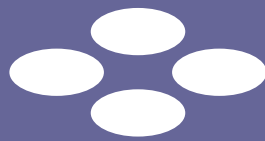


Dissertation

Nonverbal Human-Drone Interaction



Marc Roman Lieser
2024

NONVERBAL HUMAN-DRONE INTERACTION

A dissertation to obtain the degree
DOCTOR OF NATURAL SCIENCES (DR. RER. NAT.)

submitted to

DOCTORAL CENTER APPLIED INFORMATICS
RHEINMAIN UNIVERSITY OF APPLIED SCIENCES

presented by

M. SC. MARC ROMAN LIESER

born on 27 March 1985
in Meisenheim

Supervisors

PROF. DR. ULRICH SCHWANECKE
RheinMain University of Applied Sciences

PROF. DR. JÖRG BERDUX
RheinMain University of Applied Sciences

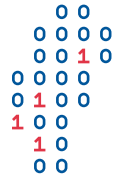
Assessors

PROF. DR. UTE BAUER-WERSING
Frankfurt University of Applied Sciences

PROF. DR. JESSICA R. CAUCHARD
Vienna University of Technology

Submission	Published	Disputation
20.06.2024	WIESBADEN 2024	29.10.2024

Marc Roman Lieser: *Nonverbal Human-Drone Interaction* © 2024
THIS WORK IS LICENSED UNDER CC BY 4.0 © ⓘ



Promotionszentrum
Angewandte Informatik

HAW Hessen



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim



COMPUTER VISION
& MIXED REALITY
GROUP

DECLARATION OF AUTHORSHIP

I, Marc Roman Lieser, hereby declare that this thesis, entitled “Nonverbal Human-Drone Interaction”, is my own work, and all the resources and materials used in this research have been properly acknowledged. I affirm that:

1. The thesis was prepared independently and without unauthorized outside help and only with the specified aids;
2. All text passages taken literally or in spirit from published writings and all information based on verbal information are identified as such;
3. The principles of good scientific practice have been complied with;
4. This thesis has not been submitted for any other degree or qualification at any other institution.

Parts of this thesis are based on the first-authored published articles listed in Appendix B.

I understand that any act of plagiarism, fabrication, falsification, or other forms of academic misconduct in connection with this thesis may lead to the rejection of my work, the revocation of my degree, and other appropriate disciplinary actions.

Wiesbaden, 20.06.2024
Marc Roman Lieser

ABSTRACT

As robots become increasingly integrated into daily life, it is crucial to ensure their effective and safe interaction with humans. Failure to meet human expectations in their responses can quickly lead to frustration. Poorly implemented verbal and nonverbal communication can sour the acceptance of robots, turning positive encounters into unpleasant experiences.

The field of human-drone interaction is dedicated to enhancing the relationship and collaboration between humans and flying robots in various interaction aspects, ultimately improving human-robot encounters in everyday environments. It is essential for robots to comprehend human commands, while humans should intuitively understand a robot's intent. The responsibility lies with robot designers and researchers to craft social robots and provide appropriate interaction methods.

There are robots, however, for which the adoption of established forms to communicate their status or mood through visual cues such as displays, is a challenge. Flying robots often encounter significant limitations in payload capacity, frequently requiring them to prioritize equipment essential for flight-related functionalities. But despite their mechanical design, they can be retrofitted with features that enhance human acceptance and interaction experience, such as exploiting possible channels of nonverbal communication.

This thesis explores various aspects of human-drone interaction through a series of novel approaches. The challenge of drones appearing overly mechanical is addressed by animating their movements. The human inhibition threshold regarding proximity to mechanically designed drones is studied, aiming to assess the potential for improved interaction in confined spaces. For close proximity interaction, virtual buttons are introduced using onboard sensors, eliminating the need for extra hardware. Finally, as a feedback mechanism, drone trajectories are augmented by acoustically perceivable information, employed to complement and improve the differentiation of aerial gestures.

ZUSAMMENFASSUNG

Mit zunehmender Integration von Robotern in den Alltag wird es immer wichtiger, deren effektive und sichere Interaktion mit Menschen zu gewährleisten. Das Versäumnis menschlichen Erwartungen gerecht zu werden, kann schnell zu Frustration führen. Schlechte Kommunikation kann die Akzeptanz von Robotern beeinträchtigen und Begegnungen in unangenehme Erlebnisse verwandeln.

Der Bereich Mensch-Drohnen-Interaktion widmet sich der Verbesserung der Beziehung zwischen Mensch und Drohne in verschiedenen Aspekten der Interaktion, um Begegnungen in alltäglichen Umgebungen aufzuwerten. Es ist wichtig, dass Roboter menschliche Befehle verstehen und dass Menschen die Absichten von Robotern intuitiv erfassen können. Designer und Forscher sind dafür verantwortlich, soziale Roboter zu entwerfen und geeignete Interaktionsmethoden bereitzustellen.

Es gibt jedoch Roboter, bei denen die Übernahme etablierter Methoden zur Kommunikation, z. B. durch Visualisierungen auf angebrachten Displays, eine Herausforderung darstellt. Flugroboter stoßen an die Grenzen ihrer Nutzlastkapazität und müssen häufig der Ausrüstung für flugrelevante Funktionen Vorrang einräumen. Trotz ihres mechanischen Aussehens können sie jedoch um Eigenschaften erweitert werden, die die Akzeptanz und das Interaktionserlebnis verbessern, z. B. indem nonverbale Kommunikationskanäle ausgeschöpft werden.

Diese Arbeit erforscht verschiedene Aspekte der Mensch-Drohnen-Interaktion durch eine Reihe neuartiger Ansätze. Der Herausforderung, mechanisch wirkenden Drohnen lebendiges Verhalten zu verleihen, wird durch die Animation ihrer Bewegungen begegnet. Die menschliche Hemmschwelle in Bezug auf die Nähe zu mechanisch wirkenden Drohnen wird untersucht, um das Potenzial für eine verbesserte Interaktion auf engem Raum zu bewerten. Für die physische Interaktion in unmittelbarer Nähe werden virtuelle Tasten eingeführt, die mithilfe von Onboard-Sensoren realisiert werden und den Bedarf an zusätzlicher Hardware eliminieren. Abschließend werden Drohnen-Trajektorien durch akustisch wahrnehmbare Informationen erweitert, die als Kommunikationsmittel und zur besseren Unterscheidung von Fluggesten eingesetzt werden.

ACKNOWLEDGEMENTS

First of all, I would like to thank Prof. Ulrich Schwanecke, my first supervisor and head of our small Computer Vision & Mixed Reality group. Without this inspiring working environment he founded at RheinMain University of Applied Sciences in Wiesbaden, his trust, his patience, his support, and his constant revival of my motivation, this work would not have been possible. His positivity radiates, and I am glad it still does, even though there was a point when I thought I had finally exhausted the world's greatest optimist.

I would also like to thank Prof. Jörg Berdux for agreeing to be my second supervisor and for the inspiring discussions that sparked some of the ideas that eventually became part of this thesis.

I am very grateful to Prof. Ute Bauer-Wersing, who kindly agreed to be the first assessor of this thesis. Her constructive input and ideas were essential to the completion of this dissertation.

My special thanks go to Prof. Jessica Cauchard. As a leading researcher in the field of human-drone interaction, her work has inspired some of my own, and I am very honored and grateful that she agreed to be the second assessor.

I would like to thank my colleagues for the fruitful daily discussions, for their support and especially for their ideas and contributions, which were indispensable for the development of the drone infrastructure. I would also like to thank all the students who helped shape the testbed by using it for their own research and for trusting me to supervise them. I would like to thank the study participants, as well as the friends and administrative staff at RheinMain University who helped me recruit them. The staff has always provided quick solutions to a variety of challenges over the years.

I am deeply grateful to my family and friends for always being there for me in a supportive and benevolent way, especially for sticking by me even when times got worse and I let my frustration and disappointment show.

CONTENTS

Notation	xvii
Acronyms	xix
1 Introduction	1
1.1 Robots	1
1.2 Human-Robot Coexistence	3
1.2.1 Embodiment	5
1.2.2 Communication	6
1.3 Uncrewed Aerial Vehicles	8
1.3.1 Modern Drones	11
1.3.2 Design Principle	13
1.4 Human-Drone Interaction	18
1.4.1 Kinesics	19
1.4.2 Proxemics	20
1.4.3 Haptics	21
1.4.4 Vocalics	21
1.5 Quadrotor Testbeds	22
1.6 Research Objectives	23
1.7 Thesis Outline	24
2 Background Theory	27
2.1 Rigid-Body Motion	28
2.1.1 Coordinate Frames	29
2.1.2 Representation of Rotation	32
2.1.3 Rotational Motion	38

2.1.4	Coordinate System Transformations	40
2.1.5	Homogeneous Coordinates	42
2.2	Projective Geometry	44
2.2.1	Camera Model	45
2.2.2	Camera Calibration	48
2.3	Quadrotor Dynamics	50
2.3.1	Differential Flatness	55
2.3.2	Trajectories	56
2.4	Quadrotor Control	57
2.4.1	PID Feedback Control	58
2.4.2	Position Control	60
2.4.3	Trajectory Control	62
3	A Testbed for Human-Drone Interaction	67
3.1	Pose Estimation	69
3.1.1	Camera Calibration	71
3.1.2	Quadrotor Pose Estimation	72
3.1.3	Reference Frame Definition	76
3.1.4	Pose Averaging	78
3.1.5	Pose Smoothing	79
3.1.6	Trajectory Visibility Check	80
3.2	Radio Control	82
3.3	Quadrotor System Identification	86
3.3.1	Thrust Mapping	87
3.3.2	Battery Voltage Compensation	89
3.3.3	Motor Identification	91
3.4	Implementation	95
3.4.1	Interface	96
3.4.2	Flight Control	98
3.4.3	Simulation	101
3.4.4	Reality	101
3.5	User Studies	106
3.6	User Code Extensions and Applications	107
3.6.1	FPV Simulator	108
3.6.2	Light Painting	109
3.6.3	Ground Robot Collaboration	109

3.6.4	Inside-out Tracking Extension	111
3.6.5	Inspection of Unknown Environments	112
3.6.6	Reinforcement Learning Control	113
3.6.7	Gesture-based Interaction	113
4	Nonverbal Communication in Human-Drone Interaction	115
4.1	Kinesics	118
4.1.1	Animation Principles	119
4.1.2	User Study	122
4.1.3	Discussion	125
4.2	Proxemics	126
4.2.1	Tolerated Distances	127
4.2.2	User Study	129
4.2.3	Discussion	139
4.3	Haptics	144
4.3.1	Touch Detection	145
4.3.2	Applications	148
4.3.3	Discussion	153
4.4	Vocalics	155
4.4.1	Airborne Gestures	158
4.4.2	User Study	160
4.4.3	Discussion	165
5	Discussion	169
5.1	Integration of Findings	169
5.2	Comparative Analysis	170
5.3	Implications for Human-Drone Interaction	172
5.4	Limitations and Future Directions	173
6	Conclusion	175

A	User Studies	179
A.1	Ethics Statement	179
A.2	Data Summary	181
A.3	Kinesics and Proxemics	182
A.3.1	Questionnaire	182
A.3.2	Research Data	191
A.4	Vocalics	198
A.4.1	Survey Pages	198
A.4.2	Research Data	204
B	Publications	211
	References	213

NOTATION

MATHEMATICAL NOTATION

$\mathcal{M} = \{m_1, m_2, \dots, m_n\}$	A set of n elements like m
\mathbb{R}	The set of real numbers
\mathbb{E}^3	Three-dimensional Euclidean space
\mathbb{S}^3	The set of all unit quaternions
$a, b, c \in \mathbb{R}^3$	Scalars
$ s , s \in \mathbb{R}$	Absolute value of a scalar
k, τ	Constants
$\mathbf{u} = [u_1, u_2, u_3]^T \in \mathbb{R}^3$	Vectors
$\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{F}$	Vectors
$\tilde{\mathbf{x}}, \tilde{\mathbf{u}}$	Homogeneous vectors
$\bar{\mathbf{x}}, \bar{\mathbf{u}}$	Augmented vectors
\mathbf{v}^T	Transpose of a vector
u_i, v_j, w_k	Vector elements
$\mathbf{u} \times \mathbf{v} \in \mathbb{R}^3$	Cross product of two 3D vectors $\mathbf{u} \times \mathbf{v} = \widehat{\mathbf{u}}\mathbf{v}$
$\mathbf{u}^T \mathbf{v} \in \mathbb{R}$	Dot product of two column vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$
$\ \mathbf{u}\ = \sqrt{\mathbf{u}^T \mathbf{u}}$	Euclidean norm of a vector
$A, B, C \in \mathbb{R}^{n \times m}$	Matrices
$\widehat{\mathbf{u}} \in \mathbb{R}^{3 \times 3}$	Skew-symmetric matrices
$\det A, A \in \mathbb{R}^{n \times n}$	Determinant of a matrix
$\text{tr } A, A \in \mathbb{R}^{n \times n}$	Trace $\text{tr } A \doteq \sum_{i=1}^n a_{ii}$ of a square matrix
a_{ij}, b_{ij}	Elements from matrix A, B , at row i and column j
$M^T \in \mathbb{R}^{m \times n}$	Transpose of a matrix $M \in \mathbb{R}^{n \times m}$
$\{\mathbf{r}\}, \{\mathbf{b}\}$	Reference frame $\{\mathbf{r}\}$ and body frame $\{\mathbf{b}\}$
$\{\hat{\mathbf{x}}_{\mathbf{r}}, \hat{\mathbf{y}}_{\mathbf{r}}, \hat{\mathbf{z}}_{\mathbf{r}}\}$	Unit axes of the coordinate system $\{\mathbf{r}\}$
T_{rb}	Transformation matrix from coordinate system $\{\mathbf{b}\}$ to $\{\mathbf{r}\}$
$\mathbf{q}_{\text{rb}} = [q_w, q_x, q_y, q_z]^T \in \mathbb{S}^3$	Unit quaternion rotating from coordinate system $\{\mathbf{b}\}$ to $\{\mathbf{r}\}$
$\mathbf{v}' = \mathbf{q} \odot \mathbf{v}, \mathbf{v} \in \mathbb{R}^3, \mathbf{q} \in \mathbb{S}^3$	Quaternion-vector multiplication, i. e., rotating a vector by the rotation induced by a unit quaternion

$\dot{\mathbf{v}}, \ddot{s}, \overset{4}{\ddot{\mathbf{x}}}, \overset{4}{\dot{y}}, \dots, \overset{n}{\dot{y}}$	Newton's notation of first, second, third, fourth, and n th time derivative
$\bar{\mathbf{x}} = \frac{1}{n}(\sum_{i=1}^n \mathbf{x}_i)$	Arithmetic mean of $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^m$

FREQUENTLY USED SYMBOLS

ω	Angular velocity
ϕ, θ, ψ	Roll, pitch, and yaw angles
$\boldsymbol{\omega} = [p, q, r]^\top$	Angular velocity vector
m	Rest mass

PHYSICAL CONSTANTS

$g = 9.806\,65 \text{ m s}^{-2}$	Standard acceleration of gravity
----------------------------------	----------------------------------

ACRONYMS

AC	alternating current	15
ANC	active noise cancellation	155
AR	augmented reality	106
AUW	all-up weight	87
BPM	beats per minute	150
DC	direct current	14
DOF	degree of freedom	18
EMI	electromagnetic interference	146
ESC	electronic speed controller	14
EWMA	exponentially weighted moving average	80
FOV	field of view	48
FPV	first-person view	14
GPS	global positioning system	58
GUI	graphical user interface	96
HDI	human-drone interaction	18
HRI	human-robot interaction	4
IMU	inertial measurement unit	15
IR	infrared	58
LED	light-emitting diode	72
LiPo	lithium polymer	15
LQ	linear quadratic	63
LTI	linear time-invariant	51
MEMS	micro-electromechanical system	146
MIDI	musical instrument digital interface	156
MPC	model-predictive control	62
NMPC	nonlinear model-predictive control	63
NUI	natural user interface	96
PCB	printed circuit board	14
PID	proportional-integral-derivative	57
PnP	perspective-n-point	73
PPM	pulse-position modulation	85
QP	quadratic programming	99

RC	radio control	11
RMSE	root-mean-square error	142
RTT	round-trip time	99
RK	Runge-Kutta	99
SAM	self-assessment manikin	122
SAR	search and rescue	109
SDK	software development kit	82
SLAM	simultaneous localization and mapping	69
slerp	spherical linear interpolation	36
SFU	semantic-free utterance	21
SQP	sequential quadratic program	64
STFT	short-time Fourier transform	93
ToF	time-of-flight	97
UAV	uncrewed aerial vehicle	8
UDP	user datagram protocol	96
UGV	uncrewed ground vehicle	109
VTOL	vertical take-off and landing	11
VR	virtual reality	106

INTRODUCTION

Through today's fast-paced world, we humans are accompanied by a vast amount of cutting-edge technology. Virtual assistants such as Alexa, Siri, Cortana and the Google Assistant seamlessly integrate into our daily routines. They raise our shutters and brief us about our individual day in the morning, read step-by-step instructions of our favorite cooking recipe at lunchtime, and switch into *do not disturb* mode at night, playing soothing sounds to aid in falling asleep. As we gain more free time to spend with our family, friends, work, or hobbies, robots with endearing pet names handle household chores, and are teased when our smartphone app alerts us to their occasional struggle on the edge of the carpet. Again.

Physically embodying virtual assistants, which are primarily on-screen synthetic social characters, results in lesser-known social robots like Jibo [Gui15] or Vector¹ (Figure 1.1). They are designed to interact with people in a natural, interpersonal fashion and find their application in education, health, and entertainment, among others. Relatively new to the consumer market are flying robots, whose natural interaction capabilities are severely limited and largely unexplored. Both technically and in terms of their versatile applications, their full potential is still far from being fully exploited. How to naturally interact with flying robots and how they could play a useful role in everyday life is subject of ongoing research. Because of the technical complexity in building accurate testbeds, many inspirations remain on a conceptual level.

1.1 ROBOTS

Developed to do the laborious work of humanity and to improve efficiency of fulfilling individual tasks, robots can be found in almost every possible domain.

¹<https://www.digitaldreamlabs.com/products/vector-robot>



Figure 1.1: Interactive companion robot *Vector*.

Industrial robots such as welding robots have become an indispensable part of the automotive industry. Shelves in large warehouses are moved by robots to the workers instead of having workers go to the shelves to pick orders. Medical and health-care robots perform surgery or provide bionic prosthetics. Military robots scout for explosive devices or assist by carrying heavy gear. Disaster response robots scan for victims, deliver essential supplies, and inspect the aftermath of emergencies. At home, consumer robots do menial tasks, like vacuum or window cleaning and mow the lawns, while robotic toy pets entertain the kids. Educational robots teach children basic design, programming and problem solving skills at an early age. Therapy for autistic children is sometimes complemented by specially designed robots. Entertainment robots “live” in theme parks or play instruments to evoke emotional responses in humans. Underwater robots dive into remote, dangerous places to collect samples or map the ocean floor.

In many meaningful scenarios, robots are employed to protect humans from dangerous situations, rescue them when in need, simplify their tasks, optimize efficiency, or even replace them completely. The interdisciplinary research field that integrates study, design and construction of robots is called *robotics*. The origins of the terms ‘robot’ and ‘robotics’ are discussed in Box 1.1.

Box 1.1: Terminology

According to Jordan [Jor16], the terms ‘robot’ and ‘robotics’ originate from the following sources. Originally, the term ‘robot’ derives from the Czech word *robota* which has its linguistic root in the Slavic word for *slave*. It was first introduced in 1920 in a science fiction play by Karel Čapek, who, however, credited his brother Josef as the true inventor of the word. The term ‘robotics’ was coined by Isaac Asimov, an American writer and professor, in a story from 1941. At the time, he assumed that, since, e. g., the science of electrical devices is called electronics, *robotics* was already an established term.

A robot is a physically embodied, incrementally artificially intelligent and autonomous machine, capable of sensing and manipulating its physical environment. Processing ability is usually given by a computer that processes the input of sensors and controls actuators, connecting them by software. Actuators provide the robot with the ability to move; this can be the entire body or just parts of it. A variety of sensors enables the robot to become aware of its environment, with cameras to see or microphones to hear being the most apparent ones for humans.

Future scenarios, where humans are likely to share their homes and workplaces with various robots, call for the development of novel and natural ways to facilitate interaction. In addition to the multitude of robot applications that we already have today, it is likely, that the number of robots will continue to increase. Demographic change and nursing shortages may require more social robots to care for the increasing number of elderly people. While a robot may not be able to replace the warmth of a human hand, it can alleviate, e. g., the workload of nurses. This allows workers to focus on the humanitarian aspects of their roles while robots handle the physical labor. As machines increasingly replace or augment human labor, it becomes even more crucial to address how humans and robots can interact effectively and appropriately.

1.2 HUMAN-ROBOT COEXISTENCE

As robots become increasingly integrated into daily life, it is essential to ensure that they can interact with humans effectively and safely. When interacting with a robot, humans naturally tend to transfer their experiences of human-human interaction to these encounters. If the robot does not react or respond according to

human expectations, this can quickly lead to frustration, turning initial acceptance into discomfort or even outright rejection of the robot.

A key challenge in designing robots and human-robot interactions is addressing the *inhibition threshold*—the instinctive psychological and behavioral point at which a person begins to feel discomfort or hesitation during interaction with a robot. This threshold can be influenced by factors such as unnatural or sudden movements, intrusion into preferred personal space, or misaligned communication behaviors. The highly interdisciplinary field of human-robot interaction (HRI) aims to address these challenges by improving the relationship and collaboration between humans and robots, creating interactions that are smoother, more natural, and more intuitive.

HRI integrates diverse domains, including computer science, psychology, sociology, mathematics, design, and various branches of physics and engineering, to enhance human-robot encounters in everyday environments. Applications span manufacturing, transportation, healthcare, education, and entertainment, where robots perform hazardous tasks in manufacturing plants, operate vehicles, assist with medical procedures, and provide companionship to the elderly. By addressing the psychological and behavioral aspects of interaction, HRI research ensures that robots are not only functional but also approachable, ultimately fostering their seamless integration into daily life.

One of the key questions in HRI is:

What are the common social mechanisms of communication and understanding that can produce efficient, enjoyable, natural and meaningful interactions between humans and robots? [SK16, p. 1936]

Additional fundamental questions that the field of HRI seeks to address, drawing upon the knowledge of the various disciplines it encompasses, include the following:

- What should a robot look like?
- How should a robot move?
- What distances foster comfortable interaction?
- How can humans and robots communicate naturally?

- How can physical interaction be enabled?
- How can a robot communicate feedback?

Ideally, interaction methods are so well designed that they are intuitive for humans without the need for instructions. The robot must understand the human's commands and the human must naturally be able to understand the robot's intent. It is the job of robot designers and HRI researchers to design social robots and provide suitable interaction modalities. This encompasses both the physical embodiment of the robots and methods of communication.

1.2.1 *Embodiment*

Robot designs and embodiments are as diverse as robot applications and mostly dependent on these. But the image of a robot that usually first appears in people's minds is the humanoid robot. Perhaps the most famous example hails from the Star Wars space opera and is named *C-3PO* (the golden half of the robot duo). A character likely more familiar to many than its real-world, functioning humanoid robot counterpart *Atlas*² by robotics company Boston Dynamics. Humanoids are designed based on the human model and therefore fit well in environments built by humans for humans. This gives them an advantage over robots of other designs, for example when an environment tailored to humans suddenly becomes hostile. They fit right in and can take care of the aftermath, e. g., by operating construction machinery that was originally designed for humans. When humanoid robots share an environment with humans, their human-inspired design has another advantage: They can leverage nonverbal channels such as gestures, posture, and body movements, which play a fundamental role in human communication. When human characteristics are imitated only in parts, without necessarily replicating the complete human form, these robots are referred to as anthropomorphic robots. If they assume animal-like forms or are endowed with animal-like traits, they are categorized as zoomorphic robots, thereby also facilitating acceptance.

When designing non-humanoid social robots, it is particularly helpful for the interaction experience and acceptance to consider their intended possibilities of interaction as early as possible in the design process [HJ14]. In the traditional

²<https://bostondynamics.com/atlas>

process, social scientists only subsequently studied existing robotic platforms for their ability to interact with humans and—if necessary—extend them in order to do so [Bar+20]. These robots, however, were usually designed by engineers solely to fulfill a single purpose as efficiently as possible. This can result in awkward, choppy, or mechanical-looking movements that are unaesthetic or unnatural and not conducive to human acceptance. No matter the chosen approach, robot design must always be balanced with the task, the user and the context. Robots are not always more compelling just because they are more human-like or playful, but also when they look and act appropriate to their task context [GKP03].

There are appearance-constrained robots that—despite their mechanical, pragmatic design—are still capable of being retrofitted with features that can increase human acceptance and interaction experience. A display that shows human facial expressions is a straightforward feature to implement on a ground-based platform. In contrast, a flying robot has a severely limited payload, which must usually be prioritized for equipment that serves the flight or to reduce its overall size in order to increase flight time. However, there are additional features that can be leveraged to enhance social interaction with robots, especially those constrained in their appearances.

1.2.2 *Communication*

In the exchange of information between humans, spoken and written language is a powerful tool that enables the communication of ideas, concepts, stories, or experiences. However, human interaction goes beyond the mere use of words used to communicate. Nonverbal communication, which encompasses every factor in human communication but the spoken and written language [Kna72], adds depth to interpersonal experiences and is crucial in the conveying of emotions. The research of nonverbal communication dates back to Darwin [Dar72]. It refers to the nonverbal cues that people use to convey meaning in addition to or instead of using language and includes: appearance, including clothing; kinesics (body movements), including gestures, facial expressions and eye gaze; proxemics (interpersonal distances); haptics (touch); vocalics (tone of voice). Nonverbal communication can convey a wide range of emotions, attitudes, and intentions, and can have a significant impact on how messages are received and interpreted by others [MHF16]. Without the supplementation of nonverbal communication,

emotions would be more challenging to convey and this information may entirely be lost in isolated language. Researching nonverbal communication plays an important role in human interaction and understanding and effectively using nonverbal cues can enhance communication and facilitate better understanding and connection between people. Nonverbal interaction enables humans to convey their inner state in a meaningful way, opening up to others, touching people, and strengthening human bonds.

The power of nonverbal means in interpersonal human communication is equally important in HRI. Thus the use of nonverbal communication has been adopted by the robotics community in order to apply the same goals to interaction with robots. Being able to transfer human nonverbal communication modalities to robots enables them to appear more natural, and thus creating more natural interaction experiences for humans. For non-anthropomorphic robots engaging with humans, it is necessary to explore and establish new communication channels, often drawing inspiration from those naturally existing between humans. Additionally, already established communication methods may have to be reevaluated to accommodate new robot designs. By introducing nonverbal communication channels, a positive perception of the interaction or the robot in general can be significantly improved [Bar+20]. Robots have demonstrated the ability to effectively convey messages through nonverbal communication alone, eliminating the need for natural language. Thus lending a robot features that humans are familiar with, is a natural and common approach in HRI. The addition of a screen already allows to show facial expressions; gestures familiar to humans, that accompany or fully substitute speech, can be mimicked and understood without necessarily requiring a (full) human form. Expressive movement can let robots appear lively and communicate their inner state, mood, intent, or even personality [HJ14]. Depending on the way a robot approaches a human, it can appear intimidating or friendly. The possibilities are many and vary greatly depending on the type of robot.

Unlike C-3PO, the other robot in the Star Wars space opera (the smaller, non-humanoid rolling trashcan) cannot communicate through natural language, and its stereotypical *beep boops* are challenging to interpret. Nevertheless, it serves as a notable example of how science fiction has influenced the conceptualization of robots and shaped human expectations of when encountering robots in real life. The historical transition from character animation to consumer robots is further explored in Box 1.2.

Box 1.2: How to breathe life into inanimate objects?

This question finds its most insightful response in the 12 principles of character animation by Disney's Nine Old Men, the core animators of The Walt Disney Company. They chronicled their seminal ideas in their book *Disney Animation: The Illusion of Life* [TJ81], published in 1981. Pixar Animation Studios, which was acquired by Disney in 2006, continued this narrative with the release of the computer-animated short film *Luxo Jr.* in 1986. It was written and directed by John Lasseter, who explored the computer animation of his Luxo desk lamp, conveying the lamps' emotions through the application of the principles of character animation. In 2007, Guy Hoffman introduced the *AUR Robotic Desk Lamp* [Hof07], a collaborative robot inspired by this short film. In 2008, Pixar released the computer-animated movie WALL-E, featuring a small garbage-collecting robot that is inadvertently left operational after mankind abandons Earth, having turned the planet into a wasteland. The idea originated during a lunch gathering of Pixar pioneers in 1994 [Ons08]. Fast forward to 2018: Anki, a company specializing in consumer robotics, introduced Cosmo, a robot companion designed by former Pixar animators [Pie16], which, along with its successor Vector, shares noteworthy resemblances with the iconic character WALL-E.

1.3 UNCREWED AERIAL VEHICLES

The majority of robots is ground-based. However, decreasing prices and miniaturization of electronic devices, as well as advances in battery development have led to robots gaining the ability to fly. This evolution marks a significant paradigm shift, expanding the scope of robotic applications and challenging traditional ways of human interaction. Uncrewed aerial vehicles (UAVs), commonly known as drones, are either autonomous or remotely piloted aircraft. Over the past twenty years, the utilization of miniature UAVs has transitioned from military applications to the enthusiast hobby sector, further extending to research and eventually reaching general consumers. By now, small UAVs have become part of everyday life and emerge in more and more professional and recreational applications. Advancing miniaturization enabled them to fly safely even indoors. Likely, this history commenced with a dream or a human aspiration, briefly summarized in Box 1.3.

Box 1.3: A Brief History of Aviation

Inspired by birds, the dream of flying is as old as mankind. The earliest historical records date back to one of the most famous and oldest Greek legends, that tells the story of Daedalus and Icarus. In order to escape the Minotaur from the labyrinth on Crete, Daedalus makes wings for his son Icarus and himself out of bird feathers and wax. When Icarus flies too close to the sun, the wax melts. Plummeting into the depths, many pioneers in the history of aviation shared Icarus' fate. Yet, thanks to them, the dream of flying became a reality more than two centuries ago, and today's life would be hard to imagine without it.

