Fiducial Tags. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: IEEE, 3. November 2019 - 8. November 2019, S. 1898-1903. ISBN 978-1-7281-4004-9
- [31] HERTZBERG, J., K. LINGEMANN und A. NÜCHTER. *Mobile Roboter. Eine Einführung aus Sicht der Informatik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. eXamen.press. ISBN 978-3-642-01725-4
- [32] VANHIE-VAN GERWEN, J., K. GEEBELEN, J. WAN, W. JOSEPH, J. HOEBEKE und E. de POORTER. Indoor Drone Positioning: Accuracy and Cost Trade-off for Sensor Fusion [online]. *IEEE Transactions on Vehicular Technology*, 2021, S. 1. ISSN 0018-9545. Verfügbar unter: doi:10.1109/TVT.2021.3129917

- [33] MILIJAS, R., L. MARKOVIC, A. IVANOVIC, F. PETRIC und S. BOGDAN. A Comparison of LiDAR-based SLAM Systems for Control of Unmanned Aerial Vehicles. In: *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*: IEEE, 15. Juni 2021 - 18. Juni 2021, S. 1148-1154. ISBN 978-1-6654-1535-4
- [34] BULTMANN, S., J. QUENZEL und S. BEHNKE. Real-Time Multi-Modal Semantic Fusion on Unmanned Aerial Vehicles. In: *2021 European Conference on Mobile Robots (ECMR)*: IEEE, 31. August 2021 - 3. September 2021, S. 1-8. ISBN 978-1-6654-1213-1
- [35] QIN, T., P. LI und S. SHEN. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 2018, **34**(4), S. 1004-1020
- [36] CAMPOS, C., R. ELVIRA, J.J.G. RODRIGUEZ, J.M. M. MONTIEL und J. D. TARDOS. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM [online]. *IEEE Transactions on Robotics*, 2021, **37**(6), S. 1874-1890. ISSN 1552-3098. Verfügbar unter: doi:10.1109/TRO.2021.3075644
- [37] KINEXON GMBH. *Echtzeit-Lokalisierung (RTLS)* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://kinexon.com/de/technologie/echtzeit-lokalisierung-rtls/>
- [38] KRAM, S., M. STAHLKE, T. FEIGL, J. SEITZ und J. THIELECKE. UWB Channel Impulse Responses for Positioning in Complex Environments: A Detailed Feature Analysis [online]. *Sensors (Basel, Switzerland)*, 2019, **19**(24). Verfügbar unter: doi:10.3390/s19245547
- [39] INTEL CORPORATION. *Intel RealSense Tracking Camera T265* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://www.intelrealsense.com/tracking-camera-t265/>
- [40] INTEL CORPORATION. *Depth Camera D435* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://www.intelrealsense.com/depth-camera-d435/>
- [41] BISHOP, C.M. *Pattern recognition and machine learning*. New York, NY: Springer Science+Business Media LLC, 2006. Information Science and Statistics. ISBN 0-387-31073-8
- [42] HARTLEY, R. und A. ZISSERMAN. *Multiple view geometry in computer vision*. Second edition. Cambridge: Cambridge University Press, 2011. ISBN 9780511811685
- [43] TONG QIN, JIE PAN, SHAOZU CAO und SHAOJIE SHEN. A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors, 2019
- [44] QIN, T. und S. SHEN. Online Temporal Calibration for Monocular Visual-Inertial Systems. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, S. 3662-3669
- [45] QIN, T., S. CAO, J. PAN, P. LI und S. SHEN. *VINS-FUSION* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://github.com/HKUST-Aerial-Robotics/VINS-Fusion>
- [46] PARAZERO DRONE SAFETY SYSTEMS LTD. *SafeAir Mavic* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://parazero.com/products/safeair-for-dji-mavic/>
- [47] POUNDS, P.E.I. und W. DEER. The Safety Rotor—An Electromechanical Rotor Safety System for Drones [online]. *IEEE Robotics and Automation Letters*, 2018, **3**(3), S. 2561-2568. Verfügbar unter: doi:10.1109/LRA.2018.2811842
- [48] BRIOD, A., P. KORNIATOWSKI, J.-C. ZUFFEREY und D. FLOREANO. A Collision-resilient Flying Robot [online]. *Journal of Field Robotics*, 2014, **31**(4), S. 496-509. ISSN 15564959. Verfügbar unter: doi:10.1002/rob.21495

- [49] STECKEL, J. und H. PEREMANS. Biomimetic sonar for biomimetic SLAM. In: *2012 IEEE Sensors*: IEEE, 28. Oktober 2012 - 31. Oktober 2012, S. 1-4. ISBN 978-1-4577-1767-3