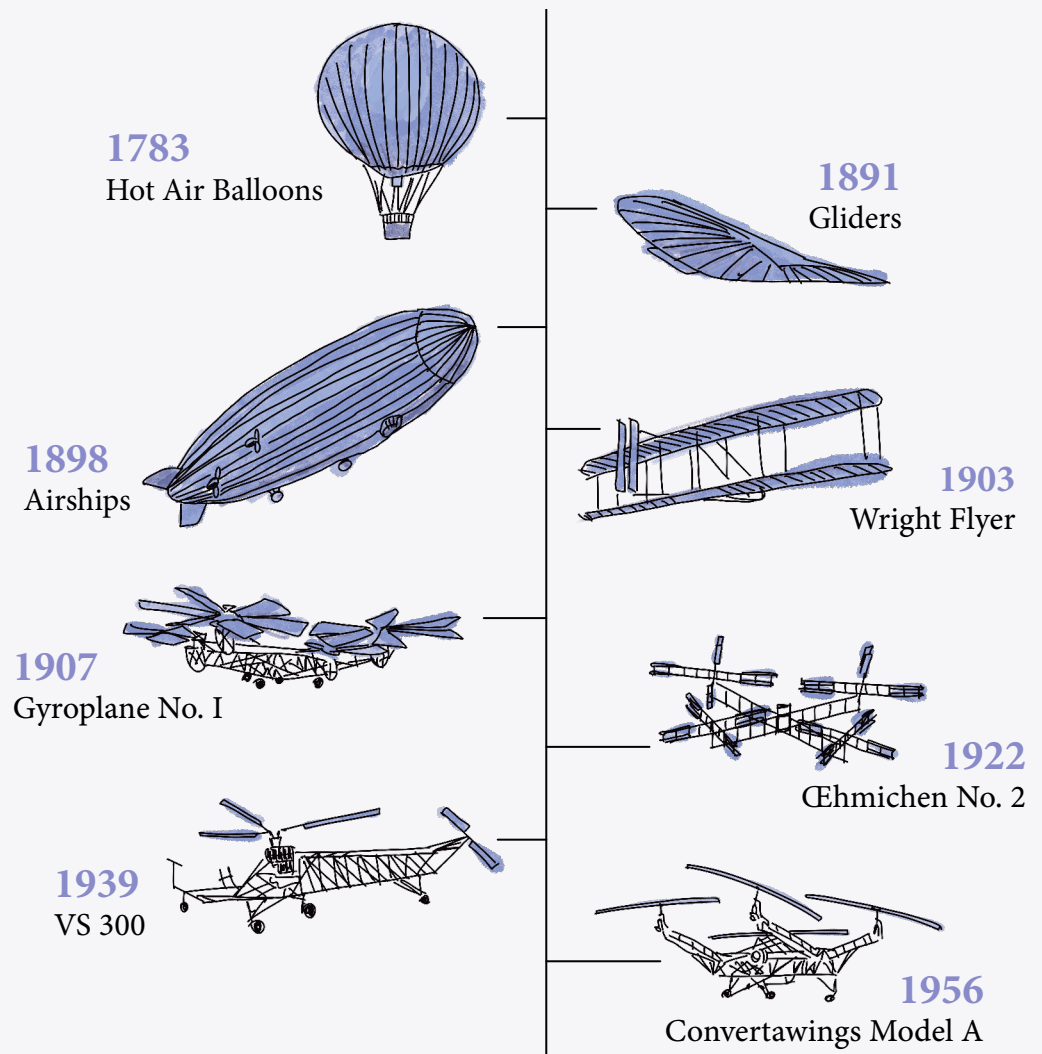


Figure 1.2: A brief history of aviation.

From unmanned balloons, kites, and toys like the Bamboo-Copter, invented several centuries before the common era, to DaVinci's conceptional drawings from the 15th century, the first manned flights started in the late 18th century. According to *Flight: The Complete History of Aviation* [Gra17], hot air balloons (1783) were followed by gliders (1891), airships with combustion engines (1898), the first controlled powered flight of the Wright brothers' airplane (1903), and finally, the prototype of the first mass-produced helicopter (1939).

The first manned heavier-than-air flight that lifted vertically took place in 1907. The Gyroplane No. I was a four-rotor helicopter, where two rotors turned clockwise and the other two rotors turned counterclockwise, effectively counteracting torque. Its structural design can be considered the first instance of what is known today as a *quadrotor*. However, these pioneering flights were tethered and neither controllable or steerable. In late 1922, with the Cehmichen No. 2, the first reliable vertical take-off and landing (VTOL) aircraft took off to complete over 1000 flights. Its design is also a multirotor helicopter with four main rotors that have adjustable blade angles, referred to as *pitch*. In addition, it used vertically mounted rotors for stabilization, steering and propulsion. Their arrangement countered torque introduced by the main rotors. With the invention of the swashplate, a device that translates control inputs of the pilot to the main rotor, engineers focused on the development of a helicopter that became the traditional single-rotor configuration known today. In this design, a vertical tail rotor counters the torque that is introduced by the main rotor and is used to control the rotation of the helicopter about its vertical axis. Vertical movement is achieved by adjusting the collective pitch of the main rotor while movement along the longitudinal and transverse axes is obtained by adjusting the cyclic pitch. Due to the complexity of the swashplate, the quadrotor design has been reconsidered by engineers. In 1956, the first "true" quadrotor, the Convertawings Model A, successfully demonstrated forward flight by solely adjusting the thrust of the individual rotors, eliminating the necessity for cyclic pitch control.

Other rotorcraft designs have been developed throughout the following years. Another design that solves the problem of yawing introduced by a single main rotor is the coaxial rotor helicopter. It uses a pair of rotors with the same axis of rotation but turning in opposite directions and thus canceling out each other's torques. Another common multirotor helicopter design is the tandem rotor helicopter, where two rotors are mounted one behind the other. Next to cancelling out the yawing motion introduced by a single main rotor, multiple rotors also allow for

increased payloads. Convertiplanes combine the speed and range of fixed-wing aircraft with the VTOL capability of rotorcraft. In a tiltwing design, rotors are attached to the wing, and after vertical take-off and gaining speed, the entire wing and rotor assembly progressively tilts forward. Tiltrotor helicopters, like the Bell Boeing V-22 Osprey, have rotors attached to a rotating shaft mounted at the end of a fixed wing. This concept has been applied to modern drones, enabling them to combine VTOL capabilities with long-range flight.

Vertical take-off and landing (VTOL) aircraft possess several advantages over fixed-wing aircraft, primarily stemming from their ability to perform vertical takeoff and landing—eliminating the need for a runway. This feature enables effective operation even in confined spaces, such as those typically found in urban areas. The capability to hover, maintaining a stationary mid-air position, is another key advantage, facilitating tasks such as observation and surveillance. Additionally, VTOL aircraft offer enhanced maneuverability, allowing them to rotate during hover and navigate effectively in constrained or cluttered spaces, which is particularly useful in search and rescue scenarios. However, despite these advantages, VTOL aircraft come with their own set of challenges and trade-offs. These include typically higher fuel or battery consumption during vertical takeoff and landing, and a generally limited range due to less efficient lateral flight.

1.3.1 *Modern Drones*

The design of the four-rotor helicopter experienced a revival with the beginning of the 21st century. While not every human can fly like a bird, advancements in radio control (RC) technology, briefly outlined in Box 1.4, and the ongoing process of miniaturization now allow individuals to experience a bird's eye view through modern UAVs, with quadrotors being one of their simplest forms.

Box 1.4: A Brief History of Remote-Controlled Aircraft

In 1898, Nicola Tesla demonstrated a small vessel whose propeller, rudder, and onboard gadgets could be controlled without any visible connection; the first wirelessly remote-controlled vehicle. When a science writer discussed its use as a remote-controlled torpedo, Tesla clarified that “you do not see there a wireless

torpedo; you see there the first of a race of robots, mechanical men which will do the laborious work of the human race” [ONe44]. His patent *Method of and Apparatus for Controlling Mechanism of Moving Vessels or Vehicles* [Tes98] was granted in the same year.

Forty years later, in 1938, the first successful radio-controlled airplane was built by the Good brothers [Goo96]. The first helicopters appeared in 1960 and the first commercial radio-controlled model helicopter was presented by Dieter Schlüter on a toy fair in Germany in 1972 [Sch07]. In 1999, the Draganflyer [Inc21], the first commercialized quadrotor UAV, captured the interest of researchers and the quadrotor design evolved to become the standard platform for aerial robotics.

In early 2021, the small coaxial helicopter “Ingenuity” completed its test flight on Mars and became humanity’s first controlled powered flight on another planet. Both its rotors have swashplates allowing for collective and cyclic control [Bal+]. It carried a piece of fabric of the wing of the Wright brothers’ airplane [Wal21].

Over the past two decades, drones have become increasingly popular and commonplace due to their simplicity, stability, and affordability. Their practical applications are manifold. With their ability to hover and different levels of autonomy, they have already been integrated into numerous everyday applications in both professional and private life.

Drones are increasingly being used in *search and rescue* missions [WT10; NSB11; Sam+12], primarily because of their ability to quickly access remote locations or places that are difficult for humans to reach. Carrying cameras, they can be used to assess the damage after natural disasters such as earthquakes, floods, and hurricanes, providing valuable information to aid in rescue and recovery efforts in *disaster management* [Res+15; Dau+22]. Swarms of quadrotors are quick to deploy and can create a three-dimensional map of a local area of interest in no time [SC17]. This way hazardous terrain like collapsed or contaminated buildings, or crime scenes can be mapped before the arrival of rescue services or police forces without endangering humans. Incorporating this information on the way to a mission can save lives by directly targeting already located disaster victims. In 2021, the first fully automated commercial drone flights were approved by the Federal Aviation Administration (FAA) and drones are now being successfully utilized for *package delivery* [Fra19], particularly in areas where traditional delivery methods are difficult, costly, or time-critical. Drones are used for *security*

and surveillance purposes, monitoring public events, critical infrastructure, and border security. Companies already offer out-of-the-box solutions for fully autonomous surveillance of industrial facilities. Without any human involvement at all, multirotors follow trajectories and return to their base stations in order to recharge batteries. In the field of *environmental monitoring* drones can be used to monitor wildlife [Cor+21] and track the movement of migratory animals [Saq+18]. They are further utilized to assess the health of forests [Zha+16] and other natural habitats, or to collect environmental DNA for biodiversity monitoring [Auc+23]. In *precision agriculture*, drones are used for crop monitoring, yield analysis, and precision spraying, reducing the need for manual labor and improving crop management [MD18]. Drones equipped with visual, thermal, and infrared detection technologies can be used to protect wildlife in grain fields during harvests or in the fight against poaching in South Africa [Jon22]. In *aerial photography and videography*, drones that are equipped with high-definition cameras are increasingly being used for capturing stunning aerial shots for movies, advertisements, real estate, and events. As flying selfie video cameras, they accompany athletes as well as regular people pursuing their endeavors. Drones reduce the need for manual inspections and improve safety in *infrastructure inspection*, where they are used to inspect wind turbines [Shi+19], bridges [SDW18], high voltage power lines [Ive+21], oil pipelines [Alh+20], and other critical infrastructure [Noo+21]. For *aerial surveying and mapping*, drones provide accurate and up-to-date information for construction, e. g., by estimation of construction site elevations [JB20], urban planning, and other industries. In *personal transportation*, electric two-seaters are currently transitioning from experimentation to reality [Len18].

1.3.2 Design Principle

The modern quadrotor follows the design of the *Convertawings Model A* four-rotor helicopter, which was introduced in Box 1.3, and achieves motion not by adjusting the pitch of the rotors, but by adjusting the speed of the individual motors. This design results in a particularly simple to control flight system. Among commonly used rotor configurations illustrated in Figure 1.3, quadrotors represent the simplest specific design.

Rotor configurations with fewer than four rotors necessitate additional servo motors to tilt the rotors in order to achieve full motion in three dimensions. While

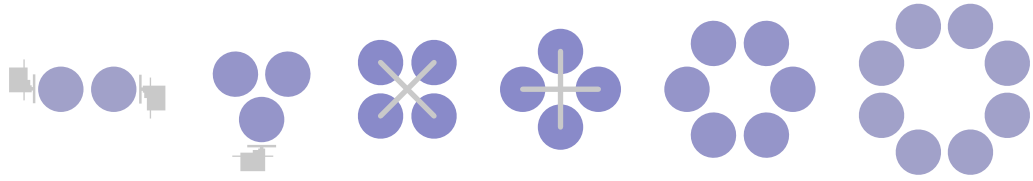


Figure 1.3: Rotor configurations with two, three, four, six, and eight rotors, usually referred to as bicopter, tricopter, quadcopter, hexacopter, and octocopter, respectively. The quadcopter (or quadrotor) at the center is depicted in both \times and $+$ configurations. It has the minimum number of fixed rotors required to achieve full motion in three dimensions. Configurations with fewer rotors require servo motors to tilt one or more of the rotor planes.

more complex to control, servo motors also have a very low update frequency, which renders these configurations unsuitable for agile maneuvers. Rotor designs with more than four rotors are usually used to carry heavier payloads. The redundancy of the rotors also ensures fail-safety so that valuable payloads can be emergency landed. With four or more rotor axes, it is also common to use coaxial configurations in order to increase the drone's payload.

A basic quadrotor system, like the first-person view (FPV) hobby drone shown in Figure 1.4, consists of four, fixed-pitch rotors driven by electric motors, a receiver that receives radio signals, a flight controller printed circuit board (PCB) that translates these signals into independent motor commands, and a usually cross-shaped frame.

In RC applications, there are mainly two types of electric motors: brushed direct current (DC) motors and brushless DC motors. Both use a coil assembly, that changes current in order to cause the rotor (rotational part) to “chase” the unlike pole of a fixed magnetic field attached to the stator (stationary part). In a *brushed* DC motor, the coils are attached to the rotor within a fixed magnetic field, that consists of two permanent magnets with opposite polarities attached to the stator. Power is supplied to the coils by conductive brushes, that make contact with a rotating commutator, which continually flips the direction of the current into the coils, causing their polarity to reverse. The speed of brushed DC motors is proportional to the applied voltage. In a *brushless* DC motor, coils are affixed to the stator, and permanent magnets are attached to the rotor, eliminating the need for brushes. To control rotation, the current fed to the coils is adjusted in both magnitude and direction. This requires electronic speed controllers (ESCs) for

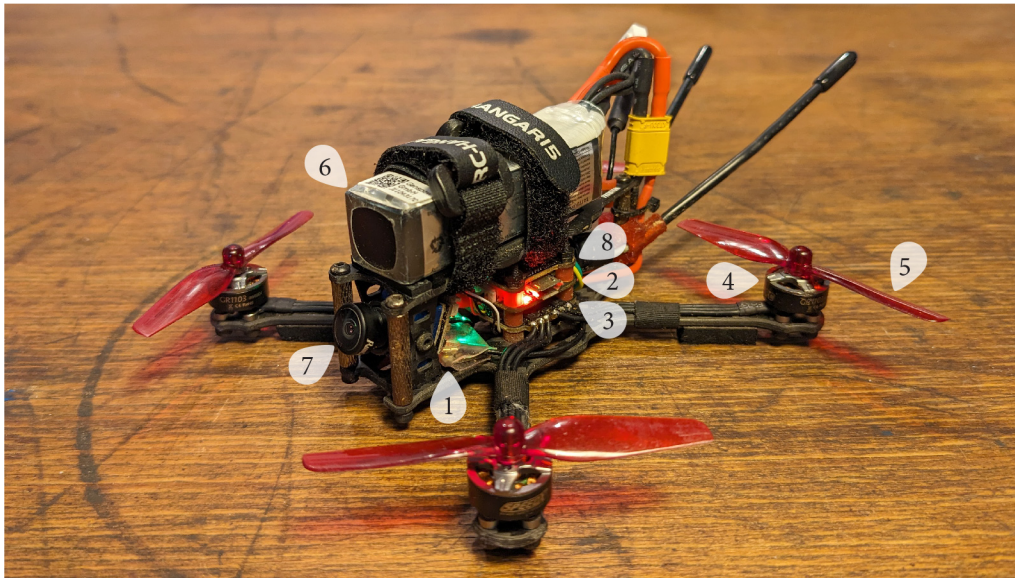


Figure 1.4: This photograph of a hobby quadrotor showcases its key components, namely a receiver (1), printed circuit boards for the flight controller (2) and electronic speed controllers (3), brushless motors (4), 65 mm 2-blade rotors (5), and a lithium polymer battery (6). The camera (7) and video transmitter board (8) are specific to the FPV hobby, providing a live feed to the operator, allowing them to navigate the drone in real-time through a first-person view. All components are mounted to a carbon frame in \times configuration with a wheelbase of 125 mm. This drone has an all-up weight of 104 g.

each motor, which convert the battery voltage into high-resolution three-phase alternating current (AC) power. ESCs can be placed altogether on a dedicated PCB (as seen on the FPV drone in the photograph), or integrated with a receiver and flight controller on a single PCB, a common feature in miniature hobby drones. The speed of both the brushed and the brushless motor linearly depends on the voltage of the lithium polymer (LiPo) battery. For this reason, the rating of motors is given by the kV constant, e. g., 8000 kV, that is the number of rotations per minute per every volt applied.

An indispensable sensor for stabilization and control of a drone is the inertial measurement unit (IMU). Typically situated on the flight controller PCB, in very critical situations, it may be placed externally to isolate from motor vibrations. An IMU combines several instruments—usually accelerometers, gyroscopes and magnetometers—to measure the force, angular velocity, and orientation of a body in three dimensions.

A quadrotor typically has four control inputs, namely, *thrust*, *roll*, *pitch*, and *yaw*. Thrust, or sometimes throttle, controls vertical motion by adjusting the overall rotor speeds. By addressing individual rotor speeds, roll, pitch, and yaw inputs control the angular speed of a quadrotor about its three principal axes shown in Figure 1.5. Their etymology is nautical, and they refer to the longitudinal, transverse, and normal axes, respectively, of a ship, aircraft, or spacecraft.

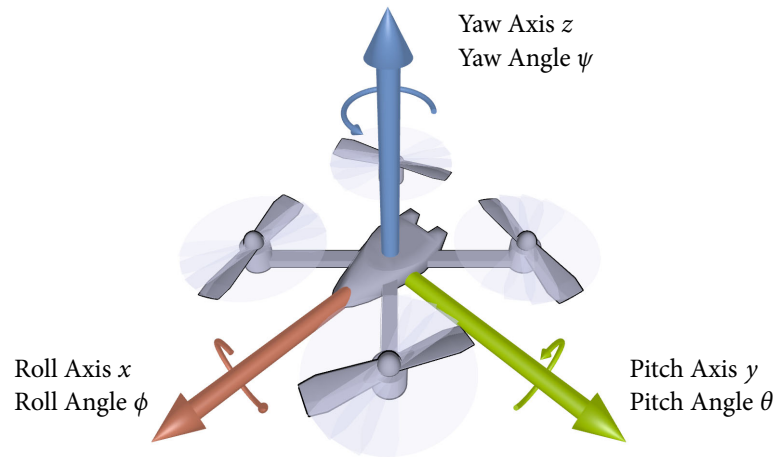


Figure 1.5: Principal axes of a quadrotor in \times configuration as used in this thesis. The x axis points in direction of flight, the y axis points to the left, and the z axis points upward.

The aerodynamic lift force generated by the rotors along the vertical axis causes the quadrotor to hover, as the lift generated exceeds gravity. The vertically oriented fixed-pitch rotors are usually placed equidistantly on the frame shaping either a \times or a $+$. In a $+$ configuration, rotor pairs are in alignment with the roll and pitch axes, while in the \times configuration, they are offset by 45° . Opposing pairs of rotors turn in the same direction. One pair rotates clockwise, while the other pair rotates counterclockwise. Thus, the torques generated by the individual rotors cancel each other out and do not affect the orientation of the quadrotor. Control is achieved by adjusting the speed of the individual motors, which affects the torques they produce, and consequently, the thrust forces generated by the attached rotors. To initiate yaw motion, the speed of the rotors spinning in one direction is increased and the speed of the rotors rotating in the opposite direction is decreased. This results in a net torque, causing angular acceleration about the yaw axis. The quadrotor achieves horizontal movement by tilting in the desired direction. This is accomplished by decreasing the speed of the rotors on one side and increasing the speed of

the rotors on the opposite side. The relationship between each control input and rotor speed for both configurations can be seen in Figure 1.6. The × configuration

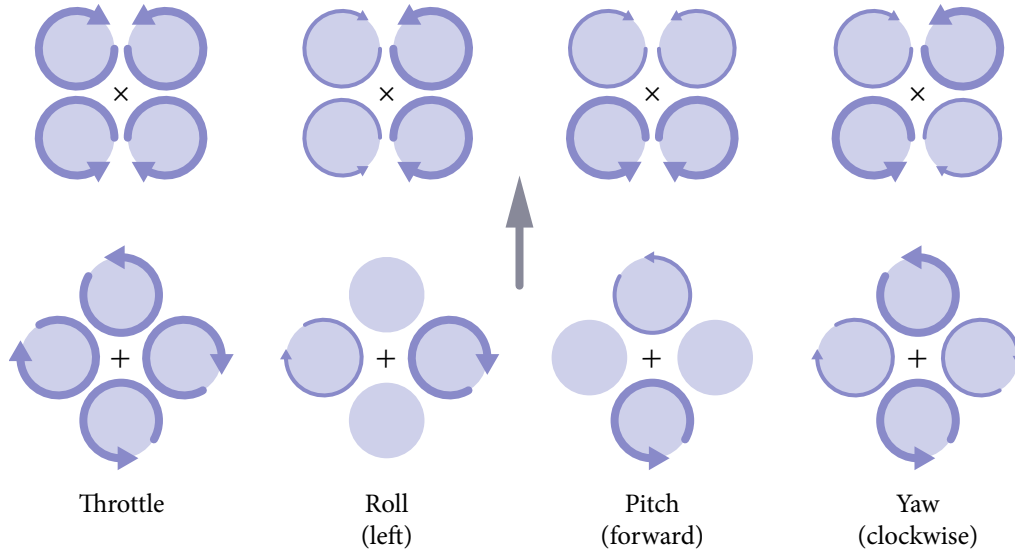


Figure 1.6: Changes in rotor speed to achieve the desired control inputs throttle, roll, pitch, and yaw, for × and + configurations. The arrow is pointing into the direction of flight.

of a quadrotor is typically chosen because it provides a clear forward view for a mounted camera. Additionally, a × configuration offers some physical advantages, described later in Section 2.3.

The size of multirotors is usually defined by the diameter of an imaginary circle on which the rotors are placed, shown in Figure 1.7. It is referred to as *wheelbase* and

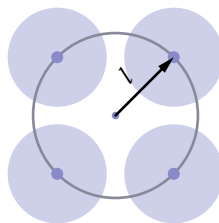


Figure 1.7: A commonly used size specification for multirotors is their wheelbase, an imaginary circle with diameter $2l$ through its rotor centers, where l is the center-to-rotor-distance.

gives an idea of the physical dimensions of a rotor configuration.

The motion of a quadrotor in three-dimensional space can be described as a rigid body, which has six degrees of freedom (DOFs): three translational along the roll, pitch, and yaw axes, and three rotational about those axes. A system's DOFs are the minimum number of independent parameters that define its state. However, the six DOFs of a quadrotor are not independently controllable, as it only has four inputs. This makes the quadrotor an *underactuated* system because only four actuators control the six DOFs. Four DOFs (vertical movement and rotations about the three axes) are independent while two DOFs (translation in the two-dimensional, horizontal plane) are coupled to the rotations about the roll and pitch axes. Rotating the quadrotor as a whole tilts the rotor axes and, consequently, the direction in which the rotors generate thrust. The corresponding thrust component acting horizontally accelerates the vehicle in the desired direction. To stop horizontal motion, the quadrotor needs to be tilted in the opposite direction.

1.4 HUMAN-DRONE INTERACTION

HRI and the study of effects that robots have on humans are still in an early phase [SK16] and the field of nonverbal human-drone interaction (HDI) is still emerging. Quadrotors have made their way to the consumer area just about 10 years ago, when they were still quite big (0.5–1.0 m). Only in recent years they have become so small, autonomous and affordable that they can fly indoors without safety concerns. As the presence of flying robots continues to grow in both the commercial and consumer sectors, there is a need to understand appropriate methods for nonverbal interaction with humans. Increasing autonomy, which makes previously required remote controls more and more obsolete, demands new ways of interaction as well. Usually some intermediary devices are required in order to enable HDI: motion capture systems to track robot motion or the human's body, cameras or depth sensors to track the user's skeleton or to determine facial expression, IMUs to determine human hand attitude, or microphones to capture voice commands.

In 2017, DJI introduced the Spark shown in Figure 1.8, a quadrotor (0.17 m) that can be deployed from a person's hand and controlled, e. g., by gestures detected in the video of the onboard camera. The first consumer quadrotor that could autonomously navigate through unknown terrain to pursuit a human target in order to capture their endeavor was launched in early 2018 by Skydio. The second

version, Skydio 2, was released in late 2019. Many insights from the field of HRI



Figure 1.8: The DJI Spark is a consumer drone with a 175 mm wheelbase, which can be deployed from the operator's hand and controlled using gestures.

have already been incorporated into these drones, now available on the market. But due to their aerial nature, established concepts from ground-based robots are not always directly transferable to flying robots. Since consumer drone technology is still so young, the specialized subtopic of HRI is only slowly taking shape and numerous possibilities for interaction remain unexplored. Kinesics and proxemics stand out as the most prominent research topics in the field. There is limited research on haptic interaction with drones, and vocalics have remained largely untouched so far, leaving untapped potential to enhance interaction.

1.4.1 *Kinesics*

Motion communication with parts of the (human) body or the body as a whole is summarized under the term *kinesics*. Drone kinesics studies how emotions or intent can be conveyed through the way a robot moves [Cau+16; Den+18; LSB21a; BD21], which also includes aerial drone gestures, e. g., to be used for visual pedestrian guidance [Col+17]. Several gesture control approaches have been suggested for effective communication over longer distances in a natural fashion. For example,

to command a quadrotor to *follow* a person, *take a picture*, or to *land* [NSC13], or combinations of gestures and voice commands to let a quadrotor *fly higher*, *fly to a precise location*, *stop motion*, *follow*, or to *take a selfie* [Cau+15].

In situations where cameras cannot capture the complete human body, e. g., because they are too close, human pose estimation and thus gesture control may fail. In these cases, interaction methods in even closer proximity may be desired. Therefore researchers have examined facial expression [Nag+14] and voice [NS11; Kri+15] for controlling drones. Among other approaches, these have been extended to multimodal interaction, e. g., including hand gestures [Cac+16; Suá+16]. The majority of these contactless scenarios require the use of higher level sensors such as depth cameras or motion sensors. These are usually permanently installed in controlled indoor testbeds or mounted to the drone's frame. Under certain conditions, it is also possible to attach the sensors to the human body [Cal+16; Gro+19].

1.4.2 Proxemics

The research field of *proxemics*, i. e., the study of interpersonal distances, investigates how individuals utilize space during social interactions [Hal66]. Originating from disciplines such as psychology, sociology, and architecture, proxemics has expanded to encompass HRI. This area of study explores how spatial relationships influence communication, social relationships, and perceptions of identity and status. Proxemics also addresses the arrangement of objects within an environment, helping designers create spaces that foster effective communication, positive interactions, and cultural sensitivity. In the HRI domain, proxemics has been applied to investigate how robots, including drones, can approach humans in ways that respect their comfort zones and facilitate interaction.

Specifically, drone proxemics focuses on understanding the appropriate distances and approaches for drones to initiate interactions with humans while ensuring they remain within a range that feels comfortable and non-threatening. This research is critical for enabling intuitive and socially acceptable interactions between humans and autonomous systems.

1.4.3 *Haptics*

Various forms of nonverbal communication through the sense of touch, occurring between humans, between animals, or between humans and animals, are included under the term *haptic* interaction. Handshakes, high fives and patting someone's shoulder are examples of haptic communication or touch interactions. They play an important role in human interaction, as these hand gestures convey a wide range of social and emotional messages, from formal greetings to expressing joy, offering comfort, and providing support. Touch is essential in building and maintaining social bonds and is the earliest sense to develop in the fetus [CWE99]. This is why it is considered one of the most intuitive and instinctive ways of interaction.

Within close proximity, the area of drone haptics studies direct interaction through touch. Such scenarios enable direct interaction with flying interfaces and spatial movement of programmable matter [Gom+16]. Drone haptics further include direct control with physical buttons that have been attached to the frame of a quadrotor [RBS17] enabling close-range interaction. In an outdoor physical interaction setting, a human operator works with a drone to transport a payload collaboratively using a custom-built sensor system [PV23]. Drone haptics even include the landing of drones on the human body [Aud+21].

1.4.4 *Vocalics*

The study of *vocalics*, a subset of paralanguage, is part of the broader field of paralinguistics. Vocalics refer to nonphonemic properties of speech and are consciously or subconsciously utilized to support or contradict messages, which can be verbal, kinesic, or proxemic, among others. It includes nonverbal voice qualities, voice modifiers, and utterances, as well as momentary silences. Paralinguistic features include timbre, resonance, loudness, tempo, pitch, intonation, syllabic duration, and rhythm [Poy93]. Semantic-free utterances (SFUs) are another form of paralanguage. They can be categorized under four general types: gibberish speech, non-linguistic utterances, musical utterances, and paralinguistic utterances [Yil+16]. SFUs are employed in children's games [Zag+16] and are an integral part of animated movie characters such as WALL-E, allowing them to communicate affect and intent. Proven on the screen, SFUs also offer great potential in the field of robotics, where inspired researchers improved robot communication

through artificial movement sounds [RVB21]. It also motivated efforts to generalize sound design recommendations [RBV22] based on interviews with the designers of robotic consumer products like the Anki Vector (its predecessor was designed by former Pixar animators [Pie16]), which uses non-lexical (i. e., without words) audio for communication [PBK20]. While research within the HRI community on vocalics has increased in recent years [ZF23; Pel+21], it has hardly found its way into the field of HDI. Published works include the acoustic navigation of visually impaired individuals, where the drone’s rotor sound serves as an auditory cue for orientation [AF18], and the incorporation of natural sounds through a loudspeaker [Wan+23].

1.5 QUADROTOR TESTBEDS

In general, quite sophisticated testbeds are required in order to enable institutes to utilize UAVs for their specific research. They typically consist of multiple quadrotor platforms and a pose estimation system that sends position and attitude estimates of the vehicles to a ground station running control algorithms and piloting the quadrotors via a radio control system. Regardless of weather conditions, lighting or the time of day, controlled indoor test environments provide safe and reproducible flights. Safety hazards for both humans and machines can be avoided by introducing appropriate safety measures and utilizing simulation environments before testing algorithms on real platforms. State-of-the-art indoor testbeds, such as the General Robotics, Automation, Sensing, and Perception testbed [Mic+10] at the University of Pennsylvania, the Flying Machine Arena [Lup+14] at ETH Zurich, and the Robotics and Perception Group’s testbed [FFS18] at the University of Zurich, are equipped with advanced but expensive technology. Commonly used professional quadrotors settle in the lower four-figure US dollar range. However, progress in the development of motors, batteries, and flight controllers, driven primarily by the hobbyist sector—especially companies involved in FPV—has enabled a considerable reduction of costs of individual components over the past decade. Today, a platform built from hobby parts is about an order of magnitude cheaper to purchase than it was ten years ago. There are also off-the-shelf development platforms available at similar prices, like the Bitcraze Crazyflie³. The significant costs of a UAV testbed are the widely used optical motion capture systems, settling

³<https://www.bitcraze.io/products/crazyflie-2-1/>

in the six-figure US dollar range. With these high expenses, groups of smaller institutes in particular face financial obstacles entering the exciting field of UAVs.

Indoor testbeds are also crucial for user studies in the field of HDI, given their capability for accurate and reproducible flights. However, due to technical maintenance and high costs, it is common to conduct user tests as a Wizard of Oz experiment, where subjects believe they are interacting with an autonomous system when, in fact, it is controlled by a human. One of the reasons for this is often that a proper system was not developed yet or is too expensive and complex to maintain, like a drone testbed equipped with technology enabling precise flight. Regrettably, both inexpensive off-the-shelf drones and human operators face challenges in achieving consistent trajectory flights, resulting in non-repeatable outcomes that are, however, crucial for reliable research findings.

1.6 RESEARCH OBJECTIVES

By contributing to multiple peer-reviewed conferences on HRI, this thesis examines the new field of HDI using computer-aided methods for the development of solutions to highly topical practical problems. More specifically, the goal of this thesis is to evaluate suitable nonverbal interaction mechanisms that can be implemented by onboard means. In pursuit of this objective, the core questions of HRI, introduced earlier, will be elaborated and addressed, focusing on the drone's motion, appropriate distances to humans, natural interaction methods, and possibilities for the drone to respond. To enable the practical exploration of these questions, the design and implementation of a drone testbed are first presented. Within the scope of this dissertation, five conference papers were produced, which are listed in Appendix B.

A suitable testbed is developed and enables further research in the field of aerial robots. The quadrotor testbed *ICARUS* [Lie+17] realizes a scalable low-cost testbed for UAV research, especially for HDI. It combines hardware and software for tracking and control of multiple miniature off-the-shelf quadrotors without the need of elaborate safety measures or detailed knowledge about the quadrotor platforms.

In HRI, a natural inhibition threshold exists as humans often struggle to interpret the intentions of robots, especially when robots like quadrotors cannot be endowed

with a human-like appearance. One possibility to overcome this challenge is by animating their movements. This thesis evaluates established concepts from animated movie creators for their suitability in adapting to quadrotors [LSB21a]. Additionally, a method to anticipate drone movement is implemented and assessed through a user study.

To further evaluate the human inhibition threshold, a user study examines the proximity to which participants allow the approach of a plain functional, i. e., visually unaltered, quadrotor [LSB21b]. The study is conducted in a seated setting and simulates future workplace scenarios, involving the drone approaching from four directions. Additionally, the participants' inclination to engage in physical interaction is assessed by providing them with the option to stop the drone from approaching by tapping their frame.

This thesis further contributes with a novel approach of physical close proximity HDI by employing “virtual buttons” whose operation is registered using onboard sensors [LSB21c]. This interaction method can thus be used without the need for additional hardware. As a proof of concept, a playful HDI scenario is created with the intention of improving the bond between humans and robots through shared musical experiences.

In order to further enhance HDI, part of this thesis is to implement acoustic information into quadrotor trajectories, which ultimately enables a novel possibility of communication. The minimalist approach of using onboard means is extended to the use of rotor noise to provide additional acoustic information to quadrotor flights. A user study shows that this information helps to better distinguish between otherwise similar-sounding aerial gestures [LS24].

1.7 THESIS OUTLINE

In Chapter 2, the background theory necessary to construct and operate a controlled quadrotor testbed is described. This includes rigid-body motion, projective geometry for pose estimation, and the dynamic model of a quadrotor with the required theory in dynamical systems and classical mechanics. Additionally, it covers how to create and control quadrotor trajectories.

Chapter 3 delves into technical aspects of the quadrotor testbed developed as an integral part and foundation of this thesis. It outlines its components including pose estimation and the drones employed, and provides insights into the implementation. It further encompasses examples of applications for which the testbed has been employed up to the present.

The main contribution is described in Chapter 4, where appropriate forms of nonverbal communication for a pragmatic quadrotor platform are examined and implemented. Section 4.1 investigates how quadrotor movements and trajectories can effectively convey intent, with findings supported by a user study. Drone proxemics, including tolerated distances in HDI, are explored through a user study presented in Section 4.2. In Section 4.3, a novel method for physical close-proximity interaction with quadrotors is proposed and evaluated. Finally, Section 4.4 assesses the feasibility of producing distinguishable sounds using a quadrotor's rotors in midflight, with the results validated through a user study.

The findings are discussed in Chapter 5, which includes a comparative analysis of the results and implications for the field of HDI. Concept limitations and implementation issues are discussed, and potential future directions are proposed. Chapter 6 summarizes the key findings and provides a recapitulation of the conducted research.

2

BACKGROUND THEORY

To enable control of robots and conducting user studies in controlled environments, several fields of research come together to form an interdisciplinary scientific field. Control of robots requires accurate and time-critical knowledge about their current state. It includes three-dimensional position, linear velocity and linear acceleration, which are related to the translational part, as well as attitude, angular velocity, and angular acceleration, that are related to the rotational part. There are various approaches and a variety of sensors available, in order to determine a robot's current state. Its position and orientation, for example, can be determined by optical pose estimation using cameras. This method can be implemented either from the robot's point of view, which is called inside-out tracking, or from a third person's perspective, e. g., a mounted camera in a controlled environment, called outside-in tracking. With a mathematical representation of translational and rotational motion, that is captured, e. g., by cameras, and a mathematical model of the robot, it can be driven from its current state to a desired state using methods of control theory. An ordered sequence of multiple desired states separated by time intervals composes a trajectory, or flight path.

The disciplines described above include multiple branches of mathematics, physics and engineering. Movement and impact of robots on humans is influenced by psychology, sociology, the robot's trajectory, and interaction design. To build a quadrotor test environment, suitable hardware is required, which is tied together with the above disciplines by computer science. The purpose of this chapter is to provide the fundamental prerequisites for understanding the chapters that follow. It starts with the description of rigid-body motion based on spatial transformations in Section 2.1. Then, a short introduction to projective geometry and the mathematical camera model is given in Section 2.2. In Section 2.3, the dynamical model of the quadrotor and trajectory generation is described. How quadrotors are controlled is subject of Section 2.4.

2.1 RIGID-BODY MOTION

In the field of robotics, the precise description of a robot's configuration is crucial for achieving accurate control, effective navigation, and communication. The minimum number of continuous coordinates required to specify a robot's configuration are the robot's DOFs. A robot's *configuration space* is the n -dimensional space containing all possible configurations; the robot's configuration is one point in this configuration space. A quadrotor flying through three-dimensional physical space has six DOFs: three coordinates specifying its position and three angles defining its orientation. These parameters are specified relative to a fixed coordinate frame of reference, as illustrated in Figure 2.1.

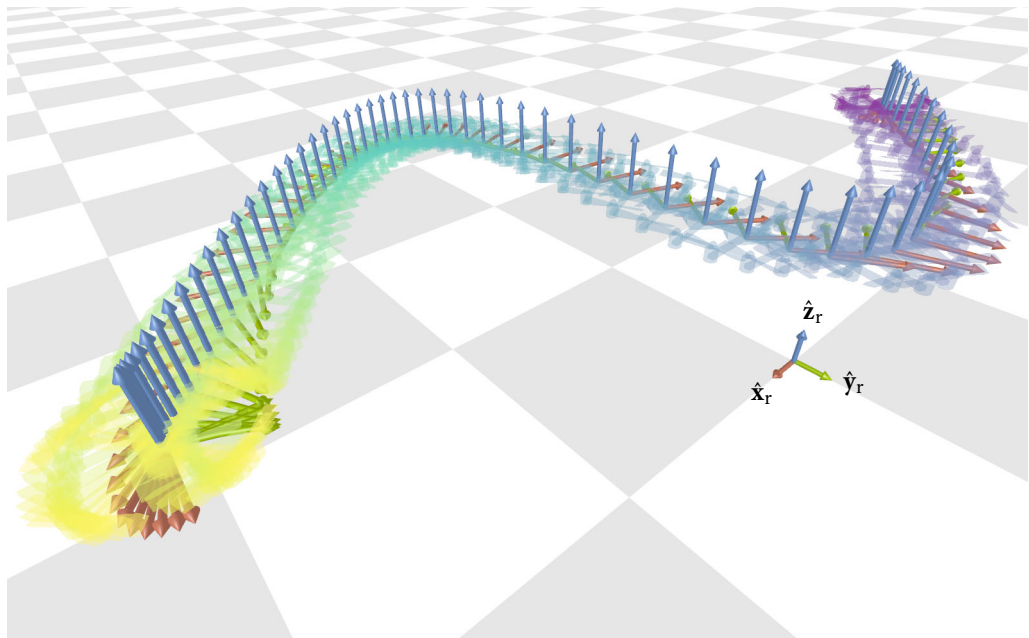


Figure 2.1: Rigid-body motion of a simulated quadrotor relative to a fixed reference frame $\{r\}$. The temporal progression of trajectories is visualized using the viridis color map [WS16], with the **start** in purple and the **end** in yellow.

As a rigid body moves, its pose changes over time but as the body is rigid, distances between points on the body remain constant. Thus it is sufficient to describe its state by a single point on the object and a coordinate frame attached to that point, the body frame $\{b\}$, relative to a fixed reference frame $\{r\}$. The set of all coordinates representing the position of a rigid body and all possible attitudes of the body-fixed

coordinate frame at that point is the configuration space of rigid-body motions, described in the following sections.

2.1.1 Coordinate Frames

The mathematical model of the descriptive space we are familiar with is called *Euclidean space*. Points in Euclidean space can be represented by numerical values, enabling mathematical computations and various operations, such as measuring distances, calculating angles, and performing transformations. A single point in three-dimensional Euclidean space \mathbb{E}^3 can uniquely be specified by three-dimensional real-valued *Cartesian coordinates* \mathbb{R}^3 . They describe the distances from the *origin* of a Cartesian coordinate system along its three orthogonal unit axes \hat{x} , \hat{y} , and \hat{z} . Using a Cartesian coordinate system to represent Euclidean space allows to translate geometric problems into algebraic problems and vice versa.

A visual representation of Cartesian coordinate frames is given in Figure 2.2, which shows a reference coordinate frame $\{\mathbf{r}\}$ and a rigid body-attached coordinate frame $\{\mathbf{b}\}$. The position and orientation of a rigid body in physical space can be described by a *translational* part, that is a vector from the origin of the reference frame to the origin of the body frame, and a *rotational* part, that is the relative orientation of the body frame to the fixed world coordinate frame.

With established coordinate frames, the origin of the body frame $\{\mathbf{b}\}$ can be described relative to the reference frame $\{\mathbf{r}\}$. This translation can be represented by a three-dimensional column-vector

$$\mathbf{v} \doteq [v_1, v_2, v_3]^\top = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \in \mathbb{R}^3$$

A vector $\mathbf{v} \in \mathbb{R}^3$ in three-dimensional Euclidean space is defined as a directed arrow pointing from point \mathbf{p} to point \mathbf{q} :

$$\mathbf{v} \doteq \mathbf{q} - \mathbf{p} \in \mathbb{R}^3.$$

A vector has both magnitude (length) and direction, but it is not necessarily rooted anywhere. It can represent various physical quantities like displacement between points, velocity, or force. A velocity vector's magnitude for example, represents the

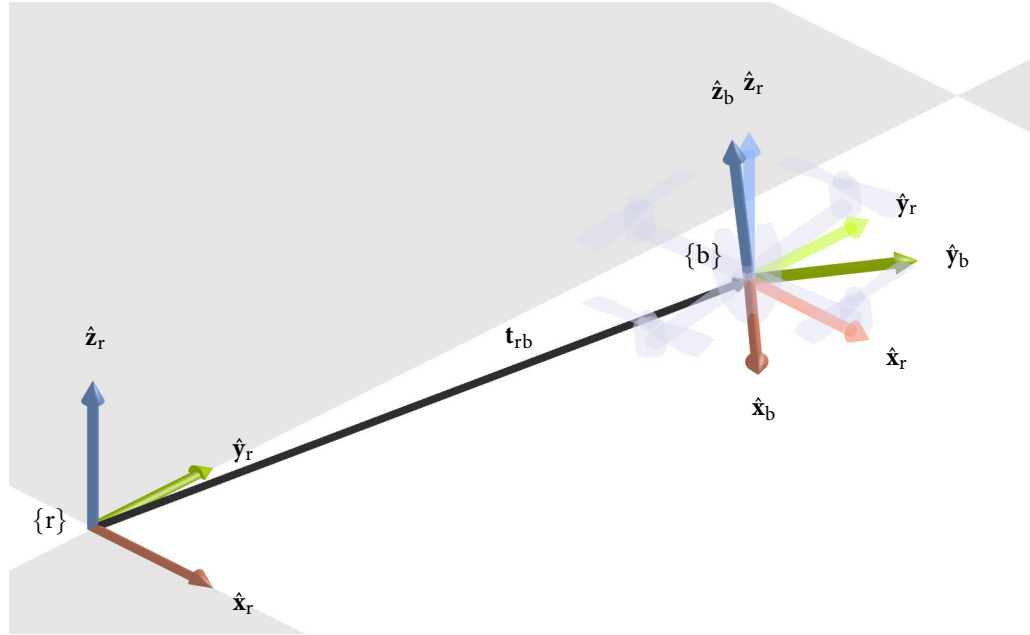


Figure 2.2: A coordinate frame is graphically represented with three arrows depicting the orthogonal unit vectors that represent the principal axes of the coordinate system. A body-fixed coordinate frame $\{b\}$ can be described relative to a world reference frame $\{r\}$. Its origin is displaced by the vector \mathbf{t}_{rb} illustrated with a black arrow. Its attitude differs by a rotation that maps the principal axes $\{\hat{x}_r, \hat{y}_r, \hat{z}_r\}$ to $\{\hat{x}_b, \hat{y}_b, \hat{z}_b\}$.

speed of the object and the direction of the vector is the direction of the object's velocity. Bound to a coordinate frame, a point in space can also be represented by a vector.

The *metric* which allows to measure angles and distances in Euclidean space \mathbb{E}^3 using Cartesian coordinates \mathbb{R}^3 , is induced by the definition of the *dot product*. The dot product or scalar product on the vector space \mathbb{R}^3 is defined as

$$\mathbf{u}^\top \mathbf{v} \doteq u_1 v_1 + u_2 v_2 + u_3 v_3, \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^3.$$

In its geometric definition, the dot product of two vectors is related to the cosine of the angle θ between them:

$$\mathbf{u}^\top \mathbf{v} \doteq \|\mathbf{u}\| \|\mathbf{v}\| \cos \theta.$$

Two vectors are orthogonal (perpendicular) when their dot product is zero. Furthermore, the dot product can be used to compare the directions of vectors through

the so-called *cosine similarity*. Additionally, it can be used to determine the *scalar projection* of one vector onto another vector. The Euclidean norm $\|\cdot\|$ is defined by

$$\|\mathbf{v}\| \doteq \sqrt{\mathbf{v}^\top \mathbf{v}} = \sqrt{v_1^2 + v_2^2 + v_3^2}, \quad \forall \mathbf{v} \in \mathbb{R}^3.$$

and measures the length of a vector \mathbf{v} , e. g., to determine the distance between two points in Euclidean space.

The *cross product* $\mathbf{u} \times \mathbf{v}$ of two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$ results in a third vector with coordinates

$$\mathbf{u} \times \mathbf{v} \doteq \begin{bmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{bmatrix} \in \mathbb{R}^3. \quad (2.1)$$

The cross product is perpendicular to each of its factors and thus perpendicular to the plane spanned by \mathbf{u} and \mathbf{v} , following the *right-hand rule* illustrated in Figure 2.3.

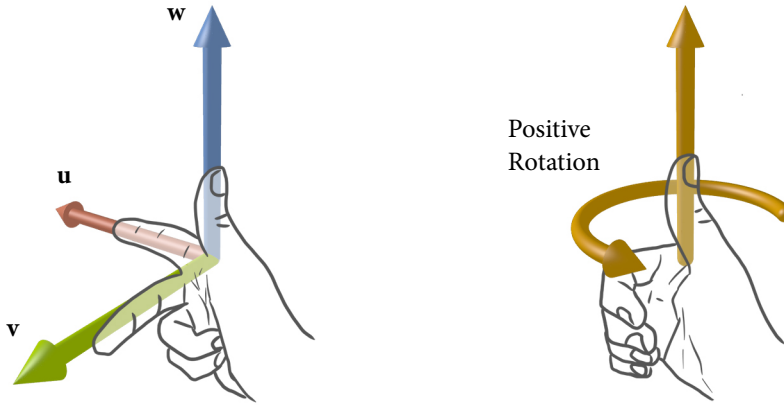


Figure 2.3: Right-hand rule: If the index finger of the right hand points along vector \mathbf{u} and the middle finger points along vector \mathbf{v} , then the thumb points in the direction of $\mathbf{u} \times \mathbf{v} = \mathbf{w}$ (left). When the thumb of the right hand points in the direction of a rotation axis, the fingers curl in the direction of positive rotation (right).

With fixed \mathbf{u} , the cross product (2.1) can be represented by a map from \mathbb{R}^3 to \mathbb{R}^3 : $\mathbf{v} \mapsto \mathbf{u} \times \mathbf{v}$. Since this map is linear in \mathbf{v} , it can be represented by a matrix $\widehat{\mathbf{u}}$

$$\mathbf{u} \times \mathbf{v} = \underbrace{\begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}}_{\widehat{\mathbf{u}}} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \widehat{\mathbf{u}} \mathbf{v}.$$

Thus the cross product defines a map from a vector \mathbf{u} to a *skew-symmetric* matrix $\widehat{\mathbf{u}}$, simply by extracting, e. g., vector $[u_1, u_2, u_3]^\top$ from $\widehat{\mathbf{u}}$. A matrix $\widehat{\mathbf{u}}$ is skew-symmetric when

$$\widehat{\mathbf{u}} = -\widehat{\mathbf{u}}^\top. \quad (2.2)$$

The set of 3×3 skew-symmetric matrices is called $so(3)$.

The vector space \mathbb{R}^3 and the space of all skew-symmetric 3×3 matrices are *isomorphic*, i. e., there exists a one-to-one map that preserves the vector space structure. The isomorphism is given by the *hat operator*

$$\wedge : \mathbb{R}^3 \rightarrow so(3), \quad \mathbf{u} \mapsto \widehat{\mathbf{u}}. \quad (2.3)$$

Its inverse map, the *vee operator*, extracts components of the skew-symmetric matrix $\widehat{\mathbf{u}}$ and is given by

$$\vee : so(3) \rightarrow \mathbb{R}^3, \quad \widehat{\mathbf{u}} \mapsto \mathbf{u}.$$

2.1.2 Representation of Rotation

To fully describe a rigid body in three-dimensional space, a rotation is required in addition to translation. To achieve this, a coordinate frame can be attached to the rigid body, as illustrated in Figure 2.2.

Rotation Matrices

Writing the principal axes of the body-attached frame $\{\mathbf{b}\}$ in terms of an arbitrary coordinate frame $\{\mathbf{r}\}$ is one possible way to represent the body's attitude.

If desired or necessary due to context, vectors can be subscripted with the coordinate frame they live in, e. g., the principal axes of the coordinate frame $\{\mathbf{b}\}$ are denoted as $\hat{\mathbf{x}}_{\mathbf{b}}$, $\hat{\mathbf{y}}_{\mathbf{b}}$, and $\hat{\mathbf{z}}_{\mathbf{b}}$. Stacking them as column-vectors results in a matrix

$$R_{\mathbf{rb}} = \begin{bmatrix} \hat{\mathbf{x}}_{\mathbf{b}} & \hat{\mathbf{y}}_{\mathbf{b}} & \hat{\mathbf{z}}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3}$$

which describes the orientation of $\{\mathbf{b}\}$ in terms of $\{\mathbf{r}\}$. The column vectors in $R_{\mathbf{rb}}$ must constitute an orthonormal frame, adhering to both the unit norm condition

(i. e., they must be unit vectors) and the orthogonality condition (i. e., the dot products of the column vectors must be zero). These conditions can be expressed as a single set of constraints on R ,

$$R^T R = R R^T = I, \quad (2.4)$$

which means that R is an orthogonal matrix and implies that its inverse equals its transpose:

$$R^{-1} = R^T. \quad (2.5)$$

Thus the *column vectors* of R_{rb} represent the orthogonal unit basis vectors of the body frame $\{b\}$ resolved in frame $\{r\}$, and, consequently, the *row vectors* of R_{rb} represent the orthogonal unit basis vectors of frame $\{r\}$ resolved in the body frame $\{b\}$. To ensure that R forms a right-handed coordinate frame, its determinant must be one

$$\det R = 1. \quad (2.6)$$

The set of matrices, that fulfill the six equality constraints contained in (2.4) as well as the additional constraint to ensure right-handedness (2.6) is the special orthogonal group $SO(3)$ of rigid rotations about the origin in three-dimensional Euclidean space \mathbb{R}^3 .

Euler Angles

Euler angles are a descriptive and thus commonly utilized representation of rotation in aerospace engineering and robotics. They decompose a rotation into a sequence of three elemental rotations around fixed axes shown in Figure 2.4.

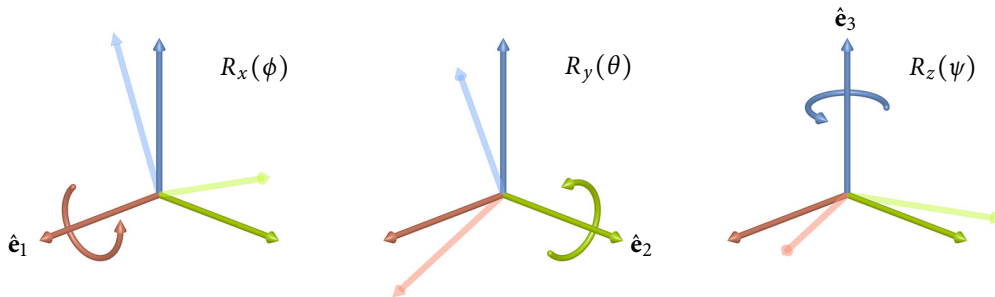


Figure 2.4: Principal rotations $R_x(\phi)$, $R_y(\theta)$, and $R_z(\psi)$ utilizing Euler angles $\phi = \theta = \psi = \frac{\pi}{8}$.

Euler angles are commonly denoted as α, β, γ , or ϕ, θ, ψ . The rotation matrix about the x axis, $\hat{\mathbf{e}}_1 = [1, 0, 0]^\top$, is given by

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \in SO(3),$$

about the y axis, $\hat{\mathbf{e}}_2 = [0, 1, 0]^\top$, by

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \in SO(3),$$

and about the z axis, $\hat{\mathbf{e}}_3 = [0, 0, 1]^\top$, by

$$R_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \in SO(3).$$

Because rotations, in general, do not commute, the sequence in which elemental rotations are applied is critical. One commonly used convention is the zyx convention, where the frame undergoes a yaw angle rotation ψ about the z body axis, followed by a pitch angle rotation θ about the new y body axis, and finally, a roll angle rotation ϕ about the new x body axis:

$$R = R_z(\psi)R_y(\theta)R_x(\phi).$$

In total, there are 12 possible combinations of applying Euler angles: $zxx, xyx, yzy, zyz, xzx, yxy, xyz, yzx, zxy, xzy, zyx$, and yxz . When two of the three rotation axes align, the rotation becomes singular and loses one DOF. This singularity is commonly known as *gimbal lock*. The non-uniqueness and singularities of Euler angles introduce limitations and drawbacks when used as a parametrization for rigid-body motion.

Exponential Coordinates

A minimal, three-parameter representation of elements in $SO(3)$ is given by exponential coordinates, where a unit vector $\hat{\mathbf{w}}$ parametrizes the rotation axis and θ

the rotation angle about that axis. The matrix exponential $e^{\widehat{\omega}\theta}$, which is also often denoted by $\exp(\widehat{\omega}\theta)$, satisfies both properties of $SO(3)$. From the definition of the exponential matrix $(e^{\widehat{\omega}t})^{-1} = e^{-\widehat{\omega}t} = e^{\widehat{\omega}^\top t} = (e^{\widehat{\omega}t})^\top$, it follows that

$$(e^{\widehat{\omega}\theta})^\top e^{\widehat{\omega}\theta} = I.$$

Since for any square matrix $A \in \mathbb{R}^{3 \times 3}$ holds $\det(e^A) = e^{\text{tr}(A)}$, it follows that

$$\det(e^{\widehat{\omega}t}) = e^0 = 1.$$

A physical interpretation of the matrix exponential is that if $\|\boldsymbol{\omega}\| = 1$, then $R(\widehat{\boldsymbol{\omega}}, \theta) = e^{\widehat{\boldsymbol{\omega}}\theta}$ simply describes a rotation by an angle of θ radians about the axis $\boldsymbol{\omega} \in \mathbb{R}^3$.

The *axis-angle* representation $\theta\boldsymbol{\omega}$ is often used to parametrize a rotation in three-dimensional Euclidean space. In general, θ can be absorbed into $\boldsymbol{\omega}$, resulting in

$$R = e^{\widehat{\boldsymbol{\omega}}}$$

for $\boldsymbol{\omega}$ with norm θ . Thus the matrix exponential defines a map from $so(3)$ to $SO(3)$, the so-called *exponential map*

$$\exp : so(3) \rightarrow SO(3); \quad \widehat{\boldsymbol{\omega}} \mapsto e^{\widehat{\boldsymbol{\omega}}}.$$

Determining the rotation matrix R from a given $\boldsymbol{\omega} \in \mathbb{R}^3$ without computing the full matrix exponential series can effectively be done using *Rodrigues' formula*:

$$e^{\widehat{\boldsymbol{\omega}}t} = I + \widehat{\boldsymbol{\omega}} \sin(t) + \widehat{\boldsymbol{\omega}}^2(1 - \cos(t)).$$

Adding multiples of 2π to the norm of the exponential coordinate results in the same rotation. If $\|\boldsymbol{\omega}\| = 1$, $t = 2k\pi$, it follows

$$e^{\widehat{\boldsymbol{\omega}}2k\pi} = I, \quad \forall k \in \mathbb{Z}.$$

Thus, there are infinitely many exponential coordinates $\boldsymbol{\omega} \in \mathbb{R}^3$ such that $e^{\widehat{\boldsymbol{\omega}}} = R$. The exponential map is therefore not a one-to-one map. It is also not commutative, meaning that for two $\widehat{\boldsymbol{\omega}}_1, \widehat{\boldsymbol{\omega}}_2 \in so(3)$,

$$e^{\widehat{\boldsymbol{\omega}}_1} e^{\widehat{\boldsymbol{\omega}}_2} \neq e^{\widehat{\boldsymbol{\omega}}_2} e^{\widehat{\boldsymbol{\omega}}_1} \neq e^{\widehat{\boldsymbol{\omega}}_1 + \widehat{\boldsymbol{\omega}}_2},$$

unless $\widehat{\boldsymbol{\omega}}_1 \widehat{\boldsymbol{\omega}}_2 = \widehat{\boldsymbol{\omega}}_2 \widehat{\boldsymbol{\omega}}_1$.

To demonstrate, that for any $R \in SO(3)$, there exists a (not necessary unique) $\boldsymbol{\omega} \in \mathbb{R}^3$, such that $R = \exp(\widehat{\boldsymbol{\omega}})$, the inverse of the exponential map is denoted by

$$\widehat{\boldsymbol{\omega}} = \log(R).$$

It can be shown by construction (see, e. g., [Ma+03; LP17]), that for any rotation matrix $R \in SO(3)$, $R \neq I$, the corresponding exponential coordinates $\boldsymbol{\omega}$ are given by

$$\|\boldsymbol{\omega}\| = \cos^{-1}\left(\frac{\text{tr}(R) - 1}{2}\right), \quad \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} = \frac{1}{2 \sin(\|\boldsymbol{\omega}\|)} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \quad (2.7)$$

If $R = I$, it follows that $\|\boldsymbol{\omega}\| = 0$, and $\boldsymbol{\omega}/\|\boldsymbol{\omega}\|$ can not be determined.

The three-dimensional exponential coordinates provide a minimal and unconstrained parametrization for rotations in three-dimensional Euclidean space. Along with a translation vector, they are a suitable representation of a six DOFs pose in rigid-body motion¹.

Unit Quaternions

With small rotation angles, exponential coordinates can be numerically sensitive [LP17] because of the division by $\sin \theta$ in the logarithm formula (2.7). Another representation of rotation that mitigates this issue are unit quaternions, though at the cost of an additional variable required for representation. Quaternions can be written as a 4-vector $\mathbf{q} = [q_w, q_x, q_y, q_z]^T$. Unit quaternions, i. e., quaternions of unit-length $\|\mathbf{q}\| = 1$ form a three-dimensional unit sphere \mathbb{S}^3 , commonly known as the 3-sphere, embedded in four-dimensional space \mathbb{R}^4 . They are double covering the space of 3D rotations $SO(3)$, meaning that *antipodal* quaternions, i. e., quaternions of opposite signs \mathbf{q} , $-\mathbf{q}$, represent the same rotation. Quaternion algebra is often simpler than the mathematical operations required for exponential coordinates. Also unit quaternions are continuous, i. e., they do not suffer from gimbal lock singularities as rotation matrices do. They are often utilized because of their ease of pose to pose interpolation and thus popular in computer graphics; spherical linear interpolation (slerp) [Sho85] follows the straightest and shortest path between two quaternions.

¹A similar coordinatization as the exponential coordinates for a rotation matrix $R \in SO(3)$ exists for full rigid-body motion, called *twists*. Twists are useful for pose optimization and tracking, but exceed the scope of this thesis.

Unit quaternions extend complex numbers and are composed of a scalar (real) part $q_w \in \mathbb{R}$ and a vector (imaginary) part $[q_x, q_y, q_z]^\top \in \mathbb{R}^3$. The unit quaternion representation of a rotation $R \in SO(3)$ with axis-angle representation $\hat{\omega}\theta$ is constructed as

$$\mathbf{q} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} w \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \cos \frac{\theta}{2} \\ \sin \frac{\theta}{2} \hat{\omega} \end{bmatrix} \in \mathbb{R}^4.$$

Using the cosine double-angle formula $\cos 2\phi = 2 \cos^2 \phi - 1$ and considering that the matrix trace tr of a rotation matrix R equals the sum of its eigenvalues², leading to $1 + 2 \cos \theta = \text{tr } R$ [Hug04; LP17], a unit quaternion can be derived from a rotation matrix R by

$$\mathbf{q}(R) = \frac{1}{2} \begin{bmatrix} \sqrt{1 + r_{11} + r_{22} + r_{33}} \\ (r_{32} - r_{23}) / \sqrt{1 + r_{11} + r_{22} + r_{33}} \\ (r_{13} - r_{31}) / \sqrt{1 + r_{11} + r_{22} + r_{33}} \\ (r_{21} - r_{12}) / \sqrt{1 + r_{11} + r_{22} + r_{33}} \end{bmatrix}.$$

Conversely, a linear transformation can be obtained [Shu93; Mar08] that rotates a vector about the rotation induced by the unit quaternion \mathbf{q} about the unit axis $[q_x, q_y, q_z]$ and an angle $2 \cos^{-1} q_w$:

$$R(\mathbf{q}) = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_w q_y + q_x q_z) \\ 2(q_w q_z + q_x q_y) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_w q_x + q_y q_z) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \in SO(3).$$

In this thesis, quaternion-vector multiplication is denoted by the \odot operator:

$$\mathbf{v}' = \mathbf{q} \odot \mathbf{v} = R(\mathbf{q}) \mathbf{v}.$$

Composing two rotations \mathbf{q}_1 and \mathbf{q}_2 is done by quaternion multiplication, which is defined as

$$\mathbf{q}_3 = \mathbf{q}_1 \mathbf{q}_2 = \begin{bmatrix} w_1 w_2 - \mathbf{v}_1^\top \mathbf{v}_2 \\ \mathbf{v}_1 \times \mathbf{v}_2 + w_1 \mathbf{v}_2 + w_2 \mathbf{v}_1 \end{bmatrix},$$

with the property $R(\mathbf{q}_3) = R(\mathbf{q}_1)R(\mathbf{q}_2)$. Quaternion multiplication is not generally commutative, just as rotation using rotation matrices. The inverse of a quaternion

²An eigenvector is a nonzero vector which only changes by a scalar factor, its eigenvalue, when a linear transformation is applied [FH15].