- [50] LABBÉ, M. und F. MICHAUD. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation [online]. *Journal of Field Robotics*, 2019, **36**(2), S. 416-446. ISSN 15564959. Verfügbar unter: doi:10.1002/rob.21831
- [51] HORNING, A., K.M. WURM, M. BENNEWITZ, C. STACHNISS und W. BURGARD. OctoMap: an efficient probabilistic 3D mapping framework based on octrees [online]. *Autonomous Robots*, 2013, **34**(3), S. 189-206. ISSN 0929-5593. Verfügbar unter: doi:10.1007/s10514-012-9321-0
- [52] NAKAJIMA, Y. und H. SAITO. Efficient Object-Oriented Semantic Mapping With Object Detector [online]. *IEEE Access*, 2019, **7**, S. 3206-3213. Verfügbar unter: doi:10.1109/ACCESS.2018.2887022
- [53] SUNDERHAUF, N., T.T. PHAM, Y. LATIF, M. MILFORD und I. REID. Meaningful maps with object-oriented semantic mapping. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: IEEE, 24. September 2017 - 28. September 2017, S. 5079-5085. ISBN 978-1-5386-2682-5
- [54] GRINVALD, M., F. FURRER, T. NOVKOVIC, J.J. CHUNG, C. CADENA, R. SIEGWART und J. NIETO. Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery [online]. *IEEE Robotics and Automation Letters*, 2019, **4**(3), S. 3037-3044. Verfügbar unter: doi:10.1109/LRA.2019.2923960
- [55] MATURANA, D., S. ARORA und S. SCHERER. Looking forward: A semantic mapping system for scouting with micro-aerial vehicles. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*: IEEE, 24. September 2017 - 28. September 2017, S. 6691-6698. ISBN 978-1-5386-2682-5
- [56] KOCH, T., M. KÖRNER und F. FRAUNDORFER. Automatic and Semantically-Aware 3D UAV Flight Planning for Image-Based 3D Reconstruction [online]. *Remote Sensing*, 2019, **11**(13), S. 1550. Verfügbar unter: doi:10.3390/rs11131550
- [57] HOFMANN, C., FICHTNER, M., LIERET, M. AND FRANKE, J. Efficient Semantic Mapping in Dynamic Environments. Zur Veröffentlichung angenommen. In: *In Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022)*
- [58] HOFMANN, C., F. PARTICKE, M. HILLER und J. THIELECKE. Object Detection, Classification and Localization by Infrastructural Stereo Cameras. In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*: SCITEPRESS - Science and Technology Publications, 25. Februar 2019 - 27. Februar 2019, S. 808-815. ISBN 978-989-758-354-4
- [59] SOLOVYEV, R., W. WANG und T. GABRUSEVA. Weighted boxes fusion: Ensembling boxes from different object detection models [online]. *Image and Vision Computing*, 2021, **107**, S. 104117. ISSN 02628856. Verfügbar unter: doi:10.1016/j.imavis.2021.104117
- [60] WOJKE, N., A. BEWLEY und D. PAULUS. Simple online and realtime tracking with a deep association metric. In: *2017 IEEE International Conference on Image Processing (ICIP)*: IEEE, 17. September 2017 - 20. September 2017, S. 3645-3649. ISBN 978-1-5090-2175-8

- [61] RUSU, R.B. und S. COUSINS. 3D is here: Point Cloud Library (PCL). In: *2011 IEEE International Conference on Robotics and Automation*: IEEE, 9. Mai 2011 - 13. Mai 2011, S. 1-4. ISBN 978-1-61284-386-5
- [62] ROSEN, D.M., J. MASON und J.J. LEONARD. Towards lifelong feature-based mapping in semi-static environments. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*: IEEE, 16. Mai 2016 - 21. Mai 2016, S. 1063-1070. ISBN 978-1-4673-8026-3
- [63] MARDER-EPPSTEIN, E., M. FERGUSON, D. LU und A. HOY. *ROS Navigation Stack* [online] [Zugriff am: 16. Dezember 2022]. Verfügbar unter: <http://wiki.ros.org/navigation>
- [64] LU, D.V., D. HERSHBERGER und W.D. SMART. Layered costmaps for context-sensitive navigation. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*: IEEE, 14. September 2014 - 18. September 2014, S. 709-715. ISBN 978-1-4799-6934-0
- [65] LU, D.V. *Social Navigation Layer* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: http://wiki.ros.org/social_navigation_layers
- [66] AGGARWAL, S. und N. KUMAR. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges [online]. *Computer Communications*, 2020, **149**(12), S. 270-299. ISSN 01403664. Verfügbar unter: doi:10.1016/j.comcom.2019.10.014
- [67] DU, R. und R.V. COWLAGI. Interactive sensing and path-planning with incremental 3D path repair for a quadrotor UAV in cluttered and partially known environments. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*: IEEE, 12. Dezember 2017 - 15. Dezember 2017, S. 933-938. ISBN 978-1-5090-2873-3
- [68] LIERET, M., M. KALENBERG, C. HOFMANN und J. FRANKE. Environment representation and path planning for unmanned aircraft in industrial indoor applications [online]. *Procedia Manufacturing*, 2021, **55**(2), S. 176-182. ISSN 23519789. Verfügbar unter: doi:10.1016/j.promfg.2021.10.025
- [69] LV, Z., L. YANG, Y. HE, Z. LIU und Z. HAN. 3D environment modeling with height dimension reduction and path planning for UAV. In: *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*: IEEE, 10. Juli 2017 - 12. Juli 2017, S. 734-739. ISBN 978-1-5090-6575-2
- [70] RÖSMANN, C., F. HOFFMANN und T. BERTRAM. Integrated online trajectory planning and optimization in distinctive topologies [online]. *Robotics and Autonomous Systems*, 2017, **88**(6), S. 142-153. ISSN 09218890. Verfügbar unter: doi:10.1016/j.robot.2016.11.007
- [71] BOCHKOVSKIY, A., C.-Y. WANG und H.-Y.M. LIAO. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. <https://arxiv.org/abs/2004.10934>, 23. April 2020
- [72] YI, K.M., K. YUN, S.W. KIM, H.J. CHANG und J.Y. CHOI. Detection of Moving Objects with Non-stationary Cameras in 5.8ms: Bringing Motion Detection to Your Mobile Device. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*: IEEE, 23. Juni 2013 - 28. Juni 2013, S. 27-34. ISBN 978-0-7695-4990-3
- [73] POUNDS, P.E.I., D.R. BERSAK und A.M. DOLLAR. Practical aerial grasping of unstructured objects. In: *2011 IEEE Conference on Technologies for Practical Robot Applications*: IEEE, 11. April 2011 - 12. April 2011, S. 99-104. ISBN 978-1-61284-482-4
- [74] QI, J., J. KANG und X. LU. Design and research of UAV autonomous grasping system. In: *2017 IEEE International Conference on Unmanned Systems (ICUS)*: IEEE, 27. Oktober 2017 - 29. Oktober 2017, S. 126-131. ISBN 978-1-5386-3107-2

- [75] LIERET, M., J. LUKAS, M. NIKOL und J. FRANKE. A lightweight, low-cost and self-diagnosing mechatronic jaw gripper for the aerial picking with unmanned aerial vehicles [online]. *Procedia Manufacturing*, 2020, **51**, S. 424-430. ISSN 23519789. Verfügbar unter: doi:10.1016/j.promfg.2020.10.060
- [76] GAWEL, A., M. KAMEL, T. NOVKOVIC, J. WIDAUER, D. SCHINDLER, B.P. von ALTISHOFEN, R. SIEGWART und J. NIETO. Aerial picking and delivery of magnetic objects with MAVs. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*: IEEE, 29. Mai 2017 - 3. Juni 2017, S. 5746-5752. ISBN 978-1-5090-4633-1
- [77] LIERET, M., M. LALLINGER, M. TAUSCHER und J. FRANKE. Localization and grasping of small load carriers with autonomous unmanned aerial vehicles. In: T. SCHÜPPSTUHL, K. TRACHT und D. HENRICH, Hg. *Annals of Scientific Society for Assembly, Handling and Industrial Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, S. 241-250. ISBN 978-3-662-61754-0
- [78] DOUGLAS, D.H. und T.K. PEUCKER. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 1973, **10(2)**, S. 112-122
- [79] *Camera Calibration and 3D Reconstruction* [online] [Zugriff am: 16. Dezember 2022]. Verfügbar unter: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga50620f0e26e02caa2e9adc07b5fbf24e
- [80] LIERET, M., F. WURMER, C. HOFMANN und J. FRANKE. An overhead docking and charging station for autonomous unmanned aircraft. In: *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*: IEEE, 23. August 2021 - 27. August 2021, S. 1358-1363. ISBN 978-1-6654-1873-7
- [81] TÜNKERS-NICKEL SYSTEME GMBH. *TÜNKERS-NICKEL Systeme GmbH, Transportdrohnen* [online] [Zugriff am: 16. Dezember 2021]. Verfügbar unter: <https://tuenkers-nickel.de/transportdrohnen/>