\mathbf{q}^{-1} is calculated by flipping the sign of either w or \mathbf{v} and has the same meaning as $R^{-1} \in SO(3)$. The inverse of a quaternion is useful to determine the incremental rotation between two quaternions, defined as the quaternion division

$$\mathbf{q}_3 = \mathbf{q}_1/\mathbf{q}_2 = \mathbf{q}_1\mathbf{q}_2^{-1} = \begin{bmatrix} -w_1w_2 - \mathbf{v}_1^\top\mathbf{v}_2 \\ \mathbf{v}_1 \times \mathbf{v}_2 + w_1\mathbf{v}_2 - w_2\mathbf{v}_1 \end{bmatrix}.$$

An algorithm that averages quaternions was introduced by Markley et al. [Mar+07]: The quaternions \mathbf{q}_i are averaged with weights w_i by determining the 4×4 matrix

$$M = \sum_{i=1}^n w_i \mathbf{q}_i \mathbf{q}_i^\top.$$

The weighted average quaternion is then the normalized eigenvector corresponding to the maximum eigenvalue of M

$$\bar{\mathbf{q}} = \arg \max_{\mathbf{q} \in \mathbb{S}^3} \mathbf{q}^\top M \mathbf{q}.$$

2.1.3 Rotational Motion

When a body with coordinate axes $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\}$ is under continuous motion, its change in orientation at times t and $t + \Delta t$ can be described by a rotation angle $\Delta\theta$ about a unit axis $\hat{\boldsymbol{\omega}}$ that passes through the coordinate frame's origin. As Δt becomes infinitesimally small, the limit of $\Delta\theta/\Delta t$ becomes the rate of rotation $\dot{\theta}$ and $\hat{\boldsymbol{\omega}}$ the instantaneous axis of rotation. Combined, $\dot{\theta}$ and $\hat{\boldsymbol{\omega}}$ define the *angular velocity*

$$\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} \dot{\theta},$$

shown in Figure 2.5. The angular velocity is a pseudovector (or axial vector, i. e., it behaves like a vector but the direction does not conform under rigid transformations) normal to the instantaneous plane of angular displacement. The orientation of the angular velocity follows the right-hand rule that was shown in Figure 2.3. The magnitude of the angular velocity vector represents angular speed.

Assume the orientation of the coordinate system in Figure 2.5 is described by rotation matrix $R(t)$ at time t with reference to the fixed frame $\{r\}$ and $\dot{R}(t)$ is the body's time rate of change in orientation. Then the column vectors of $R(t)$

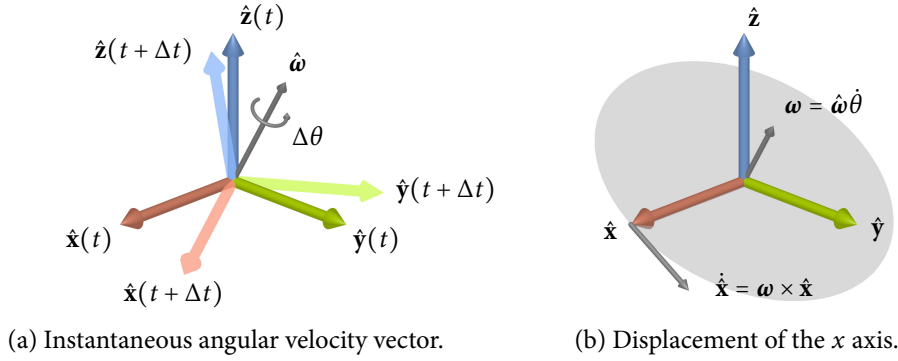


Figure 2.5: The angular velocity can be represented by a pseudovector $\hat{\omega}$ whose axis is the instantaneous axis of rotation and its length is the rotation angle (a). It is normal to the instantaneous plane of angular displacement (b).

describe the \hat{x} , \hat{y} , and \hat{z} axes in fixed-frame coordinates. When the angular velocity ω_r at a specific time t is expressed in fixed-frame coordinates, by extending the example given for the x axis displacement in Figure 2.5b to the remaining axes, the body's time rate of change in orientation results in

$$\dot{R} = \begin{bmatrix} \omega_r \times \hat{x} & \omega_r \times \hat{y} & \omega_r \times \hat{z} \end{bmatrix}.$$

Using the vee map (2.3), the cross products can be rewritten with skew-symmetric notation to obtain

$$\dot{R} = \widehat{\omega}_r R, \quad \widehat{\omega}_r \in so(3). \quad (2.8)$$

At the identity matrix I , a skew-symmetric matrix gives a first-order approximation to a rotation matrix [Ma+03]. In fact, the set of 3×3 skew-symmetric matrices $so(3)$ consists of all possible \dot{R} at the identity, i. e., $R = I$. It is also called the tangent space at the identity of the rotation group $SO(3)$. If R is not at the identity, it is simply transported to the identity by matrix multiplication on the right $\dot{R} = \widehat{\omega}R$. Locally, elements of $SO(3)$ only depend on three parameters, $[\omega_1, \omega_2, \omega_3]$.

Rewriting (2.8) results in the angular velocity vectors

$$\widehat{\omega}_r = \dot{R}R^{-1}, \quad (2.9)$$

$$\widehat{\omega}_b = R^{-1}\dot{R}, \quad (2.10)$$

with reference to frames $\{r\}$ and $\{b\}$, respectively [LP17]. The orientation of the rotating frame as seen from the fixed frame R and \dot{R} are related to angular velocity

by (2.9) and (2.10). Conceptually, $\widehat{\omega}_b$ is not the angular velocity relative to a moving frame, but the angular velocity relative to a stationary, inertial frame $\{b\}$, that instantaneously has the same position and orientation as the frame attached to the moving frame. This eliminates effects due to non-inertial moving frames attached to rotating bodies and simplifies equations. Furthermore, $\widehat{\omega}_r$ is independent of the choice of the body frame as well as $\widehat{\omega}_b$ is independent of the choice of the reference frame since the products on the right side of (2.9) and (2.10) are independent of $\{b\}$ and $\{r\}$, respectively.

The time derivative of a quaternion $\mathbf{q} = [q_w, q_x, q_y, q_z]^\top$ with angular velocity $\boldsymbol{\omega} = [p, q, r]^\top$ is given by

$$\dot{\mathbf{q}} = \frac{1}{2} \Omega(\boldsymbol{\omega}) \mathbf{q}, \quad (2.11)$$

with

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}.$$

Assuming the body rates are constant over time, the orientation can be estimated using the zeroth-order quaternion integrator [TR05]

$$\mathbf{q}_{rb}(t + \Delta t) = \left(\cos\left(\frac{\|\boldsymbol{\omega}\|}{2} \Delta t\right) \cdot I_{4 \times 4} + \frac{1}{\|\boldsymbol{\omega}\|} \cdot \sin\left(\frac{\|\boldsymbol{\omega}\|}{2} \Delta t\right) \cdot \Omega(\boldsymbol{\omega}) \right) \cdot \mathbf{q}_{rb}(t) \quad (2.12)$$

with the identity matrix $I_{4 \times 4} \in \mathbb{R}^{4 \times 4}$.

2.1.4 Coordinate System Transformations

Changing descriptions from frame to frame is done by mathematical mapping. If two frames have the same orientation, i. e., they only differ by their location, mapping of a point \mathbf{p}_b with respect to $\{b\}$ to a point \mathbf{p}_r relative to frame $\{r\}$ is done by vector addition

$$\mathbf{p}_r = \mathbf{p}_b + \mathbf{t}_{rb},$$

where \mathbf{t}_{rb} is the vector pointing from the origin of $\{r\}$ to the origin of $\{b\}$. This operation does not change the point itself but changes its description, namely from frame $\{r\}$ to $\{b\}$.

The matrix R_{rb} describes the orientation of the body frame $\{b\}$ relative to the reference frame $\{r\}$, it transforms the coordinates \mathbf{v}_b of a point in Euclidean space described relative to frame $\{b\}$ to coordinates \mathbf{v}_r relative to the reference frame $\{r\}$ by matrix multiplication:

$$\mathbf{v}_r = R_{rb}\mathbf{v}_b.$$

From (2.4) and (2.5), the properties of $SO(3)$, it follows that any two frames $\{d\}$ and $\{e\}$ hold

$$R_{de} = R_{ed}^{-1} = R_{ed}^T.$$

If a rotation matrix R_{ab} represents the orientation of $\{b\}$ in $\{a\}$ and another rotation matrix R_{bc} represents the orientation of $\{c\}$ in $\{b\}$, then the representation of $\{c\}$ in $\{a\}$ can be computed as

$$R_{ac} = R_{ab}R_{bc}$$

The notation using subscripts helps to keep track of mappings and frames of reference. For matrices, the second subscript of the first matrix and the first subscript of the second matrix “cancel” each other out, e. g.,

$$R_{ac} = R_{a\cancel{b}}R_{\cancel{b}c} = R_{ab}R_{bc}.$$

This rule also works when changing the reference frame of a vector:

$$\mathbf{v}_a = R_{a\cancel{b}}\mathbf{v}_{\cancel{b}} = R_{ab}\mathbf{v}_b.$$

Combining both translation and rotation leads to

$$\mathbf{x}_a = R_{ab}\mathbf{x}_b + \mathbf{t}_{ab}. \quad (2.13)$$

In general, rigid-body transformations are denoted by

$$g : \mathbb{R}^3 \rightarrow \mathbb{R}^3; \quad g(\mathbf{x}) \doteq R\mathbf{x} + \mathbf{t}, \quad R \in SO(3), \mathbf{t} \in \mathbb{R}^3.$$

The six DOFs given by (R, \mathbf{t}) describe the position and orientation of a rigid body relative to a fixed frame.

A *rigid-body motion* is the transformation or map from a geometric set of points that represents the rigid body in Euclidean space to another subset of Euclidean space

preserving distances and orientations. A map preserving distances (or the norm) is called Euclidean transformation and is denoted by $E(3)$. Euclidean transformations or maps, however, also include reflections, which are physically impossible for rigid bodies. In order to exclude such maps, rigid-body transformations need to also preserve orientations, i. e., the handedness of the coordinate frame. This means that next to preserving the dot product, also the cross product of any two vectors has to be preserved. Transformations satisfying these properties are called special Euclidean transformations and are denoted by $SE(3)$. A right-handed coordinate frame with its principal axes being three orthonormal (orthogonal unit) vectors still forms a right-handed coordinate frame after a transformation by an element in $SE(3)$.

2.1.5 Homogeneous Coordinates

In contrast to the pure rotational case, the coordinate transformation for full rigid-body motion is not linear but *affine*, i. e., a combination of linear transformations such as rotations, and translations. By utilizing *homogeneous coordinates*, both affine and, more generally, projective transformations can be converted to linear transformations and represented using matrices.

A two-dimensional, inhomogeneous vector \mathbf{x} is converted to a homogeneous vector $\tilde{\mathbf{x}}$ by appending a 1 to its coordinates:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \tilde{\mathbf{x}} \in \mathbb{R}^3,$$

with augmented vector $\tilde{\mathbf{x}}$. This three-dimensional extension of a two-dimensional vector is illustrated in Figure 2.6. Three-dimensional points are extended analogously to the two-dimensional case:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \tilde{\mathbf{x}} \in \mathbb{R}^4. \quad (2.14)$$

This extension of coordinates embeds the Euclidean space \mathbb{E}^3 into a hyperplane in \mathbb{R}^4 .

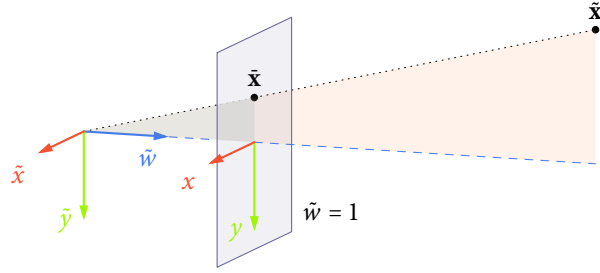


Figure 2.6: Homogeneous representation $\tilde{\mathbf{x}} = [\tilde{x}, \tilde{y}, \tilde{w}]$ of a two-dimensional vector \mathbf{x} . Homogeneous points located on the dotted line are all representations of the same point \mathbf{x} , with its augmented vector $\tilde{\mathbf{x}} = [x, y, 1]^T$ being located on the plane at $\tilde{w} = 1$.

Retrieving the Cartesian coordinates from the homogeneous representation is done by dehomogenization, i. e., dividing the coordinates by their \tilde{w} component:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \frac{1}{\tilde{w}} \tilde{\mathbf{x}} = \frac{1}{\tilde{w}} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \tilde{x}/\tilde{w} \\ \tilde{y}/\tilde{w} \\ \tilde{z}/\tilde{w} \\ 1 \end{bmatrix}$$

Homogeneous coordinates of a vector $\mathbf{v} = \mathbf{q} - \mathbf{p}$ are defined as the difference between homogeneous coordinates of the points \mathbf{q}, \mathbf{p} :

$$\tilde{\mathbf{v}} = \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ 1 \end{bmatrix} - \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \end{bmatrix} \in \mathbb{R}^4.$$

Homogeneous coordinates with $\tilde{w} = 0$ can not be converted into 3D coordinates; they are considered points at infinity.

Using homogeneous representation, a rigid-body transformation (2.13) can be rewritten in linear form:

$$\tilde{\mathbf{x}}_a = \begin{bmatrix} \mathbf{x}_a \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} R_{ab} & \mathbf{t}_{ab} \\ 0 & 1 \end{bmatrix}}_{\in SE(3)} \begin{bmatrix} \mathbf{x}_b \\ 1 \end{bmatrix}. \quad (2.15)$$

2.2 PROJECTIVE GEOMETRY

An *orthographic projection* of a world point $\tilde{\mathbf{x}}_c$ in 3D coordinates to a 2D point $\tilde{\mathbf{x}}$ can be written using homogeneous coordinates:

$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{x}}_c.$$

This projection drops the z component but keeps the homogeneous coordinate of the world vector. It is approximate for telephoto lenses and exact for telecentric lenses. To fit world coordinates, that can be measured in mm, onto an image sensor, which ultimately measures in pixels, they need to be scaled using a *scaled orthography*:

$$\tilde{\mathbf{x}} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{x}}_c,$$

where the unit for s would be pixels/mm to convert from metric 3D points into pixels.

In orthographic projections, parallel lines stay parallel but there is no perception of depth as the projection of an object has the same size no matter the z distance of the object. In perspective projections, projected lines don't stay parallel, but depth can be perceived as they are skewed toward vanishing points. Perspective projections resemble the human perceived vision and are thus commonly used in computer graphics and computer vision.

Using homogeneous coordinates, a perspective projection can be written in linear form as

$$\tilde{\mathbf{x}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0} \tilde{\mathbf{x}}_c,$$

where Π_0 is the *standard projection matrix*, which drops the homogeneous component, rendering the distance of the point from the image plane unrecoverable as expected by 2D projections.

2.2.1 Camera Model

The simplest optical device to capture an image is the *pinhole camera*. Its mathematical model describes the projection of points in three-dimensional space onto a two-dimensional plane by means of central projection. Considering the mathematical model of the pinhole camera shown in Figure 2.7, the image \mathbf{x}_s of the point \mathbf{x}_c in camera coordinates is given by the ideal perspective projection

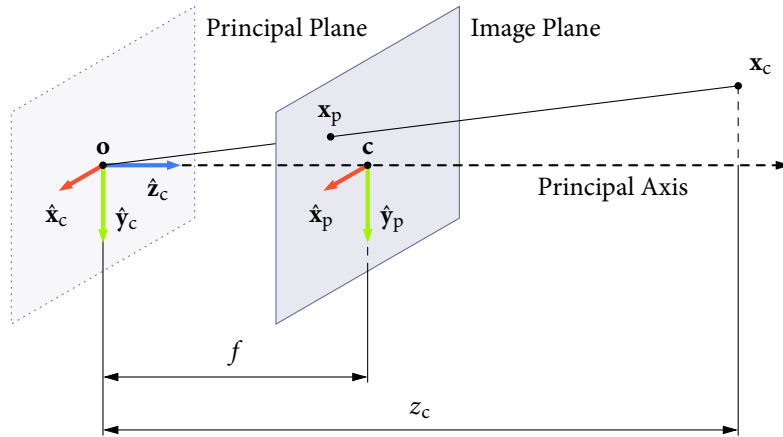


Figure 2.7: With the pinhole camera model, light rays from a 3D point \mathbf{x}_c at distance z_c pass the image plane on their way to the optical center \mathbf{o} . They intersect the virtual image plane that is located parallel to the principal plane at the distance of the focal length f , and form the two-dimensional image \mathbf{x}_p given in screen coordinates.

$$\tilde{\mathbf{x}}_s = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{x}}_c.$$

The size of the image is determined by the only parameter of the pinhole camera, the distance between the principal plane and the focal plane: the *focal length* f . The 2D coordinates of the point on the image plane can then be retrieved by *perspective division*, i. e., division by their z component:

$$\mathbf{x} = \pi(\tilde{\mathbf{x}}) = \begin{bmatrix} x_c/z_c \\ y_c/z_c \\ 1 \end{bmatrix}.$$

The matrix containing the focal length can be decomposed into the product of the two matrices:

$$\tilde{\mathbf{x}}_p = \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{K_f} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0} \tilde{\mathbf{x}}_c.$$

where Π_0 is the standard projection matrix from before. If the focal length f is known, it can be normalized to $f = 1$ by changing the units of the image coordinates. The projection of the point \mathbf{x}_c to the image plane at $z_c = 1$ then simplifies to

$$\tilde{\mathbf{x}}_p = \Pi_0 \tilde{\mathbf{x}}_c.$$

Until now, the ideal image coordinates \mathbf{x}_p were specified with respect to the image plane within the camera coordinate frame $\{p\}$ that has its origin at the *principal point* \mathbf{c} —the intersection of the optical axis and the image plane. However, in practice, measurements of a digital camera are usually given in pixels with respect to the coordinate frame $\{a\}$ of the pixel array with its origin being located at the upper-left corner. This introduces a translation, which also compensates for the physical misalignment between the center of the image sensor and the optical axis. Furthermore, if pixels do not have unit scale, they require scaling in x and y direction, s_x and s_y . If pixels are not rectangular, a *skew factor* s_θ is required. It is proportional to $\cot \theta$, where θ is the angle between the image axes $\hat{\mathbf{x}}_p$ and $\hat{\mathbf{y}}_p$. The parameters described above are introduced by the matrix K_s , extending the previous model to

$$\tilde{\mathbf{x}}_a = \underbrace{\begin{bmatrix} s_x & s_\theta & c_x \\ 0 & s_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{K_s} \underbrace{\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{K_f} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0} \tilde{\mathbf{x}}_c.$$

The described relationship between the normalized image coordinates \mathbf{x}_c and their coordinates within the pixel array \mathbf{x}_a is illustrated in Figure 2.8.

The transformation left of the perspective projection Π_0 can be expressed by a single *intrinsic* parameter matrix, also called *calibration matrix*

$$K = K_s K_f = \begin{bmatrix} f s_x & f s_\theta & c_x \\ 0 & f s_y & c_y \\ 0 & 0 & 1 \end{bmatrix},$$

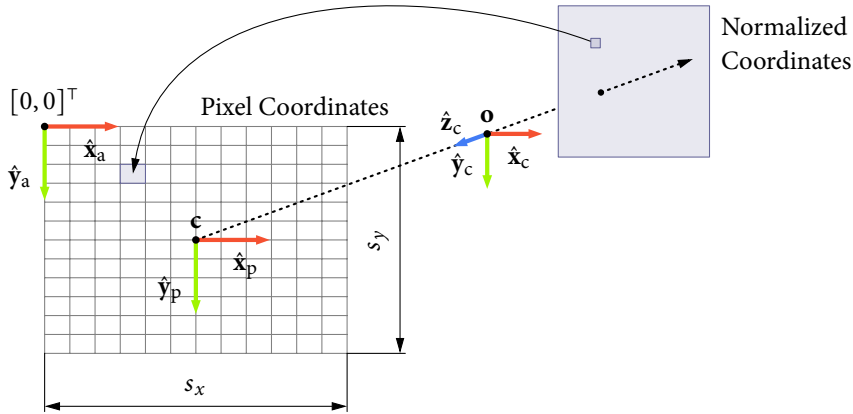


Figure 2.8: Transformation from normalized image coordinates to pixel coordinates.

where c_x and c_y are the x and y coordinates of the principal point measured in pixels, fs_x and fs_y are the size of unit length in horizontal and vertical pixels, respectively, with their ratio being the *aspect ratio* σ . The angle θ of the skew of the pixel fs_θ is typically very close to 90° and thus the skew is often close to zero. In practice, it is often assumed that $s_\theta = 0$, and since modern digital cameras usually have square pixels, $s_x = s_y$.

The coordinates of point \mathbf{x} were already given with reference to the camera coordinate system $\{c\}$. For coordinates given with respect to a world coordinate frame $\{r\}$, an additional rigid-body transformation (2.15) is required. As a function of world coordinates \mathbf{x}_r , the overall model for image formation then becomes

$$\tilde{\mathbf{x}}_p = K\Pi_0 \begin{bmatrix} R_{cr} & \mathbf{t}_{cr} \\ 0 & 1 \end{bmatrix} \tilde{\mathbf{x}}_r.$$

The process to determine a camera's individual intrinsic parameters is called *camera calibration* and is described in the next section. With known calibration matrix K , the normalized ideal image coordinates \mathbf{x}_c that correspond to the given pixel coordinates \mathbf{x}_a can be obtained by inverting K :

$$\tilde{\mathbf{x}}_c = K^{-1}\tilde{\mathbf{x}}_a.$$

2.2.2 Camera Calibration

The previous section described digital image formation and how it is modeled by the intrinsic parameters of an individual camera. The parameters of the calibration matrix K model linear distortions. But in practice, lenses also introduce distortions along tangential and radial directions. The distortions that are most frequently encountered are radially symmetric and can be classified as either barrel or pincushion distortions shown in Figure 2.9. They show significantly in images

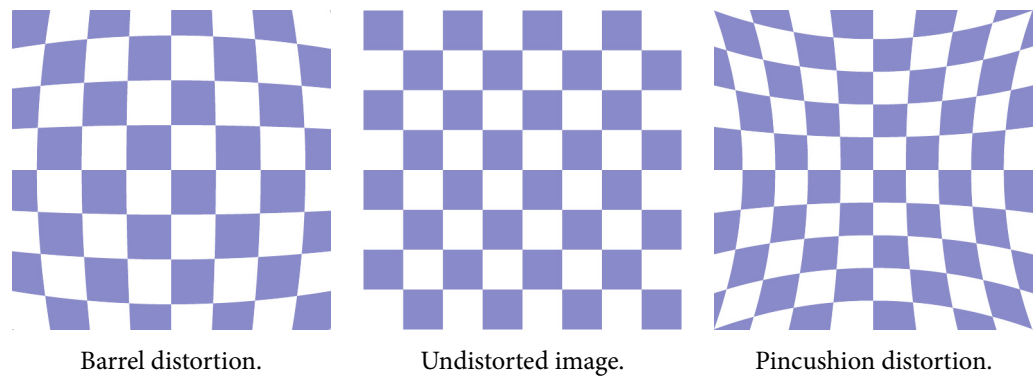


Figure 2.9: Examples of radial distortions.

from cheaper cameras that have a wide field of view (FOV), i. e., smaller focal lengths. The additional modelling of these radial distortions increases the accuracy of computer vision algorithms.

A general model for the relationship between a distorted point $\mathbf{x}_d = [x_d, y_d]^\top$ and an undistorted point $\mathbf{x} = [x, y]^\top$ can be modeled as

$$\mathbf{x} = \mathbf{c} + f(r)(\mathbf{x}_d - \mathbf{c}), \quad \text{with } f(r) = 1 + k_1 r + k_2 r^2 + k_3 r^3 + k_4 r^4.$$

Here, $r = \|\mathbf{x}_d - \mathbf{c}\|$ is the distance to the center of distortion \mathbf{c} and the distortion correction factor $f(r)$ is a polynomial of degree four with $k_{1...4}$ being the so-called distortion coefficients [Ma+03].

Under the assumption that pixels are rectangular, i. e., $s_\theta = 0$, and that the center of projection coincides with the center of radial distortion, the intrinsic parameters consist of $f_x, f_y, c_x,$ and c_y , contained in the calibration matrix K and the distortion coefficients $k_{1...4}$. To invert the image formation process, namely to determine a

world point given its image, the intrinsic parameters must be computed. This usually happens offline in advance. The parameters can be assumed not to change as long as the camera is not subject to temperature fluctuations or the focal length is changed manually by zooming in or out.

Intrinsic (and extrinsic) parameters are usually determined by taking images of a calibration rig, that can be a planar pattern or other known structure consisting of distinct features that are easily detected in images. Various methods for camera calibration are publicly available in software packages, e. g., the one described by Zhang [Zha00].

Two widely used planar calibration rigs are shown in Figure 2.10. By identifying the

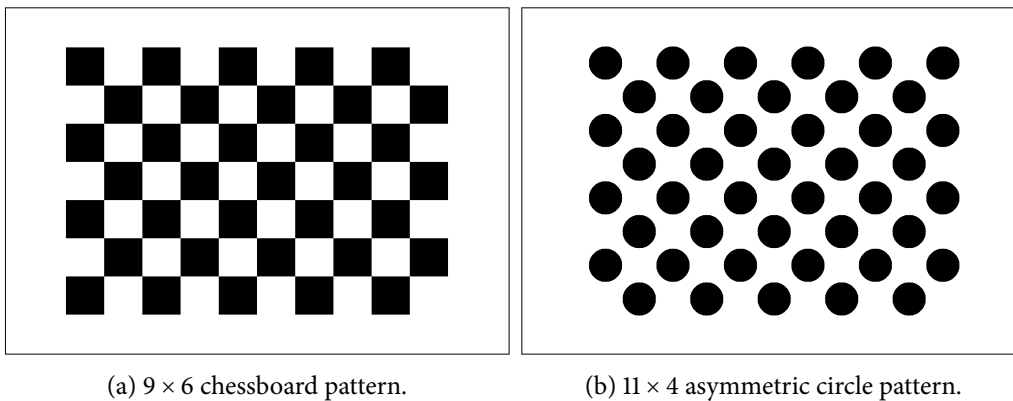


Figure 2.10: Commonly used planar calibration rigs.

structures from different angles, it is possible to determine the camera's position and orientation in each image as well as the intrinsic parameters. A series of sample images is shown in Figure 2.11. In each image, the features of the calibration rig are detected. For the chessboard pattern, these features are the chessboard corners, and for the circular grid, they are the circle centers. Due to the high contrast, these features can be robustly identified in the image projections; their 2D coordinates are referred to as *image points*.

The input to the calibration algorithm consists of the 2D image points and the 3D *object points*, a set of coordinates of the calibration rig features. It is assumed that the rig is always in the xy plane, thus $z = 0$, and feature coordinates can be provided as 2D coordinates. If the chessboard square sizes or the circle center distances are provided in metric units, the resulting calibration matrix is also metric. After the

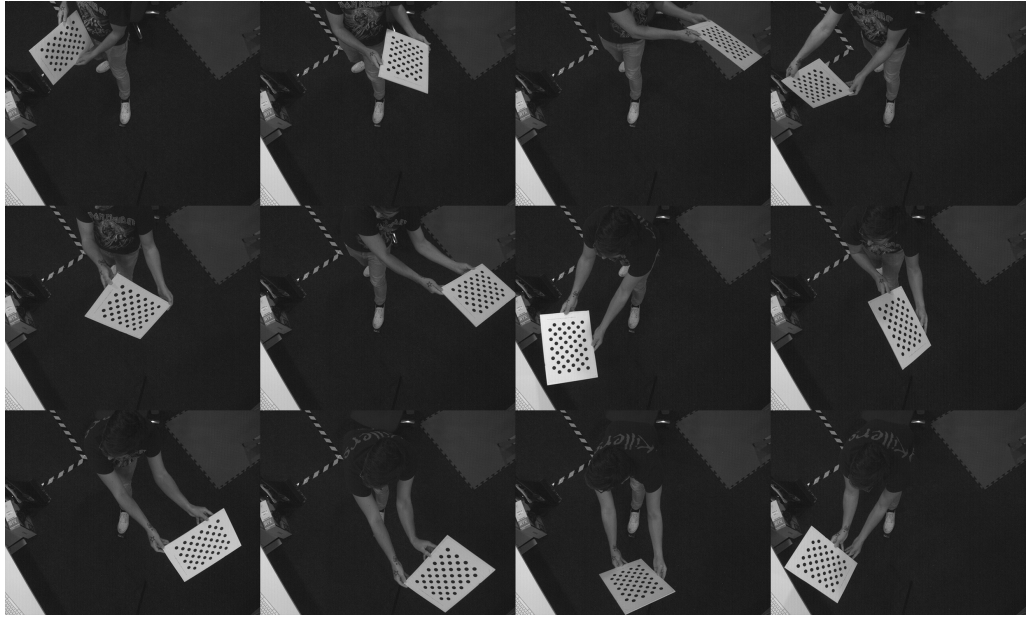


Figure 2.11: A series of sample images of an 11×4 asymmetric circle pattern printed on an A3 sheet of paper used for camera calibration.

offline calibration process, whole images, or more effectively, individual image points of interest can be undistorted.

2.3 QUADROTOR DYNAMICS

To analyze, simulate and control physical systems, mathematical models need to be obtained. Differential equations can describe the dynamic behavior of such systems, e. g., the response (rotational frequency) of an electric motor to input changes or excitation (applied voltage) can be modeled as a first-order system.

A first-order system, in its basic form, is described by the differential equation

$$\tau \dot{y}(t) + y(t) = k u(t)$$

with the time constant τ , the first-order derivative $\dot{y}(t)$, the system's response $y(t)$, the steady state value k and input $u(t)$. Forces and torque, e. g., of a quadrotor, can be modeled as first-order systems.

The time constant τ characterizes the time taken for a first-order linear time-invariant (LTI) system to reach $1 - 1/e \approx 63.2\%$ of its steady-state value in response to a step input. It is a measure of the system's response and used in dynamical systems and control theory. The larger the time constant τ of a system, the slower its response.

The motors and rotors of a quadrotor induce forces and torques acting on the quadrotor's body. Forces cause change in velocity, i. e., linear acceleration, while torques cause change in angular velocity, i. e., angular acceleration. A quadrotor can be modeled based on *Euler's laws of motion*.

The first of Euler's laws of motion states that the rate of change of a body's linear momentum $\dot{\mathbf{p}}$ is equal to the sum of all external forces \mathbf{f} acting on the body:

$$\dot{\mathbf{p}} = \mathbf{f}.$$

Euler's second law of motion defines that the rate of change of angular momentum about a point in the inertial reference frame (often the center of mass) on the rigid body is equal to the sum of torques $\boldsymbol{\eta}$ acting on that body about that point. For rigid bodies with moment of inertia $J = \text{diag}(j_x, j_y, j_z)$, angular velocity $\boldsymbol{\omega}$ and angular momentum $J\boldsymbol{\omega}$, this can be expressed as

$$\boldsymbol{\eta} = J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times J\boldsymbol{\omega}.$$

The quadrotor model presented in this section is a commonly used mathematical model [Mic+10; Lup+14], which neglects aerodynamic effects. While more accurate mathematical models [Kai+17; SMK17; FFS18] incorporate aerodynamic effects such as drag or disturbances, this level of detail is not necessary for the objectives of this testbed.

Four single rotors determine the motion of the quadrotor, that is modeled as a rigid body. Changes in the independent rotor thrusts f_i induce an acceleration c due to collective thrust in a quadrotor with mass m :

$$c = \frac{f_1 + f_2 + f_3 + f_4}{m}. \quad (2.16)$$

For simplicity, the center of mass is assumed to coincide with the geometrical center of the quadrotor. For a quadrotor in \times configuration, using the coordinate

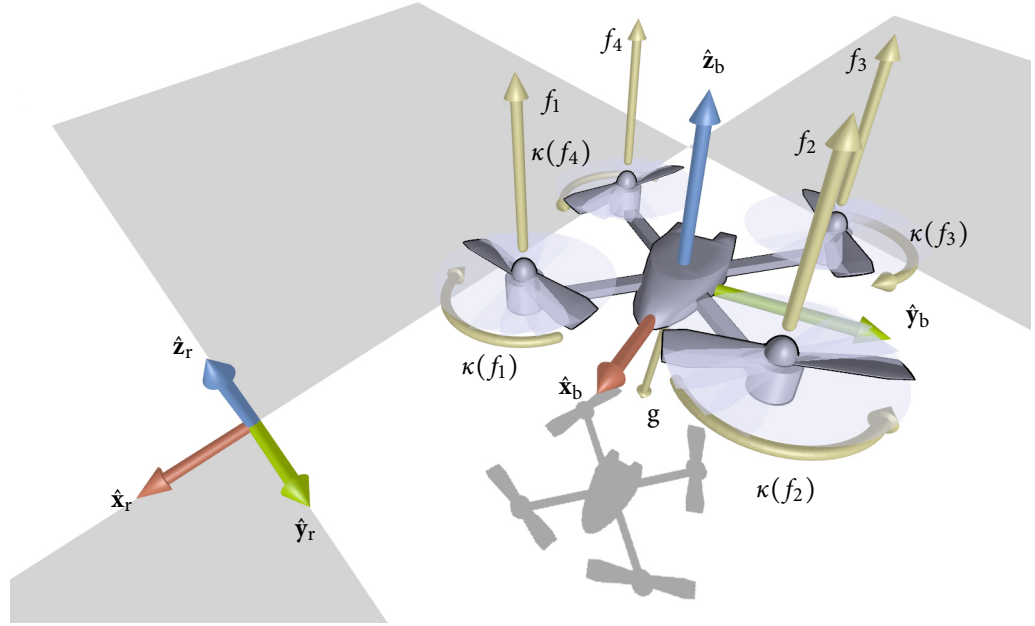


Figure 2.12: Quadrotor dynamics illustrated: The forces f_i generated by the rotors induce linear acceleration along the quadrotor's z axis \hat{z}_b . Angular accelerations about the roll and pitch axes are induced by disparities in the independent rotor thrusts, while imbalances in motor torques $\kappa(f_i)$, which counteract the direction of rotor rotation, lead to angular acceleration about the quadrotor's yaw axis. Gravity g acts along the negative world axis $-\hat{z}_r$.

system and rotor numbering illustrated in Figure 2.12, the moment of force vector $\boldsymbol{\eta}^\times$ acting on the quadrotor's center of mass can be formulated as

$$\boldsymbol{\eta}^\times = \begin{bmatrix} \eta_1^\times \\ \eta_2^\times \\ \eta_3^\times \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2}l(-f_1 + f_2 + f_3 - f_4) \\ \frac{\sqrt{2}}{2}l(-f_1 - f_2 + f_3 + f_4) \\ -\kappa(f_1) + \kappa(f_2) - \kappa(f_3) + \kappa(f_4) \end{bmatrix}, \quad (2.17)$$

where l is the quadrotor's center-to-rotor-distance, and κ is a linear polynomial that relates a single rotor's thrust to the generated drag torque. The torques about the roll and pitch axes, η_1^\times and η_2^\times , are determined by the sum of moments of forces. These torques are expressed as a three-dimensional pseudovector, defined by the cross product of the force vector \mathbf{f} and the displacement vector \mathbf{l} , given by $\boldsymbol{\eta} = \mathbf{l} \times \mathbf{f}$. Thus, imbalances in the independent rotor thrusts cause angular accelerations about the roll and pitch axes. Imbalances in rotor torque η_3 induce angular acceleration about the quadrotor's yaw axis, acting opposite to the direction of rotation of the rotors.

The rotor configuration of the quadrotor model described here is often referred to as *true* \times , where the motors are evenly distributed on a circle around the quadrotor's center. The square roots in (2.17) determining the torque components for η_1^\times and η_2^\times become clear when looking at the top view of the two common rotor configurations \times and $+$ in Figure 2.13. Torque about the yaw axis, η_3 , is solely based on the “torque

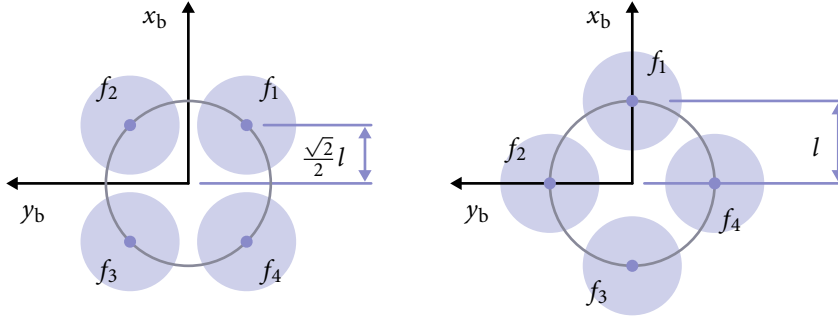


Figure 2.13: Illustration of the lever arms for quadrotors in \times (left) and $+$ (right) configurations.

effect”, which induces a rotation opposite to the direction of the rotors’ spin, and is equal to the in \times and $+$ configurations. A $+$ configuration, where the first rotor points in the direction of the \hat{x}_b axis, only differs from the \times configuration by the torques about the roll and pitch axes, (2.17) then simplifies to

$$\boldsymbol{\eta}^+ = \begin{bmatrix} l(f_2 - f_4) \\ l(f_1 - f_3) \\ \eta_3^\times \end{bmatrix}.$$

For quadrotors in \times configuration, rotational acceleration about the roll and pitch axes is affected by four rotors, as illustrated in Figure 2.13. Utilizing this configuration leads to an increase (by a factor of $\sqrt{2}$) in moments of force, resulting in improved stability compared to the two rotors inducing rotational acceleration in the $+$ configuration.

The relation between the motor input u and the resulting rotor thrust f can be described as a quadratic polynomial

$$f(u) = k_2^f u^2 + k_1^f u + k_0^f.$$

The induced drag torque can be linearly related to the generated force by

$$\kappa(f) = k_1^\kappa f + k_0^\kappa.$$

The thrust generated by a single rotor can be modeled as a first-order system

$$\dot{f} = \frac{1}{\tau_f}(f_{\text{ref}} - f) \quad (2.18)$$

with time constant τ_f , desired and current thrusts f_{ref} and f , respectively, under the condition $0 < f_{\text{min}} \leq f \leq f_{\text{max}}$. While modern electronic speed controllers actively break and thus are able to spin up and down equally fast [FFS18], other experiments (including the acoustic approach proposed in Section 3.3.3) show that motors require more time to spin down than to spin up [Lup+14], thus applies

$$\tau_f = \begin{cases} \tau_{f\uparrow}, & \text{for } f_{\text{ref}} > f \\ \tau_{f\downarrow}, & \text{else.} \end{cases}$$

The induced collective thrust (2.16) accelerates the quadrotor along its $\hat{\mathbf{z}}_b$ axis. Gravity acts along the $-\hat{\mathbf{z}}_r$ axis. Linear dynamics of a quadrotor can be described by

$$\dot{\mathbf{p}} = \mathbf{v}, \quad (2.19)$$

$$\dot{\mathbf{v}} = -g \mathbf{z}_r + c \mathbf{z}_b, \quad (2.20)$$

with the quadrotor's position \mathbf{p} , and velocity \mathbf{v} . Angular dynamics with respect to the quadrotor's center of mass can be written as

$$\dot{R}_{rb} = R_{rb} \widehat{\boldsymbol{\omega}}, \quad (2.21)$$

$$\dot{\boldsymbol{\omega}} = J^{-1}(\boldsymbol{\eta} - \boldsymbol{\omega} \times J \boldsymbol{\omega}),$$

with the quadrotor's orientation R_{rb} , the skew-symmetric matrix $\widehat{\boldsymbol{\omega}} \in so(3)$ formed from its body rates $\boldsymbol{\omega}$, moment of inertia $J = \text{diag}(j_x, j_y, j_z)$, body torques $\boldsymbol{\eta}$, angular rate $\boldsymbol{\omega}$, and angular momentum $J \boldsymbol{\omega}$.

The described quadrotor dynamics can also be expressed using quaternions to represent the quadrotor's attitude and its evolution. The linear acceleration (2.20) is then given by

$$\dot{\mathbf{v}} = -g \mathbf{z}_r + \mathbf{q}_{rb} \odot c \mathbf{z}_b, \quad (2.22)$$

and the evolution of the quadrotor's attitude (2.21) is then governed by the quaternion time derivative (2.11):

$$\dot{\mathbf{q}} = \frac{1}{2} \Omega(\boldsymbol{\omega}) \mathbf{q}_{rb}. \quad (2.23)$$

2.3.1 Differential Flatness

The quadrotor dynamics described in the previous section with the four inputs \mathbf{u} , with u_1 being the net body force and u_2, u_3, u_4 the body moments is a differentially flat system [MK11]. A differentially flat system is a type of dynamic system for which there exists a coordinate transformation, *flat output*, that makes it possible to express all the state variables and their derivatives as a function of a set of flat outputs and their derivatives. In [MK11], the flat outputs

$$\boldsymbol{\sigma} = [x, y, z, \psi]^\top$$

were constructed, with $[x, y, z]^\top$ being the center of mass of the quadrotor in the world coordinate frame and ψ its heading angle. The flat outputs facilitate trajectory generation, allowing the underactuated quadrotor to track any smooth trajectories in the space of flat outputs, as long as they are reasonably bounded.

The important property of a differentially flat system is that there exists a set of (virtual) flat outputs that can be used to express all states and inputs in terms of those outputs and their time derivatives [VM98]. More precisely, a nonlinear dynamical system of the form

$$\begin{aligned}\dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}), & \mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m \\ \mathbf{y} &= h(\mathbf{x}), & \mathbf{y} \in \mathbb{R}^m\end{aligned}$$

is differentially flat if there are outputs \mathbf{z} of the form

$$\mathbf{z} = \zeta(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \overset{k}{\dot{\mathbf{u}}}), \quad \mathbf{z} \in \mathbb{R}^m$$

such that

$$\begin{aligned}\mathbf{x} &= \mathbf{x}(\mathbf{z}, \dot{\mathbf{z}}, \dots, \overset{k}{\dot{\mathbf{z}}}) \\ \mathbf{u} &= \mathbf{u}(\mathbf{z}, \dot{\mathbf{z}}, \dots, \overset{k}{\dot{\mathbf{z}}}).\end{aligned}$$

The number of flat outputs is always equal to the number of inputs. The *tracking outputs* \mathbf{y} and the *flat outputs* \mathbf{z} are not necessarily the same. All feasible trajectories of a differentially flat system can be written as functions of its flat outputs and their derivatives. The behavior of a flat system is determined by its flat outputs. This property facilitates the generation of trajectories, as they can be planned in output space and then mapped to appropriate inputs.

2.3.2 Trajectories

Quadrotor trajectories are usually generated by algorithms that minimize the vehicle's jerk to achieve smooth motion. Jerk is the rate of change of acceleration with respect to time, which is the third derivative of the position vector. The first and second derivatives of the position vector are velocity and acceleration, respectively.

A common choice for smooth trajectories are quintic (fifth-order) piecewise polynomial functions [Cor13]. Their first and second time derivatives—velocity and acceleration—are continuous and thus quintic polynomials result in smooth trajectories. It is also straightforward to establish boundary conditions, including position, velocity, acceleration, and time. A scalar trajectory quintic polynomial and its first and second time derivatives can be expressed as

$$\begin{aligned} s(t) &= at^5 + bt^4 + ct^3 + dt^2 + et + f \\ \dot{s}(t) &= 5at^4 + 4bt^3 + 3ct^2 + 2dt + e \\ \ddot{s}(t) &= 20at^3 + 12bt^2 + 6ct + 2d. \end{aligned}$$

Writing these equations in matrix form results in a linear system

$$\begin{bmatrix} s_0 \\ \dot{s}_0 \\ \ddot{s}_0 \\ s_T \\ \dot{s}_T \\ \ddot{s}_T \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ T^5 & T^4 & T^3 & T^2 & T^1 & 1 \\ 5T^4 & 4T^3 & 3T^2 & 2T^1 & 1 & 0 \\ 20T^3 & 12T^2 & 6T^1 & 2 & 0 & 0 \end{bmatrix}}_{A_T} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}, \quad (2.24)$$

with system matrix A_T , polynomial coefficient vector $\mathbf{c} = [a, b, c, d, e, f]^T$, and condition vector $\mathbf{s}_{[0,T]} = [\mathbf{s}(0), \mathbf{s}(T)]^T$, that combines two boundary states $\mathbf{s}(t) = [s(t), \dot{s}(t), \ddot{s}(t)]$ with start time $t = 0$ and end time $t = T$, the segment duration. The boundary states $\mathbf{s}_i(t_i)$ can be viewed as keyframes. A list of $n + 1$ keyframes represents a trajectory \mathcal{K} , i. e., $\mathcal{K} = (\mathbf{s}_0(t_0), \dots, \mathbf{s}_n(t_n))$. Thus it consists of n trajectory segments $\mathbf{s}_{[i,i+1]} = [\mathbf{s}_i(t_i), \mathbf{s}_{i+1}(t_{i+1})]^T$, $i \in \{0, \dots, n-1\}$. The polynomial coefficient vector \mathbf{c} can be uniquely determined by solving the linear system (2.24) for $T \neq 0$. Specifically, the corresponding polynomial coefficients \mathbf{c}_i , $i \in \{0, \dots, n-1\}$ of a trajectory segment are determined by solving (2.24) with $T = (t_{i+1} - t_i)$, i. e.,

$$\mathbf{c}_i = A_T^{-1} \mathbf{s}_{[i,i+1]}.$$

For n trajectory segments, this results in the polynomials $\mathbf{s}_i(t)$, $i \in \{0, \dots, n-1\}$. To obtain a trajectory that is usable for the controller, the piecewise polynomials are sampled at a specific rate f_s :

$$\mathbf{s}_i(t_j), \quad t_j = \frac{j}{f_s} T, \quad j \in \{0, \dots, f_s\}.$$

In the case of a multi-dimensional vector, the piecewise polynomials are independently interpolated in each dimension and then combined to form the final trajectory that is passed as reference trajectory to the controller. The described trajectory includes only position-related parameters, but keyframes can be extended to include heading angles. It is common to linearly interpolate these angles between keyframes as needed.

2.4 QUADROTOR CONTROL

In order to autonomously pilot a quadrotor along a trajectory to a desired position and attitude, continuous control is required. The basic building block of a control system is a control loop, that consists of physical components and control functions that drive a measured process variable to a desired setpoint subject to external disturbances.

Based on an error term—in basic systems often derived by the difference of the setpoint and the process variable—a control variable is calculated to adjust the process variable toward the setpoint. The block diagram of a basic feedback (or closed-loop) control system is shown in Figure 2.14. It clarifies the relationships and introduces terms as well as the common naming of the variables.

In case of quadrotor control, the control variables are its inputs, namely, thrust, roll, pitch, and yaw. The exact error terms depend on the purpose of the controller. In this thesis, independent hover and a trajectory controllers are used. As an introduction, a proportional-integral-derivative (PID) controller is described—a widely used tool that is also employed in quadrotors. In such systems it is utilized in various functions, e. g., in low-level motor control or, in handheld flight, in driving the body rates toward the desired user input.

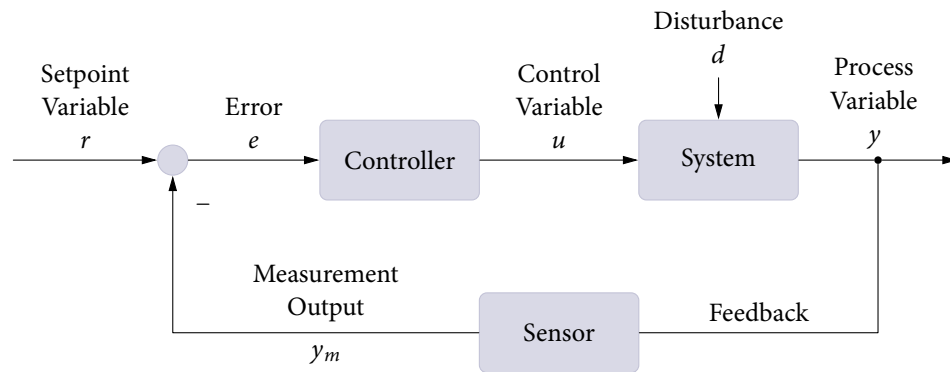


Figure 2.14: Block diagram of a basic feedback control.

2.4.1 PID Feedback Control

In control systems engineering, a highly established controller is the PID controller, which can be found in over 95 % of all control loops [ÅH06]. The history of this controller dates back to the beginning of the last century. It has become the standard in control systems engineering due to its satisfactory performance with comparative simplicity and intuitiveness, and is used in a wide range of processes across all industries [Vis06]. Detailed literature on control systems can be found, e. g., in [DB16].

A well-known use case of the PID controller is cruise control in cars. If achieving and sustaining a specific value is a requirement for the controller, feedback of the control variable, in this case, the speed of the vehicle, is necessary. Quadrotors commonly employ PID control to stabilize their orientation and manage their motion. In assisted quadrotor flight, the setpoints typically include the desired altitude and orientation. Outdoors, measurements of the desired altitude may be derived from global positioning system (GPS) or barometers. Indoors or near ground level, it may be fused from combinations of sonar and infrared (IR) ranging sensors. The orientation of a drone is typically measured using an IMU. The PID controller then calculates the appropriate correction to the quadrotor's motor speeds to achieve the desired orientation or motion.

If the controller is able to determine the control variable solely based on the setpoint variable, there is no need to feed back the process variable. A control system of this type is referred to as an open-loop system. However, open-loop control systems

lack the ability to adapt to disturbances, such as sudden changes, like encountering a steep grade in the case of cruise control.

As the initialism suggests, a PID controller is composed of three individual parts: a proportional, an integral, and a differential term. Unlike the integral and differential term, the proportional term does not take time into account and immediately acts on the current error. The integral term integrates deviations from the setpoint and relates to the past, thus compensating continuous errors. The differential term is proportional to the derivative of the error and predicts future errors by projection.

The block diagram of a standard PID controller and its relationships as well as the common naming convention is shown in Figure 2.14. The control error e represents the disparity between the setpoint variable r and the measured system output y_m , and is fed to the controller. Depending on the process, the controller calculates the control variable u , which is then applied to the system. The system is influenced by external disturbances d , affecting the process variable y . The measurement output y_m is measured by a dedicated device. The feedback of this measurement completes the control loop.

The *parallel* form of the PID controller is given by

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (2.25)$$

with the control error $e(t) = r(t) - y_m(t)$ and proportional, integral, and differential terms with their respective gain constants K_p , K_i , and K_d , also called *tuning parameters*. Time is denoted by t and $\tau = [0, t]$ is the variable of integration. This form is called *parallel* because the gain constants are decoupled.

Next to the parallel form exists the *standard* or interacting form of the PID controller

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right), \quad (2.26)$$

which has more physical meaning than the parallel form. The integral term intends to eliminate the sum of all past errors in T_i seconds or samples and the derivative term is the error estimation at time T_d seconds or samples. However, the gain constants are not strictly separated from each other. Next to the proportional action, K_p also affects the integral and derivative actions resulting in less intuitive controller tuning.

2.4.2 Position Control

A widely adopted position controller [MK11] calculates the body rates ω and the desired mass-normalized thrust c based on a reference trajectory point, which includes position \mathbf{p}_{ref} , velocity $\dot{\mathbf{p}}_{\text{ref}}$, acceleration $\ddot{\mathbf{p}}_{\text{ref}}$, body rates ω_{ref} , and heading angle ψ_{ref} . A simplified case of the calculations described in the following, where the errors in velocity and acceleration are zero, is illustrated in Figure 2.15.

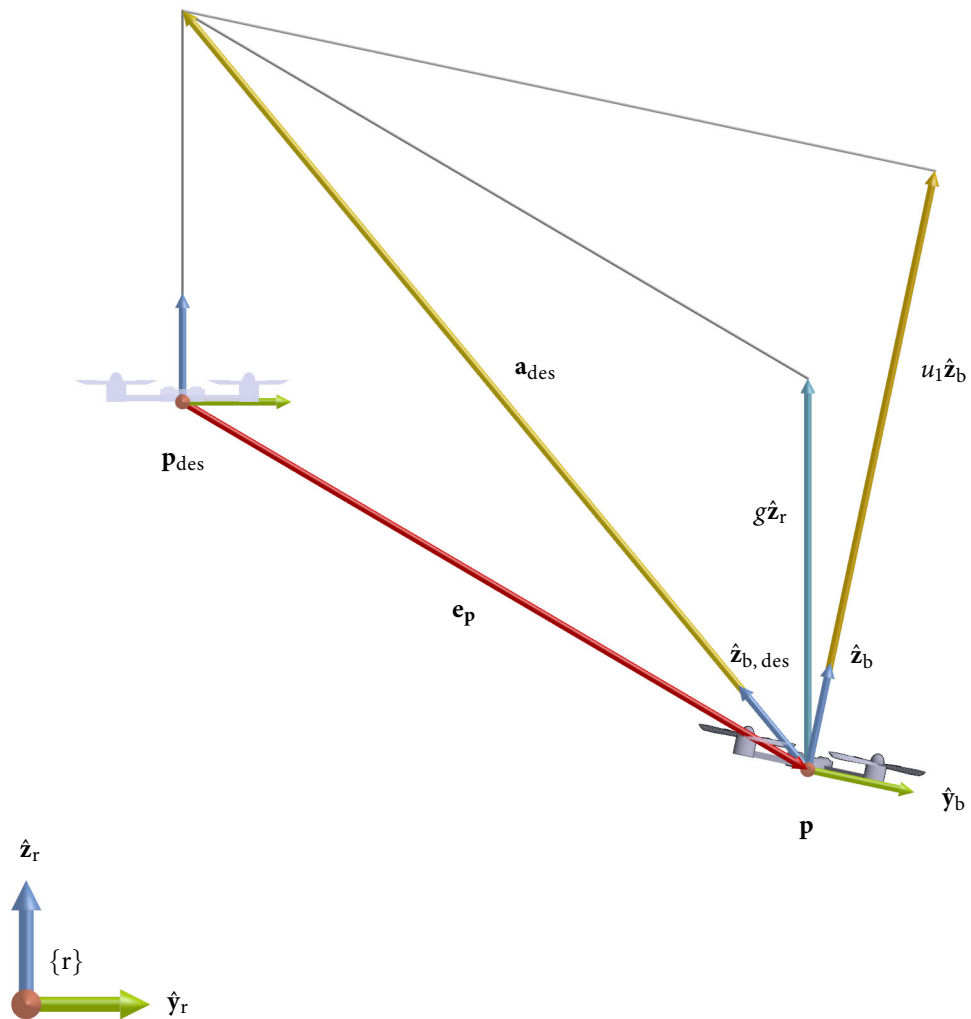


Figure 2.15: Illustration demonstrating the computation of the desired body axis $\hat{\mathbf{z}}_{\mathbf{b},\text{des}}$ and the thrust control input $u_1 = \mathbf{a}_{\text{des}}^T \hat{\mathbf{z}}_{\mathbf{b}}$ under the simplified conditions $\mathbf{e}_{\mathbf{p}} = \mathbf{0}$, $\dot{\mathbf{p}}_{\text{ref}} = \mathbf{0}$, and $K_{\mathbf{p}} = I$; gravity g is not displayed true to scale.

The errors in position \mathbf{e}_p and velocity $\mathbf{e}_{\dot{p}}$ are given by

$$\mathbf{e}_p = \mathbf{p} - \mathbf{p}_{\text{ref}}, \quad \mathbf{e}_{\dot{p}} = \dot{\mathbf{p}} - \dot{\mathbf{p}}_{\text{ref}},$$

where \mathbf{p} is the current position of the quadrotor (or process variable) and \mathbf{p}_{ref} is the target position (or setpoint). The desired acceleration vector \mathbf{a}_{des} is calculated as

$$\mathbf{a}_{\text{des}} = \ddot{\mathbf{p}}_{\text{des}} = -K_p \mathbf{e}_p - K_{\dot{p}} \mathbf{e}_{\dot{p}} + \ddot{\mathbf{p}}_{\text{ref}} + g \mathbf{z}_r,$$

where K_p and $K_{\dot{p}}$ are diagonal 3×3 gain matrices, and g is the gravity on Earth. The mass-normalized thrust control variable c is then the scalar projection of the desired acceleration vector onto the quadrotor's z axis:

$$c = \mathbf{a}_{\text{des}}^\top \hat{\mathbf{z}}_b.$$

The quadrotor's desired attitude is parametrized by the rotation matrix R_{des} whose basis is formed by the three unit vectors calculated in the following. The desired z axis is just along the previously calculated, desired acceleration vector:

$$\hat{\mathbf{z}}_{b,\text{des}} = \frac{\mathbf{a}_{\text{des}}}{\|\mathbf{a}_{\text{des}}\|}.$$

The reference heading angle ψ_{ref} is used to specify the quadrotor's desired x axis as

$$\hat{\mathbf{x}}_{b',\text{des}} = \begin{bmatrix} \cos \psi_{\text{ref}} \\ \sin \psi_{\text{ref}} \\ 0 \end{bmatrix},$$

as an intermediate step, as orthogonalization is required after calculating the desired y axis. This is done by the cross product of the desired $\hat{\mathbf{z}}_{b,\text{des}}$ axis and intermediate axis $\hat{\mathbf{x}}_{b',\text{des}}$:

$$\hat{\mathbf{y}}_{b,\text{des}} = \frac{\hat{\mathbf{z}}_{b,\text{des}} \times \hat{\mathbf{x}}_{b',\text{des}}}{\|\hat{\mathbf{z}}_{b,\text{des}} \times \hat{\mathbf{x}}_{b',\text{des}}\|}. \quad (2.27)$$

To obtain an orthonormal basis, the x axis needs to be recalculated:

$$\hat{\mathbf{x}}_{b,\text{des}} = \hat{\mathbf{y}}_{b,\text{des}} \times \hat{\mathbf{z}}_{b,\text{des}}.$$

Under the condition $\hat{\mathbf{z}}_{b,\text{des}} \times \hat{\mathbf{x}}_{b',\text{des}} = 0$ in the computation of the desired y axis (2.27), a singularity arises within $SO(3)$. This singularity induces significant changes in the unit vectors around that point in $SO(3)$, and is addressed by

determining a second, congruent solution for R_{des} with $-\hat{\mathbf{x}}_{\text{b, des}}$ and $-\hat{\mathbf{y}}_{\text{b, des}}$, and choosing the solution closer to the current attitude of the quadrotor [MK11].

Together, the desired orthonormal unit axes then form the columns of the desired rotation matrix R_{des} of the quadrotor's desired attitude:

$$R_{\text{des}} = \begin{bmatrix} \hat{\mathbf{x}}_{\text{b, des}} & \hat{\mathbf{y}}_{\text{b, des}} & \hat{\mathbf{z}}_{\text{b, des}} \end{bmatrix}.$$

The error in orientation is defined as

$$\mathbf{e}_R = \frac{1}{2} (R_{\text{des}}^\top R_{\text{rb}} - R_{\text{rb}}^\top R_{\text{des}})^\vee.$$

The error in angular velocity is the difference between the actual and the reference angular velocity:

$$\mathbf{e}_\omega = \boldsymbol{\omega} - \boldsymbol{\omega}_{\text{ref}}.$$

The desired body rates are the three remaining inputs and are given by

$$\boldsymbol{\omega} = -K_R \mathbf{e}_R - K_\omega \mathbf{e}_\omega$$

with 3×3 gain matrices K_R and K_ω .

2.4.3 Trajectory Control

A popular control technology well-suited for trajectories is model-predictive control (MPC) [RMD17]. It forecasts system behavior at regular intervals based on a dynamic model and predicts and optimizes future inputs in order to determine the best decision for the control input at the current time. The optimization is subject to input and output (or state) constraints. Input constraints are physical limitations of the system, e. g., actuator limits of a quadrotor. State constraints, e. g., the quadrotor's linear velocity, are usually *desirables* and maybe relaxed by the MPC if not achievable. MPC is probably the most widespread advanced control strategy and is used in academic and industrial context, e. g., for controlling chemical processes, or autonomous cars and robots. The popularity of MPC has also led to an increased interest in system identification because of the required dynamical model. The estimation of the required parameters of the quadrotor system model used is described later in Section 3.3.

MPC is sometimes called receding horizon control, which gives a more picturesque description of the control strategy: When driving a car through a long curve, the

driver looks at a distant point in the curve (the horizon) and reacts according to their estimation of the current state of the vehicle in order to reach this point eventually. For the current time step, the driver reacts accordingly by adjusting speed and steering direction before the entire time window, including the horizon, moves (recedes) one step ahead.

MPC is an optimal control strategy based on a model of the system and numerical optimization. For multiple steps in a short time window the control input is optimized. Then the input is applied to the immediate next step before the entire window shifts forward one time step and is optimized either from scratch or by using the previous solution as initialization.

There are several advantages of MPC over other control strategies such as PID or linear quadratic (LQ) control [RMD17]. Compared to PID control, MPC has the capability to predict costs using a cost function over the prediction horizon. While MPC supports input constraints, standard LQ control and PID controllers may calculate control inputs without physical meaning or that are unfeasible. Furthermore, with LQ control the solution is calculated one time offline in advance and then used over the entire time window. MPC, however, continuously optimizes the input online at every single time step. This is computationally intense, but enabled by fast computers. With today's hardware, even nonlinear optimization is able to run online at every time step; this method is called nonlinear model-predictive control (NMPC) [AZ12]. However, it is also possible to use linearized equations of motion for nonlinear systems, which are the easiest and fastest to optimize.

The dynamics of a system can be described as a set of differential equations $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, where \mathbf{x} is the state vector and \mathbf{u} is the input vector. The standard form of an optimization problem is then

$$\begin{aligned} \min_{\mathbf{u}} \quad & \int \mathcal{L}(\mathbf{x}, \mathbf{u}) \\ \text{subject to} \quad & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ & \mathbf{x}(t_0) = \mathbf{x}_{\text{init}}, \\ & \mathbf{r}(\mathbf{x}, \mathbf{u}) = 0, \\ & \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq 0, \end{aligned} \tag{2.28}$$

where \mathcal{L} is the objective function, \mathbf{x}_0 the initial condition, and $\mathbf{r}(\mathbf{x}, \mathbf{u})$, $\mathbf{h}(\mathbf{x}, \mathbf{u})$ represent equality and inequality constraints, respectively.

The dynamical model of the quadrotor as described in Section 2.3 is defined using (2.19), (2.22), and (2.23), and can be summarized to the 10-dimensional state space [Fal+18]

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{q}} \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{v} \\ -g\mathbf{z}_r + \mathbf{q} \odot c\mathbf{z}_b \\ \frac{1}{2}\widehat{\boldsymbol{\omega}}\mathbf{q} \end{bmatrix} \quad (2.29)$$

The state vector $\mathbf{x} = (\mathbf{p}, \mathbf{v}, \mathbf{q})^\top$ consolidates position $\mathbf{p} \in \mathbb{R}^3$, velocity $\mathbf{v} \in \mathbb{R}^3$, and attitude $\mathbf{q} \in \mathbb{S}^3$. The input vector $\mathbf{u} = (c, \boldsymbol{\omega}^\top)^\top$ consists of the mass-normalized thrust c (2.16) and angular rates $\boldsymbol{\omega}$.

To approach a target position, a quadrotor trajectory has to be planned considering two objectives. Firstly, the thrust produced by each motor is limited by upper and lower bounds, thus bounding the input vector \mathbf{u} . Secondly, the underactuated nature of the quadrotor, manifested by translational motion coupled to rotational motion, is contained in the system dynamics. The exploitation of the system dynamics is therefore necessary for quadrotors to follow a desired trajectory.

The basic formulation of the optimization problem (2.28) leads to a nonlinear program with quadratic costs, which is the typical approach used in practice. When objective function and constraints are twice continuously differentiable, a sequential quadratic program (SQP) can be used to iteratively approximate a solution to the nonlinear program that can be used as a MPC.

To solve the MPC optimization problem for a given system state \mathbf{x}_0 numerically in each time step dt , the system dynamics are discretized over the prediction horizon t_h . This results in control inputs $\mathbf{u}_i \forall i \in \{1, \dots, n-1\}$ and state trajectories $\mathbf{x}_i \forall i \in \{1, \dots, n\}$. The time-varying state cost matrix and input cost matrix are defined as diagonal matrices $Q_i \forall i \in \{1, \dots, n\}$ and $R_i \forall i \in \{1, \dots, n-1\}$, respectively. These are the tuning parameters for the MPC. Large values of Q in comparison to R result in quickly driving the system state to the origin, albeit at the expense of large control inputs. Conversely, when R is much larger than Q , it leads to a slower approach to the origin with reduced control action.

The resulting objective function is

$$\mathcal{L} = \bar{\mathbf{x}}_n^\top Q_n \bar{\mathbf{x}}_n + \sum_{i=1}^{n-1} (\bar{\mathbf{x}}_i^\top Q_i \bar{\mathbf{x}}_i + \bar{\mathbf{u}}_i^\top R_i \bar{\mathbf{u}}_i) \quad (2.30)$$

where the final step (left term) has no input cost and the values $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ are the differences from a target position or a discretized reference trajectory. The inputs \mathbf{u} and velocity \mathbf{v} are limited by the constraints

$$\begin{aligned} c_{\min} &\leq c \leq c_{\max}, \\ -\boldsymbol{\omega}_{\max} &\leq \boldsymbol{\omega} \leq \boldsymbol{\omega}_{\max}, \\ -\mathbf{v}_{\max} &\leq \mathbf{v} \leq \mathbf{v}_{\max}. \end{aligned}$$

An optimization problem is convex if both the feasible set and the objective function are convex. The feasible set refers to the largest subset of the state space for which a control action exists that satisfies all constraints. It is convex, if a line connecting any two points in the feasible set lies entirely within the set. When the state and input space of the problem are of convex domain, the quadratic costs are also convex. The objective function is convex since cost matrices are positive semi-definite ($\mathbf{v}^\top A \mathbf{v} \geq 0$) guaranteeing the existence and uniqueness of a solution. The convex nature of the optimization problem implies that local minima are, in fact, global minima, which is not the case with non-convex problems.

To conduct reliable research in the field of UAVs and HDI, controlled indoor test environments are the foundation. They usually include an optical outside-in pose estimation component and a radio control system to operate multiple UAVs. The General Robotics, Automation, Sensing, and Perception testbed [Mic+10], the Flying Machine Arena [Lup+14] and the Robotics and Perception Group's testbed [FFS18] are pioneering environments for UAV research, though they rely on costly technology. These testbeds typically use commercial multi-camera motion capture systems, such as Vicon¹ or OptiTrack² for pose estimation. The cost of these systems can vary significantly depending on the number of cameras, commonly starting in the lower six figure US dollar range. A widely used quadrotor for these testbeds was the Ascending Technologies (acquired by Intel in 2016) *Hummingbird*, weighing 0.5 kg and priced in the mid four figure range. Testbeds that operate with larger drone platforms require additional safety measures to ensure the well-being of humans, machines, and their environment. This includes provisions such as mats, safety nets, and double-glazed windows, which add to the complexity of the setup.

The high cost of the testbeds mentioned above limits their wider adoption. Currently, hobby quadrotor platforms are available at a starting price of \$10, and there are even open source programmable platforms available for under \$300. These affordable options provide a more accessible and cost-effective alternative for conducting research in the field of UAVs. The latest miniature quadrotors are small and lightweight, which means that extensive safety measures beyond the installation of fly screens may not be necessary. These lightweight platforms are unlikely to cause significant damage to humans or infrastructure components, reducing the need for aforementioned safety precautions.

Part of this thesis is an infrastructure for general research and application purposes consisting of a lightweight framework and software architecture resorting to

¹<https://www.vicon.com>

²<https://www.optitrack.com>

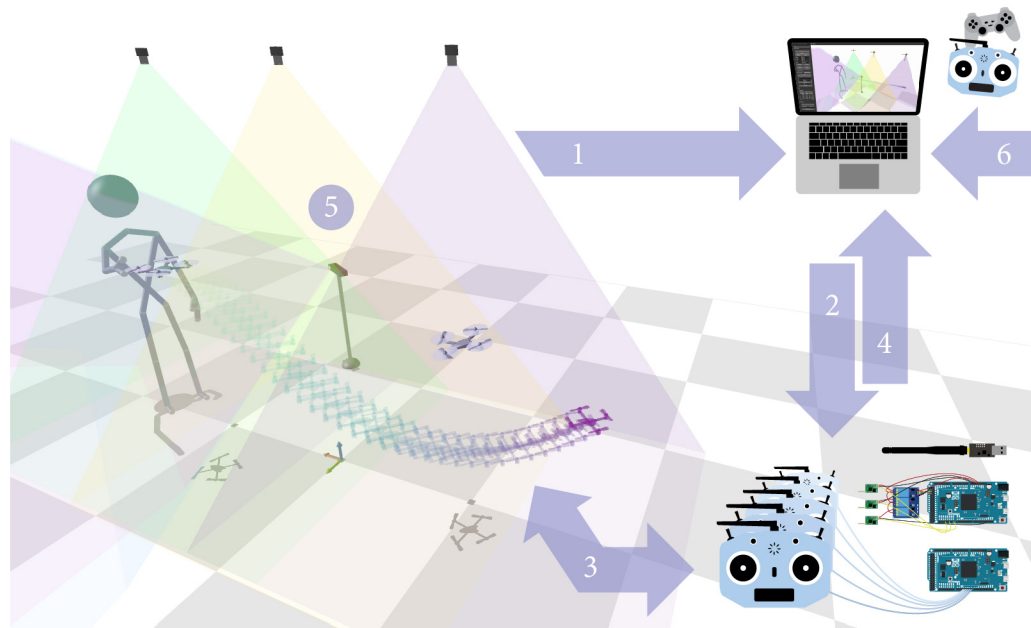


Figure 3.1: Overview of the *ICARUS* testbed: A standard laptop serves as the control station. Based on quadrotor poses (1) determined by an optical pose estimation system, control variables are determined and output (2) to various radio control systems that communicate with the quadrotors through proprietary modulation (3). Telemetry data is fed back to the control station (4) and is used to improve flight performance or to realize sensor-based user input. Additionally, skeleton data obtained through a dedicated sensor (5) can serve for gesture control or to enable security measures. Manual control of physical or simulated drones is possible with RC transmitters or gamepads (6).

affordable off-the-shelf hardware that is easily replicable. The result is the low-cost quadrotor testbed *ICARUS* [Lie+17] shown in Figure 3.1. It was developed and further improved over recent years and allows tracking and control of general off-the-shelf quadrotors in reality or simulation. This testbed offers reliable results by ensuring repeatable trajectories at an affordable cost, distinguishing it from other testbeds or settings in HDI user studies, where drones do not fly freely or autonomously. The minimum hardware setup of the system is highly portable and can be used for live demonstrations, such as in a lecture or exhibition, using just a single camera, laptop, and quadrotor. This also allows the setup in a living room environment, for example, where future interactions between humans and drones are likely to take place.

This chapter describes the individual parts and their composition to form the infrastructure. Section 3.1 covers quadrotor and human pose estimation, providing insights into the used hardware and implementation details, such as the registration of multiple cameras and safety measures. The approaches developed for the radio control of off-the-shelf hobby drone platforms by a computer are outlined in Section 3.2. The primary development quadrotor platform is described in detail in Section 3.3, including the estimation of simulation parameters and methods to compensate for battery voltage sag. In Section 3.4, insights into the software tying the aforementioned parts together are detailed. Section 3.5 discusses the challenges and opportunities of conducting user studies in HDI, highlighting solutions implemented in the testbed. The chapter concludes with Section 3.6, which describes exemplary applications already implemented using the infrastructure.

3.1 POSE ESTIMATION

To autonomously control a quadrotor, the primary requirement is determining its pose, i. e., position and orientation in space. This can be accomplished outdoors by utilizing information obtained from GPS. However, in GPS-denied environments, such as indoor spaces or disaster sites like collapsed buildings, pose estimation must depend on alternative sensors and algorithms. The most common method in such scenarios is the use of onboard cameras in conjunction with simultaneous localization and mapping (SLAM) algorithms for estimating the quadrotor's pose [ESC14].

In controlled indoor test environments, typically professional motion capture systems are used. A quite common tracking system comes from Vicon. To track a rigid-body object such as a quadrotor, their system uses five *passive markers*, that are 15 mm diameter spheres covered in retroreflective tape. They are attached to the object in a fixed arrangement. The tracking cameras are equipped with a ring flash of IR emitters, that are synchronized with the shutter of the camera. In an ideal environment, the images captured by the camera contain white projections for each of the markers on a black background. The pose estimation system used within the *ICARUS* infrastructure employs an inverted approach: The markers themselves are IR-emitters, commonly referred to as *active markers*, and they are used in combination with regular vision cameras equipped with appropriate filters.

Motion capture systems by Vicon are likely better known for tracking humans, such as body motion or facial expression for computer games, or motion analysis

in the field of performance diagnostics. However, the price factor may exclude smaller laboratories. For the purpose of tracking human movement to control quadrotors using gestures or to implement safety measures, a depth camera is integrated into the tracking system.

Being able to specify the skeletons and quadrotor poses with respect to a common coordinate frame is useful, e. g., to measure distances between drones and humans or to determine which quadrotor an interacting user pointed to. An overview of the different coordinate frames and corresponding coordinate system transformations present in the *ICARUS* test environment are depicted in Figure 3.2. How these are determined is described in the following.

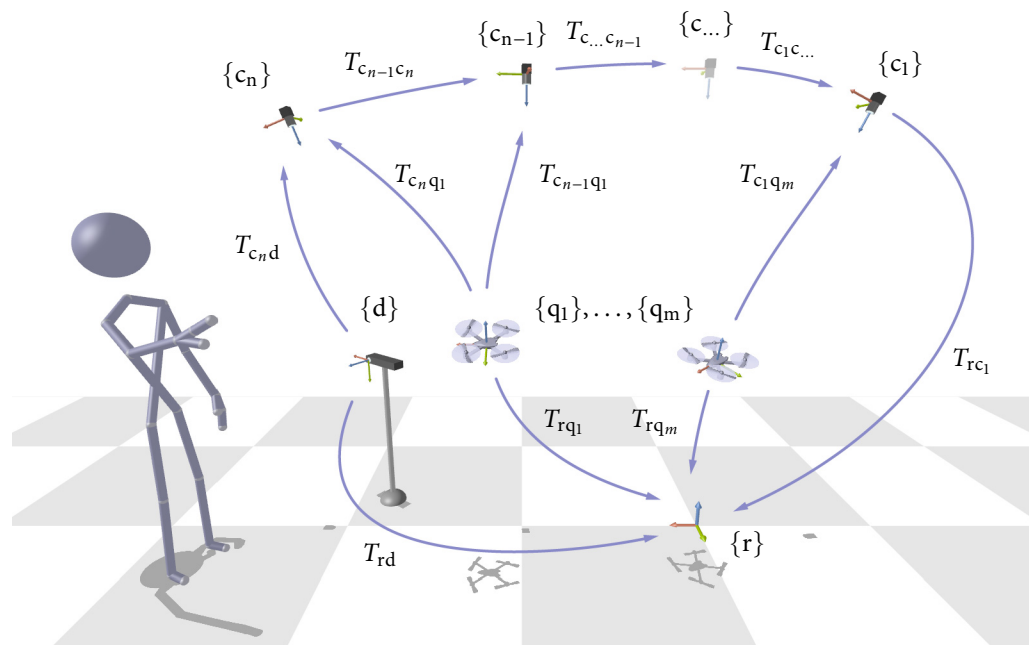


Figure 3.2: Coordinate system transformations T between individual components of the testbed and its common reference frame $\{r\}$. The individual coordinate frames correspond to the industrial vision cameras $\{c_1\}, \dots, \{c_n\}$, the quadrotors $\{q_1\}, \dots, \{q_m\}$, and a depth sensor $\{d\}$.

3.1.1 Camera Calibration

To reference skeleton poses as well as quadrotor poses measured from different cameras with reference to a common coordinate frame, the locations of all cameras, i. e., their extrinsic parameters, must be determined. Intrinsic camera parameters are required to take precise measurements and talk about points in 3D in a metric unit. The camera calibration step, as outlined in Section 2.2.2, determines both the intrinsic and extrinsic camera parameters. Calibration needs to be done one time in advance of taking measurements and does not have to be repeated until focal lengths or camera positions change.

For both the intrinsic and extrinsic calibration, a printed pattern attached to a planar surface is used as calibration rig. To allow pairwise extrinsic calibration, camera pairs need to have overlapping regions as shown in Figure 3.3, so that two cameras are able to capture the calibration pattern at the same time. For the

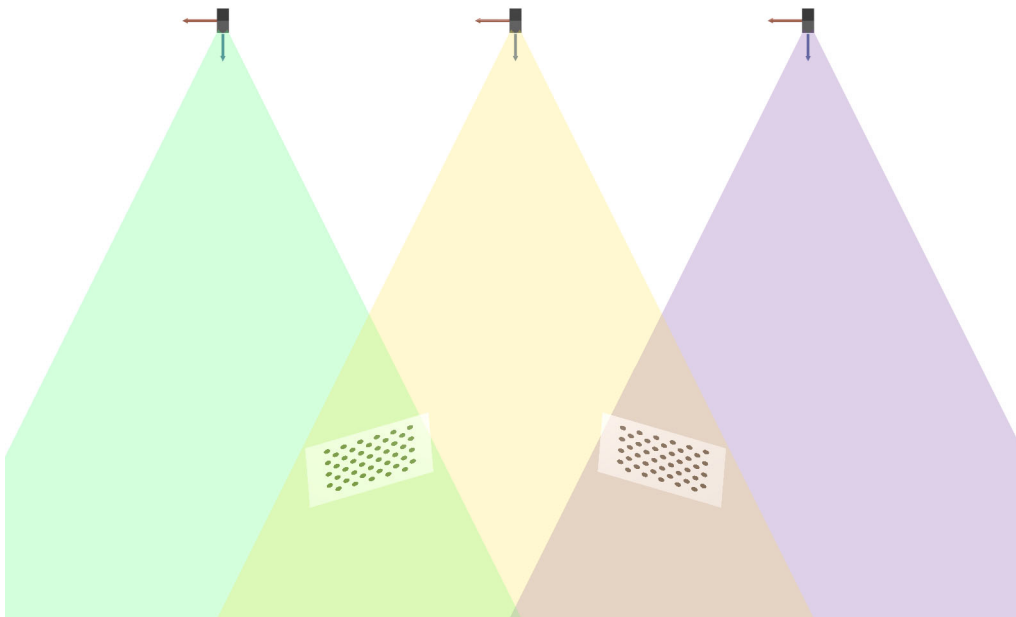


Figure 3.3: When using multiple cameras, they are stereo-calibrated in pairs. In order to do so, their frustums need to overlap so that two cameras are able to capture the calibration pattern at the same time. A tracked pose can then be transformed into the coordinate frame of a specified reference camera.

sake of simplification, the cameras are arranged sequentially. However, this is not

a prerequisite; it is essential, though, to ensure a path from each camera to the reference camera through overlapping fields of view.

A pose determined by one of the “more distant” cameras can be transformed into the coordinate system of a specified reference camera via the intermediate cameras. All tracking data, i. e., quadrotor poses and skeleton joint positions, are stated with respect to the coordinate system of that reference camera.

The result of the calibration process are the intrinsic camera parameters K_i as well the extrinsic parameters in form of transformation matrices that perform a change of basis from the coordinate system of the first camera of a stereo pair to the pair’s second camera’s coordinate system.

The sensor used for human pose estimation has to be registered within the frame of one of the vision cameras to reference skeleton data from the common coordinate frame $\{r\}$. Referencing skeletal joints in the same coordinate frame as the drones simplifies computations such as determining which drone an operator is pointing at. Depending on the type of sensor used, calibration and registration steps may differ. There are readily available neural networks that work with regular cameras, where the calibration would look similar to the one described above.

3.1.2 *Quadrotor Pose Estimation*

Poses of objects in space can be estimated by their shape, color, or by attaching passive or active markers. Passive markers, like the binary square fiducial markers that are part of the ArUco marker detection [RMM18] can be printed with a regular printer yielding camera pose estimation results with minimal effort. However, due to motion blur caused by fast-moving objects such as quadrotors, and reflections on the printed markers resulting from the attempt to achieve uniform illumination in the room, this method quickly encounters its limitations. These issues can be addressed by utilizing retroreflective markers, similar to systems like Vicon or OptiTrack, which requires cameras that emit infrared light synchronized with the camera exposure. Alternatively, regular cameras can capture markers consisting of infrared light-emitting diodes (LEDs), as demonstrated by the method employed in the *ICARUS* testbed, outlined in the following.

When tracking quadrotors, the key requirements are accuracy, frequency, robustness to rapid motion, and the ability to distinguish multiple targets. *High-speed and robust monocular tracking* [Tja+15] meets all these requirements while being very lightweight and cost-effective compared to multi-camera motion capture systems used in other testbeds. One of its significant advantages is that it does not rely on frame-to-frame strategies. Determining poses in individual frames ensures fast relocalization after temporary tracking losses, which is also essential in a UAV environment. The pose estimation system is a monocular outside-in pose estimation approach based on active infrared LED markers. The geometry of the refined marker design [Tja19] that is used for the quadrotors of the *ICARUS* testbed, is shown in Figure 3.4. Different markers are distinguished by the cross-ratio [HZ03] of the four angles between LED 1 and its coplanar LEDs 2–5.

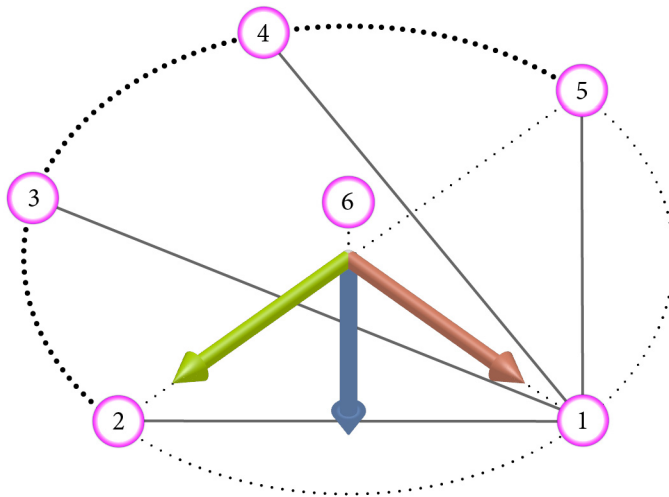


Figure 3.4: Geometry of the circular pose estimation marker pattern and its coordinate frame. LEDs 1–5 are coplanar and arranged in a circular fashion. LED 6 is slightly elevated at the center. Gray lines connecting coplanar LEDs are copunctal in LED 1. The angles between them encode a cross-ratio used to differentiate multiple markers. Consequently, LEDs 3 and 4 can be repositioned along the semicircle (thick) between LEDs 2 and 5.

At its core, the monocular tracking method uses the *perspective- n -point* (PnP) algorithm [Zhe+13] that estimates the pose of a calibrated camera by finding correspondences between n 2D image-projections of 3D world-points. The object is represented by an abstract model consisting of 3D points. An initial hypothesis of a transformation projects these model points onto the image plane. The results are then compared to the measurements from the markers of the physical object.

A pose estimate can be optimized by minimizing a least-squares problem where the energy function describes the pose fit as the sum of the squared distances between the projected and the measured 2D points. Pose ambiguities, common in preceding point-based pose estimation approaches typically manifest as “pose flips”. This issue is resolved by the tracking system’s pose optimization strategy, which leverages the spatial structure introduced by a raised LED at the center.

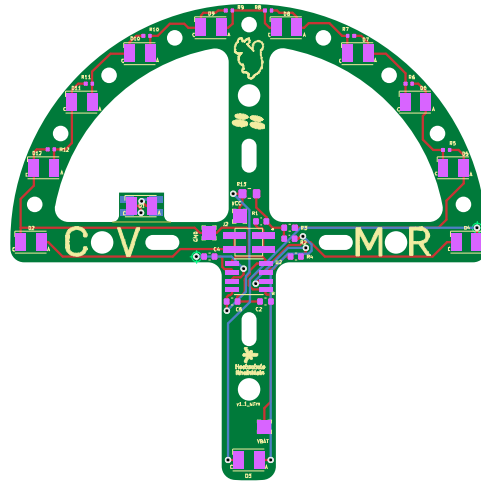
The circular marker pattern was integrated into a specifically designed PCB for the use with quadrotors. The PCB and schematics are shown in Figure 3.5. It uses a MAX682 charge pump regulator that generates 5 V from a 2.7–5.5 V input to match the voltage range 3.2–4.2 V of a standard single cell LiPo battery. This way the marker can be attached to the battery pins (VCOM) of the quadrotors. A marker-PCB attached to a Crazyflie 2.1 quadrotor is shown in Figure 3.6. The LEDs used are OSRAM SFH 4253-R emitters, that have their centroid wavelength at 850 nm. Matching bandpass filters (MidOpt FIL BN850/27) are attached to the lenses of the used vision cameras after the initial calibration process, that requires unfiltered wavelengths.

The pose of a marker $\{m\}$ with respect to the camera frame $\{c\}$ that captured it, is represented by the rigid-body transformation

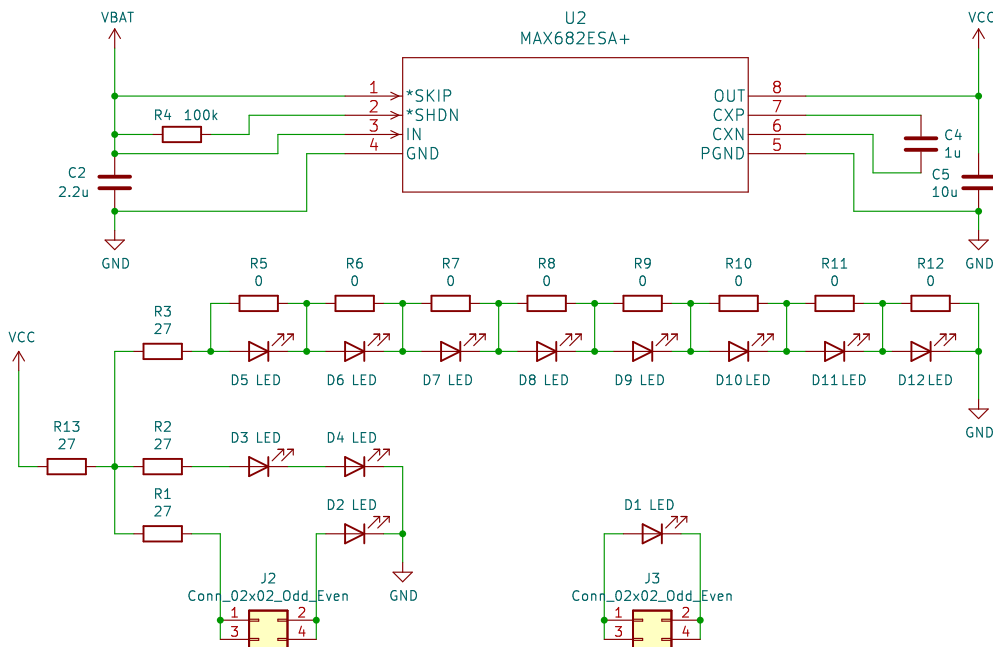
$$T_{cm} = \begin{bmatrix} R_{cm} & t_{cm} \\ 0 & 1 \end{bmatrix} \in SE(3).$$

As seen in Figure 3.4, the x axis of the coordinate frame points toward the single LED, situated at the end of the PCB’s single leg. The z axis points “into” the marker, and the direction of the y axis follows the right-hand rule (Figure 2.3). When attached to a quadrotor in \times configuration, as shown in Figure 3.6, the result is that the x axis of the marker points 45° off the desired flight direction. To track a coordinate frame different from the one pinpointed by the marker, a local offset transformation T_{mq} can be incorporated. This is desired to obtain poses that can directly be used for quadrotor calculations such as control. T_{mq} transforms the marker frame $\{m\}$ to the desired quadrotor frame $\{q\}$. Intricate in-flight fine calibration steps [Lup+14] can optimize this offset. But as the marker is firmly attached and aligned with the quadrotor’s frame, similar steps are dispensed with.

To receive poses with respect to a frame other than the camera, e. g., an arbitrarily chosen reference frame $\{r\}$, a global offset T_{rc} is incorporated into the transforma-



(a) True-to-scale, unpopulated tracking marker PCB.



(b) Tracking marker schematic.

Figure 3.5: Printed circuit board and schematic of the tracking marker specifically designed to fit quadrotors. In the manufacturing process, all LEDs are populated. However, only two of the LEDs from D5 to D12 are designed to emit light. Therefore, all unpopulated zero-ohm links R5 to R12, except for two, need to be manually bridged, effectively disabling six LEDs. LED D1 is initially attached to the main PCB for manufacturing purposes, but is intended to be detached and manually installed at the center of the marker.

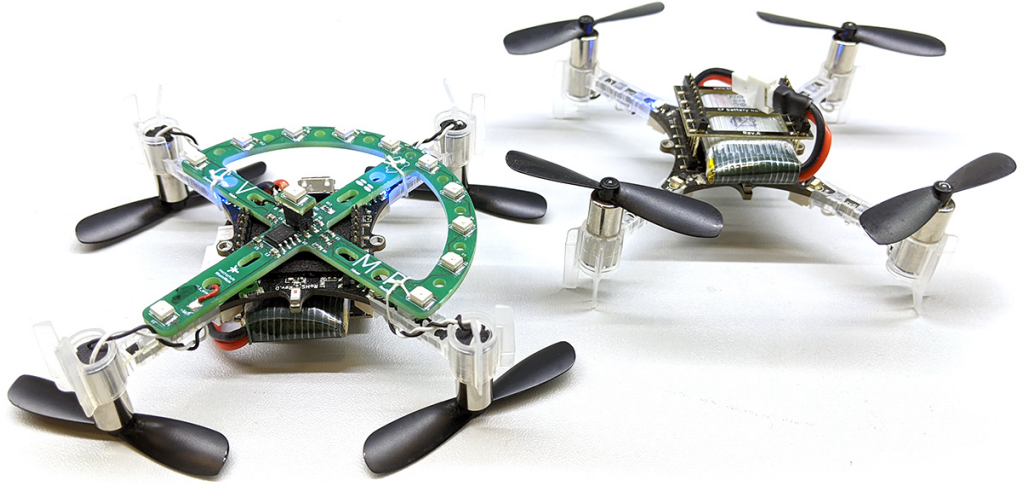


Figure 3.6: In the foreground, a Bitcraze Crazyflie 2.1 is shown with its rotors mounted upside down to avoid obscuring the LEDs of the tracking marker. The background shows an unmodified platform for comparison.

tion chain. For a quadrotor $\{q\}$ with attached marker $\{m\}$ tracked by a camera $\{c\}$, this results in the transformation

$$T_{rq} = T_{rc} \cdot T_{cm} \cdot T_{mq}.$$

This completes the entire pose estimation chain of transformations from a local frame $\{q\}$ to a global frame of reference $\{r\}$, as illustrated in Figure 3.7.

3.1.3 Reference Frame Definition

The definition of the common reference frame $\{r\}$ is divided into two steps: specification of the ground plane by collecting tracking positions on it and two further positions to describe the origin and the direction of the x axis. Both parts are carried out using one of the quadrotors, or more precisely their marker, which was described in the previous section.

After initiation of the ground plane definition, a quadrotor is manually moved across the floor. Position samples are collected throughout a predefined time window. This part of the procedure defines the reference coordinate system's up

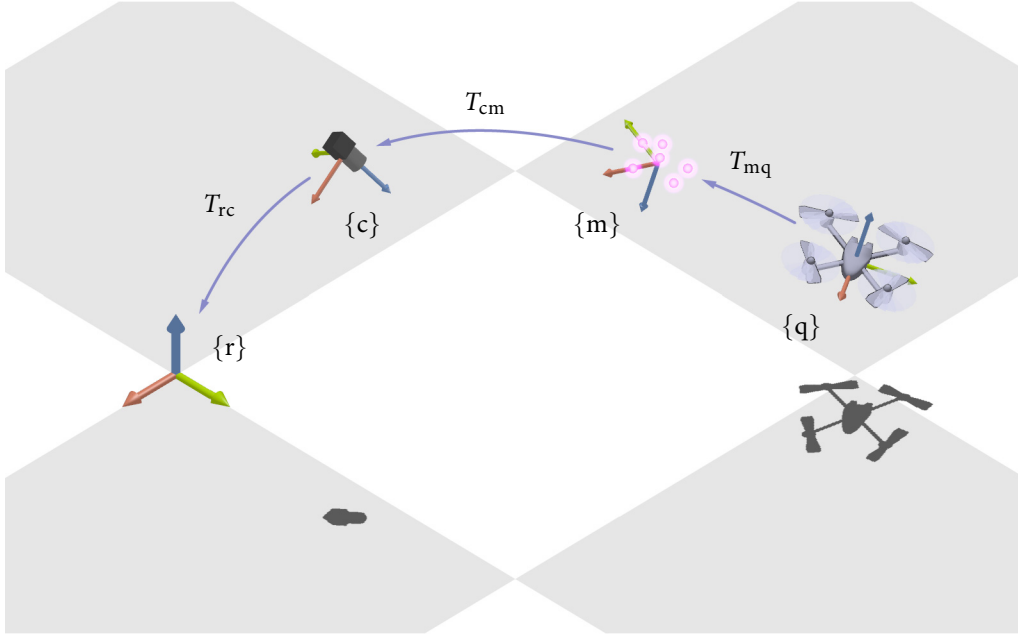


Figure 3.7: The complete pose estimation chain of transformations from the local frame $\{q\}$ to the marker $\{m\}$, then to the camera $\{c\}$, and finally to the global frame of reference $\{r\}$.

vector $\hat{\mathbf{z}}$, which is the normal vector of a plane that best fits the collected sample points. This normal is given by the eigenvector corresponding to the smallest eigenvalue of the covariance matrix of the collected samples. Since the cameras are assumed to be located in the upper hemisphere facing downward (and thus are their z axes), the direction of the determined plane normal may require correction. If the normal vector of the plane $\hat{\mathbf{z}}$ is pointing toward the same hemisphere as the camera's z axis $\hat{\mathbf{e}}_2 = [0, 0, 1]^\top$, its direction is flipped:

$$\hat{\mathbf{z}} = \begin{cases} -\hat{\mathbf{z}}, & \text{if } \hat{\mathbf{z}}^\top \hat{\mathbf{e}}_2 > 0 \\ \hat{\mathbf{z}}, & \text{else.} \end{cases}$$

For the further course of the procedure, two additional points need to be manually defined on the ground plane: point \mathbf{a} that defines the origin of the coordinate system $\{r\}$, and point \mathbf{b} that specifies the direction of the $\hat{\mathbf{x}}_r$ axis. The cross product of the ground plane's previously defined up vector $\hat{\mathbf{z}}$ and the preliminary x direction $\mathbf{x}' = \mathbf{b} - \mathbf{a}$ yields the coordinate system's y axis

$$\hat{\mathbf{y}} = \frac{\hat{\mathbf{z}} \times \mathbf{x}'}{\|\hat{\mathbf{z}} \times \mathbf{x}'\|}.$$

Its x axis is then obtained by orthogonalization:

$$\hat{\mathbf{x}} = \hat{\mathbf{y}} \times \hat{\mathbf{z}}.$$

The transformation from the reference tracking frame $\{c_1\}$ to the global reference frame $\{r\}$ is then given by

$$T_{rc_0} = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} & \mathbf{a} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

3.1.4 Pose Averaging

When using multiple cameras to expand the flight volume, it is necessary for all cameras to have overlapping view frustums. This enables pairwise stereo-calibration of all cameras into a single reference camera coordinate system. There are several approaches for handling the scenario where the same quadrotor is seen by multiple cameras simultaneously. One straightforward method involves having the poses of a single marker observed by multiple cameras within a common reference frame. In this case, pose estimation would be initialized for each camera based on its transformation relative to a reference camera, allowing the poses to be directly averaged.

However, it is important to note that errors in optical pose estimation tend to be larger near the edges of the camera image due to lens distortions. This may result in divergences of the estimated pose as seen from different cameras in the overlapping areas, causing peaks in the numerically derived velocity of the quadrotor and thus instable trajectory tracking. To address these issues and improve overall tracking accuracy, one approach is to incorporate bundle adjustment [LA09] as a final step in the camera calibration process to obtain optimal estimates. However, the simple averaging approach described in the following is a very intuitive fix resulting in smooth flights.

In monocular pose estimation, the out-of-plane transformations, i. e., the translation along the optical axis (depth) and rotations of the object out of the image plane are particularly prone to errors. These parameters are generally estimated with significantly lower accuracy than in-plane translations and rotations, which are more directly correlated with movement in the image plane. If a marker is seen by several cameras at the same time, it is therefore reasonable not to simply

calculate an arithmetic mean, but to take the distance of the marker to the camera into account. Poses that were tracked closer to an individual camera thus have more influence on the resulting, averaged pose.

For averaging and smoothing, the pose representation is changed from matrix representation to vector and quaternion representation. The translation vectors $\mathbf{p}_i(t) \in \mathbb{R}^3$ and the attitude unit quaternions $\mathbf{q}_i(t) \in \mathbb{S}^3$ determined by camera $\{c_i\}$ at time t are then processed independently. They are given with respect to the reference coordinate system $\{r\}$. For better readability, the coordinate frame is omitted, as well as the time t since all variables refer to the same point in time.

For n cameras that detected the individual marker, the weighted average translation vector is given by

$$\bar{\mathbf{p}} = \sum_{i=1}^n w_i \mathbf{p}_i. \quad (3.1)$$

The orientation is averaged by determining the 4×4 matrix M as described in Section 2.1.2:

$$M = \sum_{i=1}^n w_i \mathbf{q}_i \mathbf{q}_i^\top.$$

The inverse distance-weighted average attitude is then the normalized eigenvector corresponding to the maximum eigenvalue of M :

$$\bar{\mathbf{q}} = \arg \max_{\mathbf{q} \in \mathbb{S}^3} \mathbf{q}^\top M \mathbf{q}. \quad (3.2)$$

The average translation and orientation is weighted based on the marker's distance z_i to the individual camera $\{c_i\}$ that tracked the marker. The weights w_i are calculated by

$$w_i = \frac{z_i^{-1}}{\sum z_j^{-1}}, \quad i, j \in \{1, \dots, n\},$$

ensuring that they sum up to one. The distance is inverted so that poses closer to a camera are assigned higher weights in the averaging process.

3.1.5 Pose Smoothing

Due to sensor noise and jitter, poses tracked by the pose estimation system need to be smoothed over time. This is especially important for quadrotor control,

as, in addition to position and orientation, the control methods also require an estimate of the quadrotor's translational and angular velocity. These are determined by taking the time derivatives of the position and attitude tracked by the pose estimation system. Large differences in two successive translations and attitudes result in large derivations around the actual translational and angular velocity.

The choice of the smoothing method is very important, since smoothing always introduces lag. This is especially inconvenient when used for controlling agile robotic systems like quadrotors. For this reason, a suitable smoothing technique needs to place declining weight on older data to emphasize more recent poses. A straightforward method fulfilling this requirement is the exponentially weighted moving average (EWMA) [Per11] and can be implemented as a first-order filter:

$$s_t = \begin{cases} x_t, & t = 0 \\ \alpha x_t + (1 - \alpha)s_{t-1}, & t > 0. \end{cases}$$

It is a weighted filter that combines the previous smoothed statistic s_{t-1} with the current observation x_t using a smoothness factor $\alpha \in [0, 1]$. The tuning parameter of an exponential smoothing average is the time constant τ , that relates to α by

$$\alpha = 1 - e^{-\frac{\Delta t}{\tau}},$$

where Δt is the discrete sampling time interval. For poses, this operation is applied individually to the translational and rotational parts. The previously described inverse distance-averaged translation vector $\bar{\mathbf{p}}(t)$ is smoothed by linear interpolation, while the attitude quaternion $\bar{\mathbf{q}}(t)$ is smoothed utilizing *slerp*.

3.1.6 Trajectory Visibility Check

To limit target points and trajectories of the quadrotor to poses that lie within the visible area of the tracking cameras, a point clipping approach is used. A quadrotor target position or trajectory position \mathbf{x} is given with respect to the global reference frame $\{\mathbf{r}\}$. The homogeneous 3D position of the desired target pose $\tilde{\mathbf{x}}_{\mathbf{r}} = [x_1, x_2, x_3, 1]^T$ is first transformed into the camera frames of the individual cameras $\{\mathbf{c}_i\}$ by

$$\tilde{\mathbf{x}}_{\mathbf{c}_i} = T_{\mathbf{c}_i \mathbf{c}_i}^{-1} T_{\mathbf{r} \mathbf{c}_i}^{-1} \tilde{\mathbf{x}}_{\mathbf{r}}.$$

To adjust for the marker size and an additional safety margin, a 2D axis-aligned bounding box lying in the camera's xy plane is introduced and defined by the corner points

$$\begin{aligned}\tilde{\mathbf{a}} &= \tilde{\mathbf{x}}_{c_i} - \tilde{\mathbf{d}}, \\ \tilde{\mathbf{b}} &= \tilde{\mathbf{x}}_{c_i} + \tilde{\mathbf{d}},\end{aligned}$$

with bounding box half-size vector $\tilde{\mathbf{d}} = [d, d, 0, 0]^\top$.

Both points $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ are then transformed into the *canonical view volume* so that visible points lie within the range $[-1, 1]^3$. This is done by the perspective transformation matrix

$$T_{\text{persp}} = \begin{bmatrix} \frac{2f_x}{w} & 0 & \frac{2c_x}{w} - 1 & 0 \\ 0 & \frac{2f_y}{h} & \frac{2c_y}{h} - 1 & 0 \\ 0 & 0 & \frac{z_f + z_n}{z_f - z_n} & \frac{2z_f z_n}{z_n - z_f} \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

This matrix is build from the entries of the individual 3×3 camera matrix

$$K_i = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

with the 2D focal length $\mathbf{f} = [f_x, f_y]^\top$ and the 2D principal point $\mathbf{c} = [c_x, c_y]^\top$. The near and far planes z_n and z_f are user-defined and currently set to 1 m and 5 m, respectively, to ensure stable pose estimates; w and h are the pixel dimensions of the individual camera sensor.

A homogeneous point with clip coordinates $[x, y, z, w]^\top$ lies in the canonical view volume (and is thus visible for the camera) if the following conditions are true:

$$\underbrace{(w + x) > 0}_{\text{left}} \wedge \underbrace{(w - x) > 0}_{\text{right}} \wedge \underbrace{(w + y) > 0}_{\text{bottom}} \wedge \underbrace{(w - y) > 0}_{\text{top}} \wedge \underbrace{(w + z) > 0}_{\text{near}} \wedge \underbrace{(w - z) > 0}_{\text{far}}.$$

The target position in question \mathbf{x} is considered approachable if these conditions are met for both the clip coordinates $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ of the bounding box. In case of trajectories, each of their points are tested with this method. Flights that do not pass this test are not initiated.

3.2 RADIO CONTROL

In a regular handheld radio control, the stick positions are picked up by potentiometers or Hall effect sensors. These are converted into electrical signals with the other inputs such as switches and levers, modulated to a high-frequency signal and transmitted. The onboard receiver of the quadrotor, or any RC vehicle, demodulates the signal and converts it back to a proportional electrical signal controlling actuators such as brushed motors, the ESCs controlling brushless motors, or simply servo motors.

In order to substitute the handheld radio remote control system by a computer, drone development platforms offer corresponding USB hardware. In general, drone development platforms come with their own software development kit (SDK). Program code and the drone can communicate, e. g., via Wi-Fi or radio dongles. Alternatively, but more sophisticated, a microcontroller can generate proprietary signals and control transmitter modules from inexpensive hobby radios or generate a so-called trainer port signal that controls a professional handheld radio remote control. These three approaches implemented into *ICARUS* are described in the following.

For both the microcontroller approaches, i. e., the bare transmitter module and the trainer port solution, an Arduino Due is used. The Arduino Due operates at 3.3 V that matches the operating voltage of the transmitter module as well as the trainer port signal amplitude. The microcontroller is connected via USB to the computer that controls the quadrotors. The testbed software and the microcontroller exchange messages via serial communication. The transmitter module and trainer port approaches thus both offer cost-effective solutions for computer-controlling quadrotors from the hobby area or other RC vehicles. The current hardware for controlling the quadrotors utilizes Spektrum DSMX modulation, which is a popular standard in the model flight community. However, it is possible to use alternative hobby standards, such as Graupner HoTT, FrSky ACCST, TBS Crossfire, or the open-source ExpressLRS, by replacing the existing hardware with compatible devices. Some of the hobby drones that were previously used within the testbed are shown in Figure 3.8.

Wiring and hardware of the transmitter module approach inside its 3D-printed housing are shown in Figure 3.9. The hardware design supports up to three transmitters operating simultaneously, sourced from budget Spektrum radio transmitters,

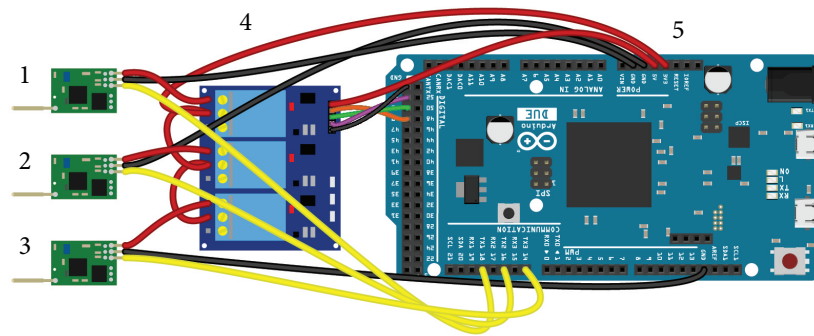


Figure 3.8: Initially, off-the-shelf hobby drones were tracked and controlled in the *ICARUS* testbed using prototype tracking markers.

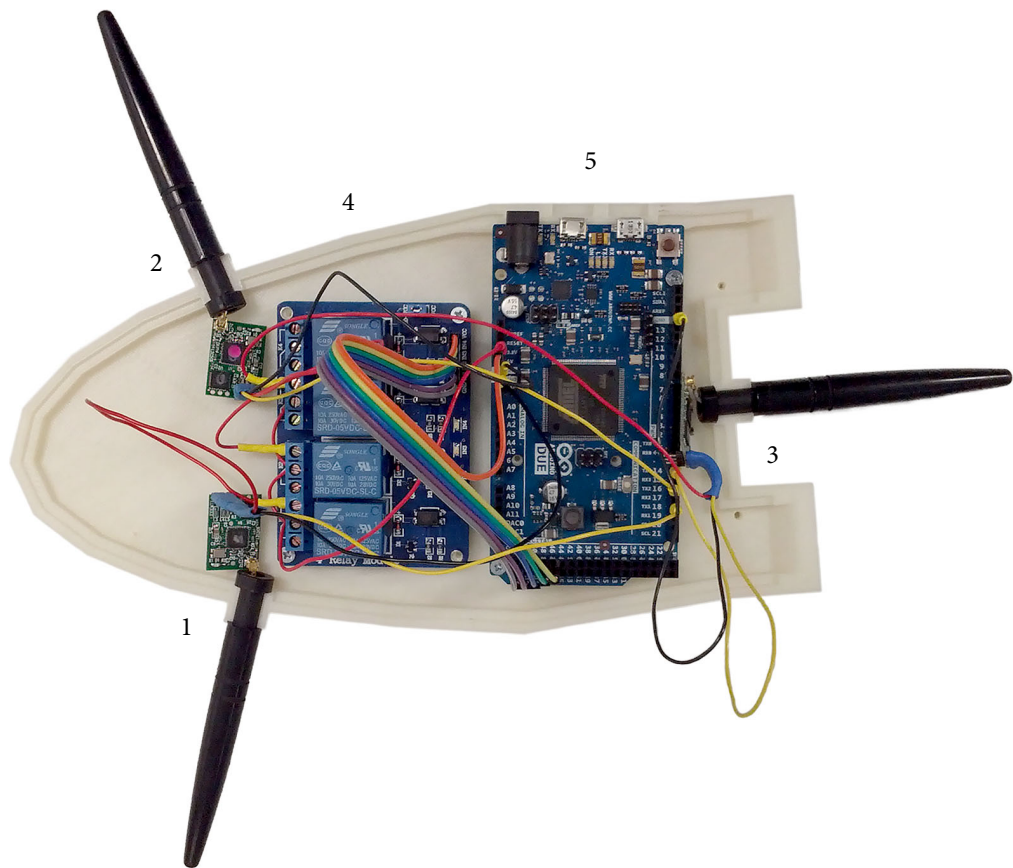
and communicates with them via serial communication. In addition to the USB interface, the Arduino Due also features three extra serial pins that enable the control of up to three transmitter modules simultaneously.

To bind a module to the receiver of a quadrotor, the binding signal must be applied before the transmitter module is switched on. The required current of 30 mA per transmitter module exceeds the current supplied by the Arduino's pins, so an additional circuit based on a transistor or relay is required to switch the units on and off. This approach simply distributes the signals received from the computer to the connected transmitter modules.

The trainer port solution, however, is more sophisticated, but also more versatile. It employs the trainer port signal and thus can be used with any radio control system equipped with a trainer port. For maximum adaptability, the radio control



(a) Wiring of the transmitter module approach.



(b) Hardware components inside a 3D-printed housing.

Figure 3.9: Radio remote control with three DSMX transmitters (1-3) taken from commercially available remote controls, which can be switched on and off via relays (4) and are controlled by an Arduino Due microcontroller (5). The wiring is shown in (a), the hardware components in their housing are shown in (b).

offers a bay for exchangeable transmitter modules. Thus, manufacturer-dependent transmitter modules can be easily exchanged, so there is no restriction to a specific radio protocol, as with the module approach.

The trainer port signal employs pulse-position modulation (PPM) and its structure is shown in Figure 3.10. One cycle of a typical trainer port signal takes 22 ms and

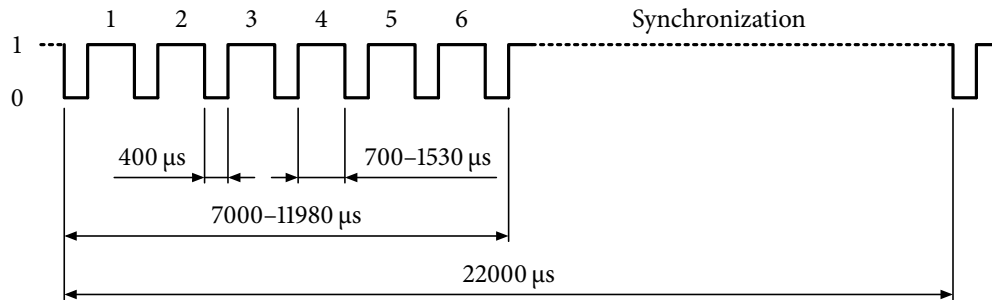


Figure 3.10: Characteristic of a six-channel trainer port signal that encodes throttle (1), roll (2), pitch (3), yaw (4), and two additional channels (5, 6) through pulse-position modulation.

encodes six channels: throttle, roll, pitch, yaw, and two additional channels (e. g., for switching flight modes). The duration of the high-level signal of a channel varies depending on the position of the corresponding stick and lasts for a period of 700–1530 μs. To encode all channels, a period of 7000–11980 μs is required, taking into account the 400 μs low-level signal that separates each channel. Until the beginning of the next period, a high-level signal is sent.

A total of 12 B (2 B per channel) are required to encode the complete control information within the signal. Limited by the maximum buffer size of 64 B of the serial interface, up to five control units can be driven simultaneously by a single Arduino Due. A pulse-position modulated signal is typically created by using sleep system calls. Due to the synchronized fashion of their implementation, these calls cannot be utilized to generate multiple signals in parallel on a single-core architecture like Arduino. To solve this issue, all five signals are discretized. In the synchronization phase of the trainer port signal, a byte pattern mirroring the current signal states is created and then transferred to the Arduino's port registers at once. Due to the time required for processing, all radio units can be controlled with a temporal accuracy of ± 2 ms. This has proven to be sufficient for the control purposes of *ICARUS*, especially since it is a cost-effective solution.

3.3 QUADROTOR SYSTEM IDENTIFICATION

With the motivation to reduce costs to a minimum, challenges arise. Cheap hobby drones are usually not as well equipped in terms of hardware as professional platforms. Moreover, most of the software is not modifiable or even open source. As a result, control tasks cannot be directly performed on the UAV. Instead, they have to be executed on a computer that remotely controls them. For the purpose of the infrastructure, in the context of developing and evaluating novel concepts of HDI, this is suitable as research and experiments are often conducted within the controlled indoor space.

Though quadrotors can be computer-controlled using basic PID controllers without the knowledge of the physical parameters of the platform used [Lie14], these parameters are essential to perform agile flights, to plan trajectories at the limit of the platform or to determine the feasibility of trajectories. As with any model of a real system, there are several levels of abstraction where the complexity increases with the degree of detail of the model. Where such details are not known or disclosed by companies manufacturing quadrotor parts, or for custom builds, these values can to be estimated as described in the following.

If a quadrotor should promptly lift from one height to another, e. g., to dodge an obstacle, the thrust it can generate by its combination of motors and rotors and its weight have to be known in order to plan the according trajectory. Since motors cannot physically reach a commanded speed instantaneously, time constants that determine how long it takes the motors to speed up or slow down can further detail the model. At even more detailed stages, the shape of the rotors can be taken into account. In this thesis, estimated coefficients suffice to enable precise flights. Lateral motion of the quadrotor depends (next to the motor and rotor characteristics) on its moments of inertia and distance of the motors to its center of gravity.

The quadrotor that is predominantly used in the testbed is the Bitcraze Crazyflie 2.1. Its model parameters that are used in the testbed implementation for trajectory generation, simulation, and control are summarized in Table 3.1. Parameters whose estimation is complex and requires expensive hardware have already been determined for this platform [För15]. Missing and additionally required parameters were determined in this thesis. Their estimation is described in the following.

Table 3.1: Crazyflie model parameters estimated in this thesis; colored parameters were estimated by Förster [För15].

Symbol	Values	Units	Quantity
m	0.034	kg	All-up weight (AUW)
l	0.046	m	Center-to-rotor distance
r_{\max}	416.667	s^{-1}	Max. motor revolutions
J_{xx}	16.571 710 $\times 10^{-6}$	$kg\ m^2$	} Moments of inertia about diagonal matrix entries
J_{yy}	16.655 602 $\times 10^{-6}$	$kg\ m^2$	
J_{zz}	29.261 652 $\times 10^{-6}$	$kg\ m^2$	
$\tau_{f\uparrow}$	0.091 35	s	} Motor time constants for spinning up and down
$\tau_{f\downarrow}$	0.348 65	s	
f_{\max}	0.635 471	N	Maximum thrust
$f_{4.0}$	0.579 573	N	Thrust at 4.0 V
k_0^f	-2.901 176 13 $\times 10^{-4}$	}	} Thrust-to-input coefficients
k_1^f	3.057 547 27 $\times 10^{-6}$		
k_2^f	1.116 090 70 $\times 10^{-10}$		
k_1^v	-0.454 091 71	}	} Voltage dependent thrust ratio coefficients
k_2^v	2.655 543 51		
k_0^k	1.563 383 000	}	} Thrust-to-torque coefficients
k_1^k	0.005 964 552		

3.3.1 Thrust Mapping

The control algorithms use the physical model of the quadrotor described in Section 2.3 and thus physical parameters, e. g., the target thrust. However, the throttle input of an RC model like a quadrotor is usually the percentage of the throttle stick deflection controlling the motor revolutions.

To determine the relationship between the throttle input and the resulting thrust, the quadrotor was connected to a laboratory power supply and attached to a “third hand”, that was placed on a laboratory scale, as seen in Figure 3.11. In professional applications, forces are typically quantified using precision load cells designed for accurate force measurement. However, in certain scenarios, forces can be

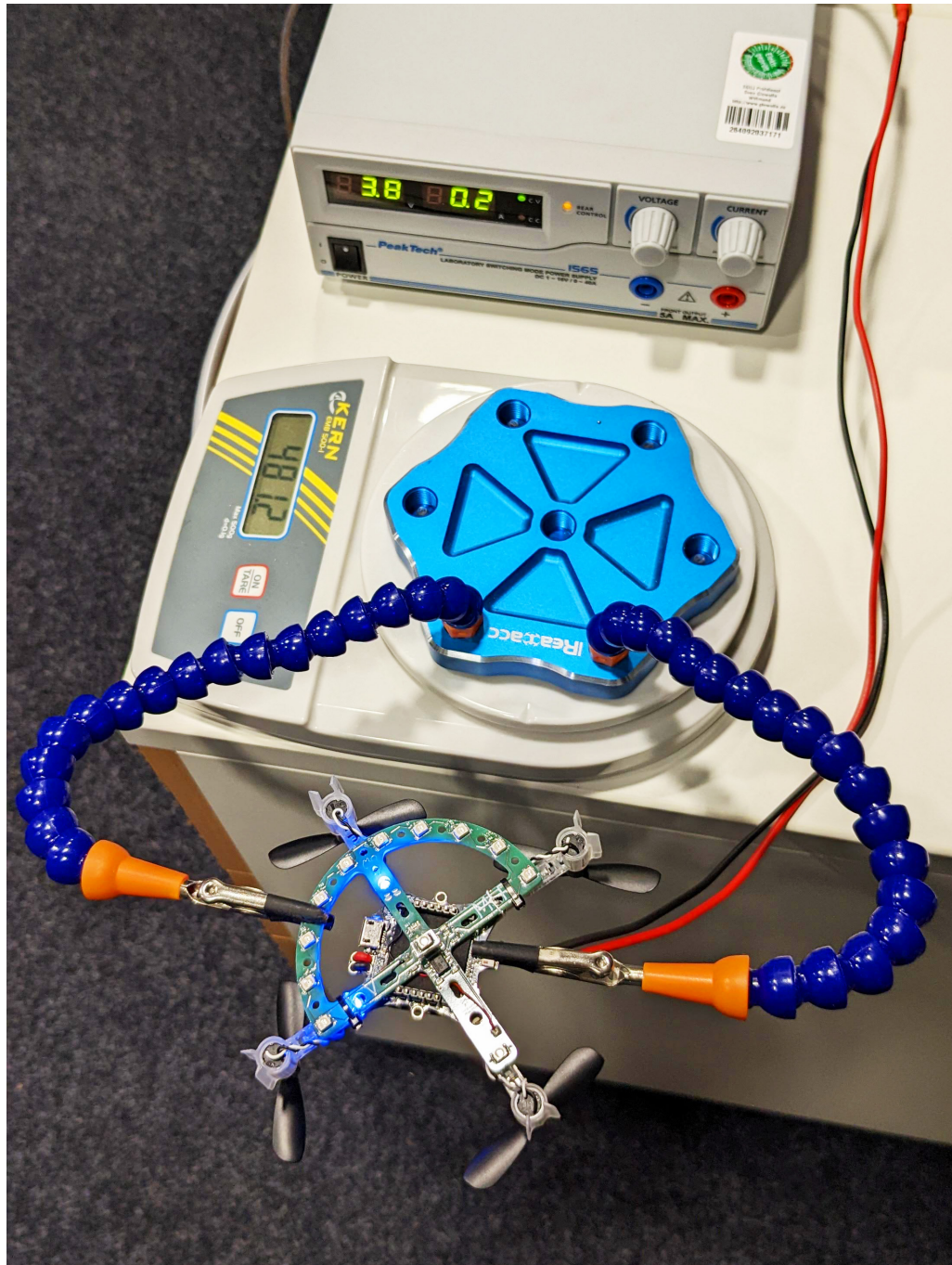


Figure 3.11: Crazyflie thrust experiment setup involving a laboratory power supply and balance. The drone is held in place by clamps of a third-hand soldering tool, with its arms extending beyond the table to minimize the influence of the ground effect.

reasonably approximated using laboratory scales. The results of the measurements using a Kern EMB 500-1 laboratory scale and the solutions to a fitted quadratic function are shown in Figure 3.12. Table 3.1 shows the parameters of the second order polynomial

$$f = k_2^f c_{\text{cmd}}^2 + k_1^f c_{\text{cmd}} + k_0^f, \quad (3.3)$$

where $c_{\text{cmd}} \in [0, 60000]$ is the Crazyflie throttle command and f is the collective force generated by the motors.

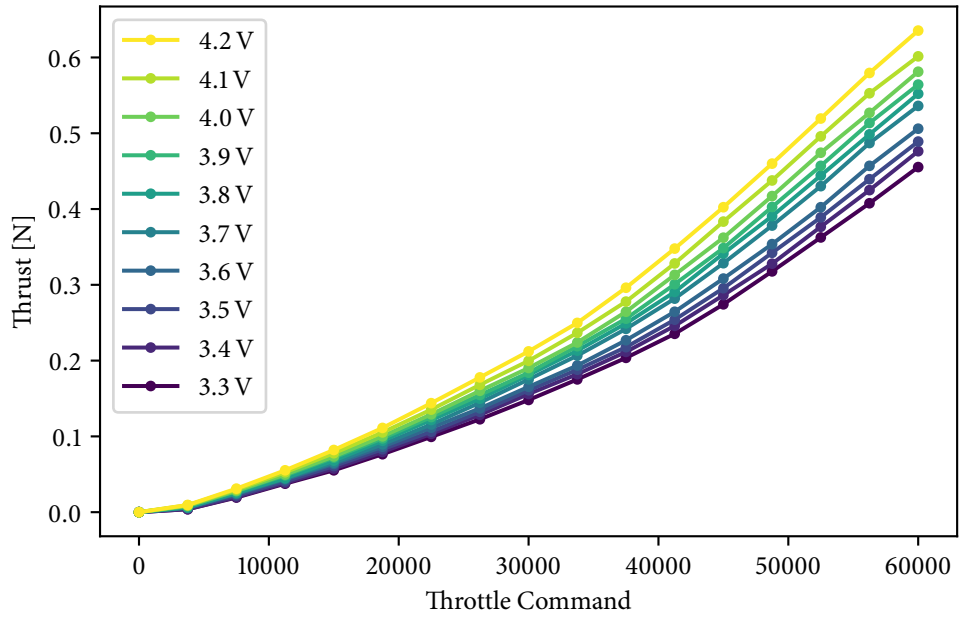
3.3.2 Battery Voltage Compensation

LiPo batteries have a characteristic voltage drop curve over the course of a discharge cycle. Since they are also subject to non-constant discharge rates, they *sag*, that is, the voltage drops temporarily, as current is charged. Since the voltage drops over the course of a flight and sags under load, e. g., full throttle results in more thrust at the beginning of a flight than at the end. With manual control, the pilot instinctively corrects the throttle lever position. Autonomous control however, calculates the desired thrust that is mapped to a drone-specific throttle command as described in the previous section. Since the rotor thrust depends on the current battery voltage, as a third step, this command needs to be compensated for the continuously dropping and temporarily sagging voltage.

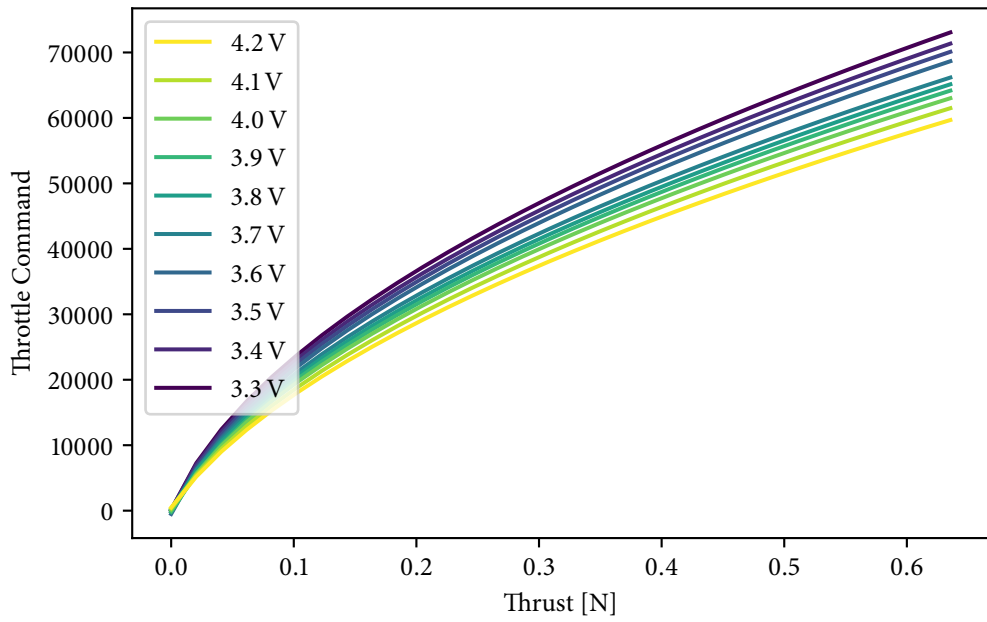
When applying the thrust-to-input-mapping at 4.0 V, which was identified in the previous section, the ratio of the commanded thrust to the actually produced thrust is a linear function of the battery voltage [FFS17; FFS18]. In order to calculate the coefficients of the linear function, an identification flight is performed, where the z acceleration and the commanded acceleration (mass-normalized thrust) are logged. For this flight, the integral part of the position controller is set to zero for the quadrotor height.

The ratio of the mass-normalized commanded thrust to the generated thrust measured by the onboard accelerometer of the identification flight along with the fitted linear function is shown in Figure 3.13. The identified coefficients are listed in Table 3.1 and are used to compensate the thrust command c_{cmd} resulting from (3.3) for the varying battery voltage by applying

$$c_{\text{cmd, comp}} = (k_1^v v_{\text{bat}} + k_2^v) c_{\text{cmd}},$$



(a) Values averaged over three measurements, in which the drone lifted the weight of a third-hand soldering tool from a laboratory scale.



(b) Quadratic polynomials fitted into the results of the Crazyflie thrust experiment.

Figure 3.12: Measurement data from the Crazyflie 2.1 thrust experiment (a) and polynomial fit (b).

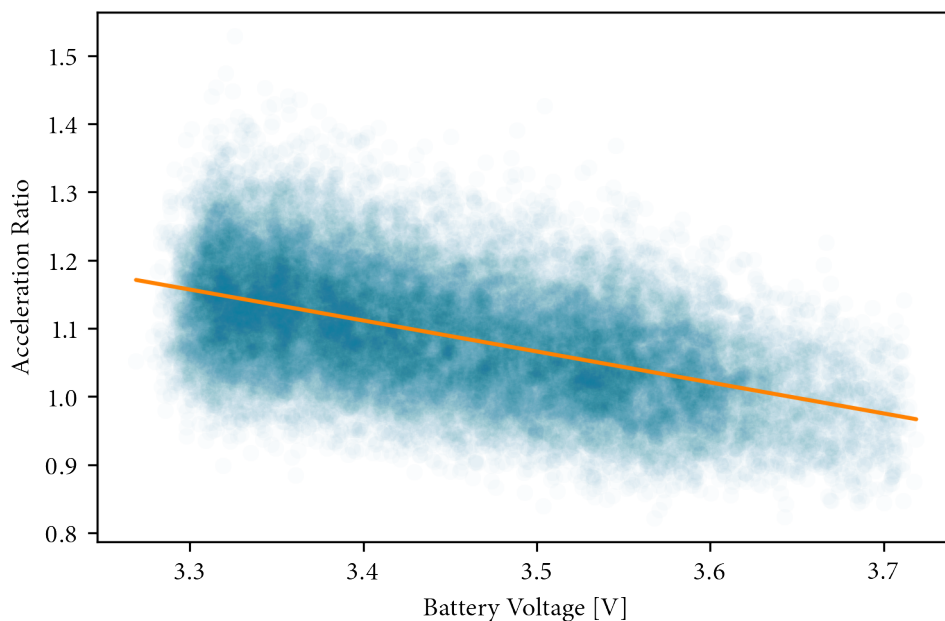


Figure 3.13: Battery voltage compensation identification: The data points (blue) represent the ratio of mass-normalized commanded thrust to actual thrust measured by the onboard accelerometer and were recorded during a 254 s hover flight. This ratio is approximated as a linear function (orange) of the battery voltage.

where v_{bat} is the current voltage of the battery, that is continuously transmitted by telemetry data. Height errors during a validation flight are shown in Figure 3.14. If high-accuracy height tracking is required, any remaining height errors can be compensated by an additional (integral) controller after applying the battery voltage compensation.

3.3.3 Motor Identification

The thrust generated by a rotor can be modeled as a first-order system, characterized by its time constant τ . The dynamical model of a quadrotor, as described in Section 2.3, requires two time constants: one for the force generated by the rotor spinning up, $\tau_{f\uparrow}$, and another for when it is spinning down, $\tau_{f\downarrow}$. In the following, an experimental approach is outlined, which acoustically determines both time constants for the force generated by the rotor during spin-up and spin-down phases, respectively.

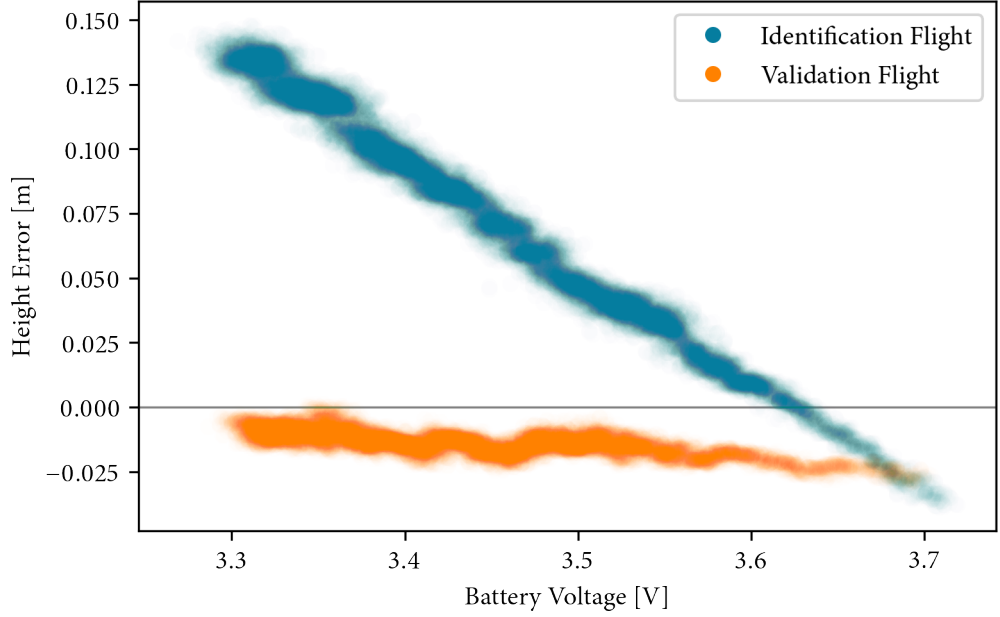


Figure 3.14: Battery voltage compensation validation: Data points represent the height error during the identification flight (in blue) and validation flight (in orange). The 194 s hover flight, with applied battery voltage compensation, demonstrates reduced and more stable height errors compared to the flight without battery voltage compensation.

The first-order system step response for a motor spinning up is given by

$$y_{\uparrow}(t) = \begin{cases} k(1 - e^{-\frac{t}{\tau_{f\uparrow}}}) & t \geq 0, \\ 0 & \text{else,} \end{cases} \quad (3.4)$$

a motor spinning down can be modeled by

$$y_{\downarrow}(t) = \begin{cases} k e^{-\frac{t}{\tau_{f\downarrow}}} & t \geq 0, \\ k & \text{else.} \end{cases} \quad (3.5)$$

The time constant $\tau_{f\uparrow}$ characterizes the time taken for the motor to reach $1 - 1/e \approx 0.632k$ of its steady-state value k in response to a step input. Returning from the unit step back to zero, the time constant $\tau_{f\downarrow}$ characterizes the system when spinning down, i. e., the time it takes to reach $1/e \approx 0.368k$ during step recovery.

To determine the duration it takes for the electric motor to drive its rotor to the maximum revolutions, where it generates maximum thrust, can be done acousti-

cally, as each rotor blade generates a measurable sound wave. The experimental setup to determine the time constants acoustically is shown in Figure 3.15.

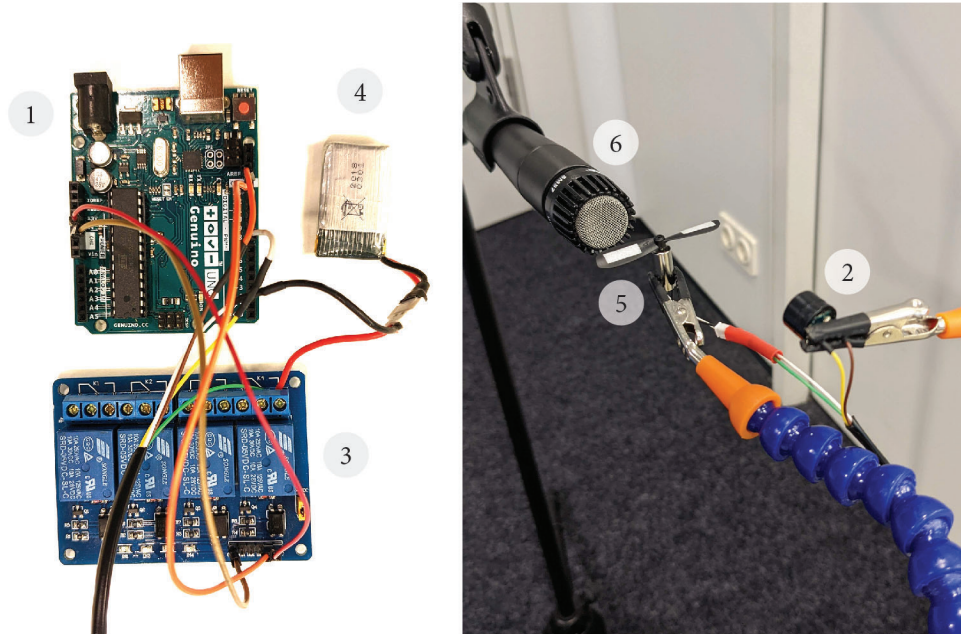


Figure 3.15: Motor time constant identification experiment: A microcontroller (1) first signals a beep through a piezo buffer (2) to ensure time synchronization before switching the relay (3). The relay allows current from a standard LiPo battery (4) to flow to a Crazyflie brushed DC motor (5), causing it to spin at its maximum speed. A second beep from the piezo marks the start of the step response recovery process, after which the relay is switched again. An instrument microphone (6) records the generated sound throughout the experiment.

For this experiment, an Arduino Uno microcontroller was equipped with a piezo speaker and a relay. To synchronize multiple repetitions, the speaker generated a short sound before switching the relay and thus supplying power from a common 4.2 V LiPo battery to the motor letting it spin at maximum speed. A Shure SM57 dynamic microphone captured the generated sound through a Focusrite Scarlett 2i2 audio interface providing a sampling rate of 192 kHz. The recorded audio signal was smoothed using a Savitzky-Golay filter [SG64] with a window size of 51 fitting samples using a third degree polynomial before further processing. In order to estimate the motor frequency, a short-time Fourier transform (STFT) with a window size of 0.02 s (3840 samples) was applied. For each time step in the STFT result, the frequency with the largest absolute magnitude was assumed to be the

rotor frequency at that time. The average maximum frequency between spin-up and spin-down phases determined k .

The first-order systems (3.4) and (3.5) are then independently fitted using the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno (BFGS) [NW06], minimizing the mean squared error. The final values for the time constants of the motor spinning up $\tau_{f\uparrow}$ and down $\tau_{f\downarrow}$, respectively, are the average values of 12 repetitions and are given in Table 3.1. The evaluation of a single measurement is shown in Figure 3.16.

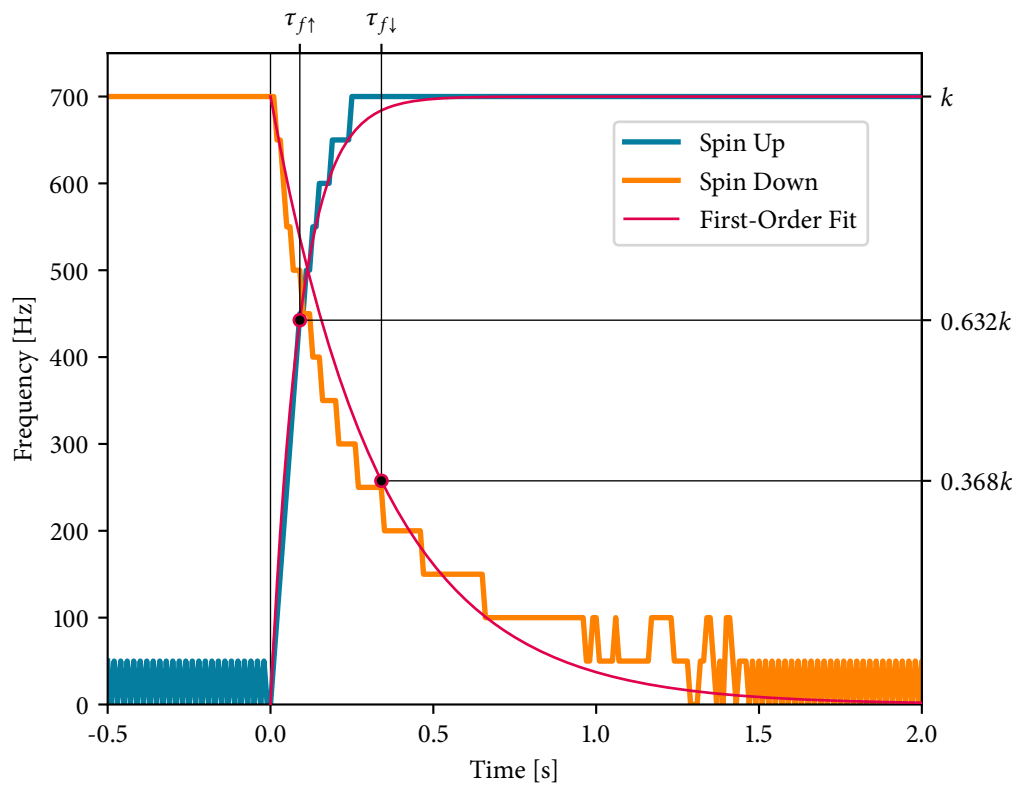


Figure 3.16: STFT of a single acoustic measurement of a Crazyflie brushed DC motor responding to the step inputs at $t = 0$ s from 0 V up to 4.2 V (blue) and from 4.2 V down to 0 V (orange). The resulting time constants for the motor spinning up and down were determined by linear approximation of the first-order fits (red) at $0.632k$ and $0.368k$, respectively.

3.4 IMPLEMENTATION

The individual software components of the *ICARUS* testbed are divided into layers, as shown in Figure 3.17. These components include the *Interface* for user interaction and the implementation of additional applications, the *Flight Control*, responsible for estimating poses and controlling trajectories, the *Simulation*, and *Reality*. After providing an overview, the remaining part of this section is structured based on these layers and describes their implementation and interaction in detail.

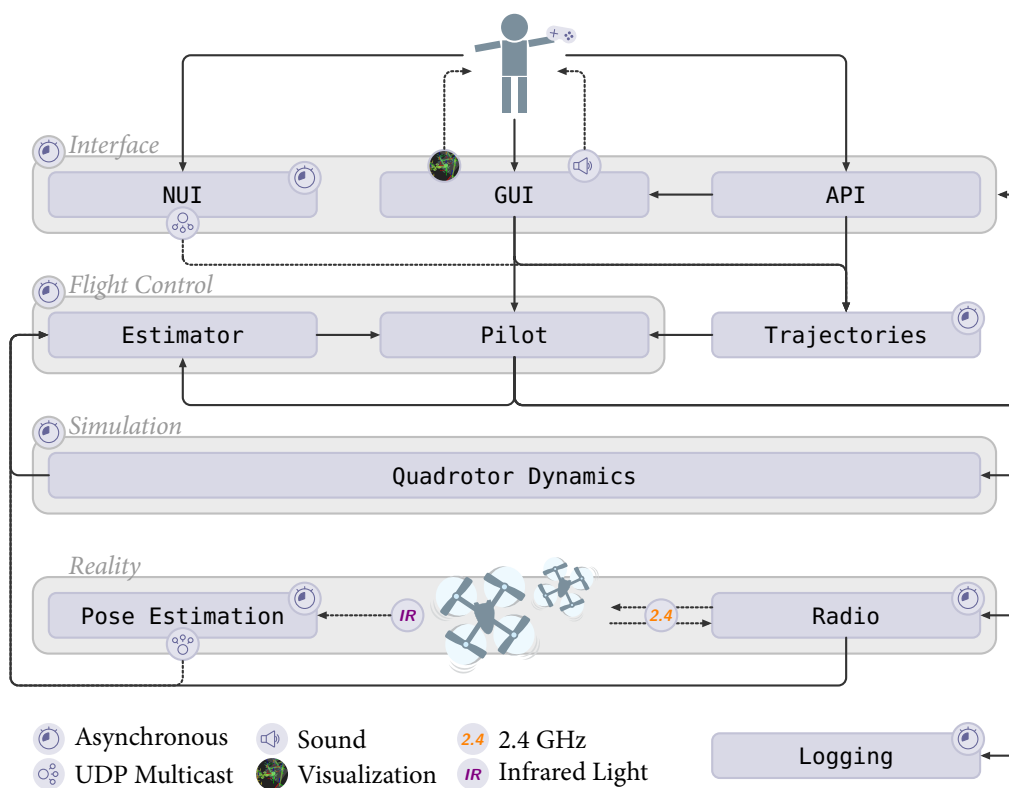


Figure 3.17: Data flow in *ICARUS*: The *Interface* layer offers classical and natural user interfaces for controlling and interacting with the testbed and its drones. It also encompasses software interfaces, allowing user code extensions and specialized applications to enhance the infrastructure's functionality. The *Flight Control* layer receives poses and calculates control inputs based on a given reference trajectory. These inputs are then passed to the radio control implementation. This can be a part of either the *Simulation* or *Reality* layers, where simulated or physical quadrotors are controlled. Critical data can be consolidated and logged through an additional component.

All software components of the testbed are platform-independent, except for the natural user interface (NUI) component, which, in the current implementation, is hardware-dependent on the depth camera, a Microsoft Kinect 2, which requires the Windows operating system. The software for the entire infrastructure, which includes pose estimation, is capable of running on a single, standard laptop. However, the pose estimation can be decoupled and run on a separate computer. The same applies to the software that implements the Kinect 2 skeleton tracking. Both components then broadcast user datagram protocol (UDP) messages to communicate with the core application.

All software components of the *ICARUS* testbed are developed in C++17. Individual components and several classes within the components run asynchronously using Boost `Asio` libraries. Communication between those components are realized using the Boost `Signals2` library implementation of a managed signals and slots system. Distributed components communicate through UDP messages utilizing the Boost `Network` libraries. For this purpose, data structures like poses are serialized with the Google `Protobuf` library. Throughout the software, for linear algebra, matrix and vector operations, the `Eigen3` library is used. A configurable range of data is consolidated in an asynchronous logging implementation for the purpose of evaluation and optimization. Logged data includes the drones' positions, linear velocities, accelerations, attitudes, and angular velocities. Additionally, it records battery voltages, the current control methods, and, if needed, skeleton joint positions.

3.4.1 *Interface*

The *Interface* layer incorporates a traditional graphical user interface (GUI) implemented using the Qt6 framework. It includes a 3D visualization of all the testbed components implemented with OpenGL and glm. A screenshot is shown in Figure 3.18. In addition to providing visual feedback, the user interface also incorporates synthesized rotor sound. This is achieved by recording the sound of a single rotor during hover and playing it back with spatial effects using the SFML library during the simulation. The pitch of each rotor's sound is adjusted based on the ratio of its revolutions to the revolutions required for hovering.

Physical and simulated quadrotors can be controlled through several approaches. Manual control through standard gaming joysticks or professional RC transmitters

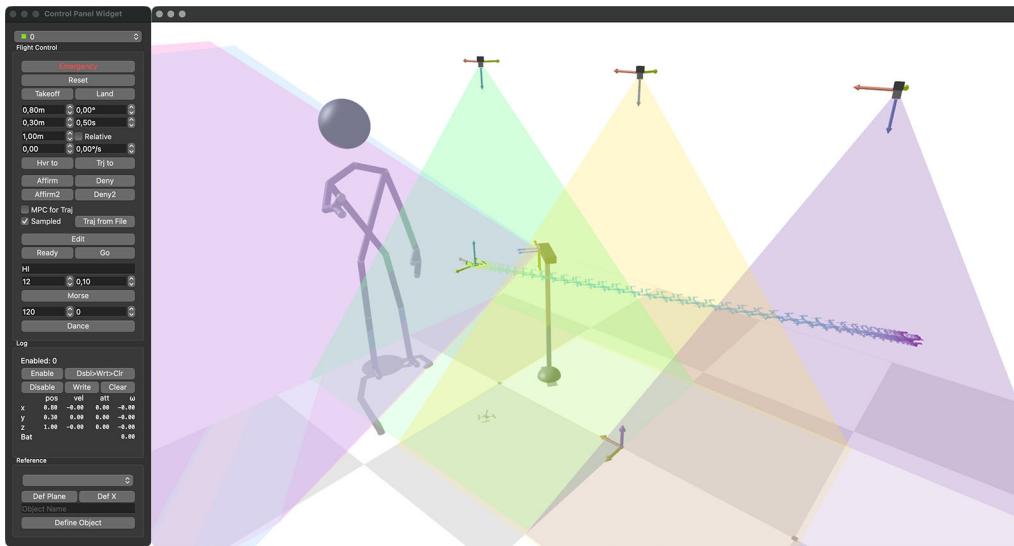


Figure 3.18: Control panel and 3D visualization of the testbed application.

is implemented with SDL. Input (“stick feeling”) can optionally be improved by so-called “PIDs” and “Rates”, where the desired angular rates (and sometimes throttle, too) are mapped as nonlinear, usually exponential functions and then applied to a PID controller before sending the signal to the drone or simulation.

Gesture-based NUIs are implemented using basic linear algebra and 3D skeletal joint positions, e. g., to determine whether a person is pointing at a drone. In the current implementation, skeleton tracking data is provided by a Microsoft Kinect 2 camera, the integration of which is described in Section 3.4.4. The Kinect 2 is an affordable time-of-flight (ToF) camera, that determines the distance to objects for each point in the camera image by measuring the round trip time of pulsing infrared LEDs. The official Kinect 2 SDK detects and provides the skeleton joints for up to six persons at a rate of 30 Hz. Since it runs exclusively on Microsoft Windows, a dedicated computer runs the skeleton tracking software implementing the Kinect 2 SDK, multicasting results as UDP messages. These messages contain all detected skeleton joint positions with respect to the Kinect’s depth frame $\{k_d\}$. Using the skeleton joint transformation (3.6), the main application transforms these joints into the global reference coordinate system $\{r\}$ that it defines. The information extracted from the skeleton joint positions is then further processed, e. g., to gesture-control quadrotors.

In order to create and visually experiment with different trajectory designs, the interface includes a trajectory editor with “keyframe” visualization. Keyframes act as boundary states for the quintic polynomial trajectories described in Section 2.3.2.

A screenshot of the user interface including the trajectory editor can be seen in Figure 3.19. Using the table view editor, lists of keyframes can be loaded, edited

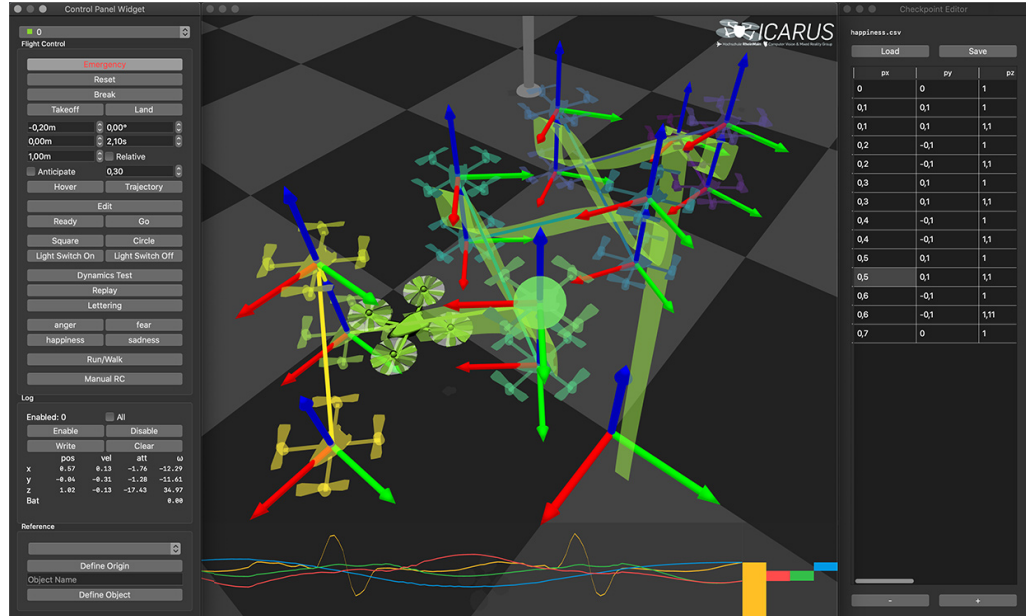


Figure 3.19: The user interface of ICARUS in manual trajectory design mode. The control panel is on the left side, the trajectory editor is on the right, and visualization in the center.

and saved. Keyframe positions can be moved directly in the visualization, velocity and acceleration can be edited in the table view if necessary.

From the interface layer, the existing infrastructure can also be extended by custom user code, like the applications and extensions described in Section 3.6.

3.4.2 Flight Control

A control loop is triggered by a pose update coming either from the simulation or from the physical pose estimation. Both the physical cameras as well as the simulation are triggered at a frequency of 100 Hz. In case of pose estimation, the

estimator receives the position vector and attitude quaternion already merged over multiple cameras and smoothed by an EWMA filter as described earlier in Sections 3.1.4 and 3.1.5. Linear and angular velocities are then determined by numerical differentiation and smoothed by a low-pass filter. The simulation calculates these values directly. These properties, i. e., position, attitude, velocity, and angular velocity, form the quadrotor state \mathbf{x}_k at time point k . The associated input is \mathbf{u}_k . Telemetry data from real quadrotors is also forwarded to the estimator and includes the current battery voltage to compensate the thrust command for varying voltages, which is not necessary using the simulation, as it does not include a LiPo battery model.

The flight control system employs a predictor-corrector estimator [Lup+14] for two primary purposes: first, to compensate for latency by accounting for the age of the pose at the time of control, and second, to predict the quadrotor's pose at the arrival of the next control signal at the drone's receiver. The age of the pose is the time difference between the timestamp of the pose and the time point of the current control step. The look ahead delta time can be approximated by half the round-trip time (RTT). The RTT is the duration of time required for a signal to be sent to the receiver and the time it takes for the sender to receive an acknowledgement. In case of the Crazyflie, the RTT is approximately 60 ms [För15]. The predicted pose will be the basis for the control, which will determine the control input \mathbf{u}_{k+1} for the future state \mathbf{x}_{k+1} . The second purpose of the predictor-estimator is to advance the prior estimation to compensate for any short-term tracking losses.

The quadrotor states determined by the estimator are used by the pilot implementation to hover in position or to track trajectories. The calculated control variables are transmitted to either the radio or simulation, and then relayed back to the estimator to prepare for the following control step. The position controller employed [MK11], as described earlier in Section 2.4.2, was tuned for the Crazyflie 2.1. The determined parameters are listed in Table 3.2.

For trajectory control, the MPC is set up with a discretization time step $dt = 0.1$ s and time horizon $t_h = 2$ s. The implementation is based on open source code [Fal+18] and uses ACADO, an open source toolkit for automatic control and dynamic optimization, to set up the optimization problem. It is constructed using multiple shooting as transcription method and a Runge-Kutta (RK) integration scheme. The implicit RK integrator of order 4 (RK4) is the most widely used integrator [RMD17]. The quadratic optimization problem is then solved using qpOASES, an open source quadratic programming (QP) solver, through a SQP.

Table 3.2: Hover controller tuning parameters used for the Crazyflie 2.1.

Symbol	Diagonal Matrix Entries	
	x, y	z
K_p	0.24	0.80
$K_{\dot{p}}$	0.18	0.30
K_i	0.10	0.01
K_R	5.20	7.00
K_ω	0.40	0.60

While the model parameters remained unchanged, the controller was experimentally tuned, resulting in deviations from the original implementation. A tolerated error in distance, e. g., 0.1 m, corresponds to a quadratic error cost of $1/(0.1^2) = 100$. For errors in attitude, e. g., a tolerated quaternion deviation of magnitude 0.2 (approximately 23°) corresponds to a quadratic error cost of $1/(0.2^2) = 25$. The costs for the horizontal and vertical position errors were maintained at 200 and 500 respectively, but the costs for the attitude error were decreased from 50 to 25. For the velocity error, they were decreased from 10 to 4. The costs for the thrust, roll and pitch inputs were retained at 1, while the costs for the yaw input was reduced from 1 to 0.1.

To ensure the safety of humans, including those involved in user studies, concurrent safety measures were implemented. Emergency mode is triggered if the quadrotor's autopilot has not received an updated pose in the past 0.2 s. It then immediately disarms the quadrotor, causing the rotors to stop spinning. As a result of this mechanism, quadrotors will also shut down when they leave the tracking volume. Additionally, if a quadrotor following a pre-planned trajectory comes within 0.3 m of a human's head, it will immediately stop mid-air and switch to hover mode. This behavior is contingent upon the Kinect system being operational and accurately providing real-time skeleton poses.

3.4.3 *Simulation*

The simulation is based on the quadrotor dynamics outlined in Section 2.3. It is implemented utilizing the `Eigen` library and supports normalized inputs for manual control, i. e., by a gamepad or radio, physical inputs (mass-normalized thrust and angular velocities) from the controller, and direct rotor revolutions, e. g., for machine learning approaches. It includes a wind gust model, which is used to roughly adjust the parameters of the controllers in order to enhance resistance to disturbances and improve its overall performance. The simulation also incorporates a basic collision model based on spatial hashing to detect collisions with surrounding geometry, which proves useful when the simulator is utilized for training regular flight maneuvers.

3.4.4 *Reality*

In the reality layer, the drones are remotely controlled by different radio approaches that can be used in parallel. Off-the-shelf (brushless) quadrotors are remote controlled by the Arduino-based serial remote control approaches described in Section 3.2. In this case, the physical radio used is a FrSky Taranis X9D radio system. A less complex setup is offered by the brushed Bitcraze Crazyflie 2.1 development platform, but it comes with shorter flight durations. To control the Crazyflie remotely and to receive telemetry data with the Crazyradio PA 2.4 GHz USB dongle, it is integrated into the infrastructure using the `crazyflie_cpp` library [HA17]. The Crazyflie itself runs the default firmware in rate mode with onboard battery compensation disabled, since this is done by the testbed implementation as described in Section 3.3.2.

The current setup of the infrastructure consists of three vision cameras for quadrotor pose estimation and a Kinect 2 depth sensor for human pose estimation. For the camera calibration process, i. e., the determination of intrinsic and extrinsic camera parameters, the `OpenCV`³ library is used. Their implementation requires a printed pattern attached to a planar surface. The same pattern can be used to estimate camera positions relative to each other, the extrinsic parameters. The pattern used in the testbed is shown in Figure 2.10b and was printed on A3 pa-

³<https://www.opencv.org>

per. Images of the pattern are captured at various orientations (see Figure 2.11) to increase information entropy, improving the overall accuracy of the calibration. Since the quadrotors fly at distances of 1.5–2.4 m to the cameras most of the time, the calibration is also done in this range.

After the intrinsic calibration process, all cameras used in the infrastructure are stereo-calibrated in pairs, again using the according OpenCV routine, in order to determine their extrinsic parameters. The concrete transformations between individual components are shown in Figure 3.20. The extrinsic transformation matrices that result from the camera calibration step described in Section 3.1.1 are $T_{c_0c_1}$ and $T_{c_1c_2}$ for the vision cameras.

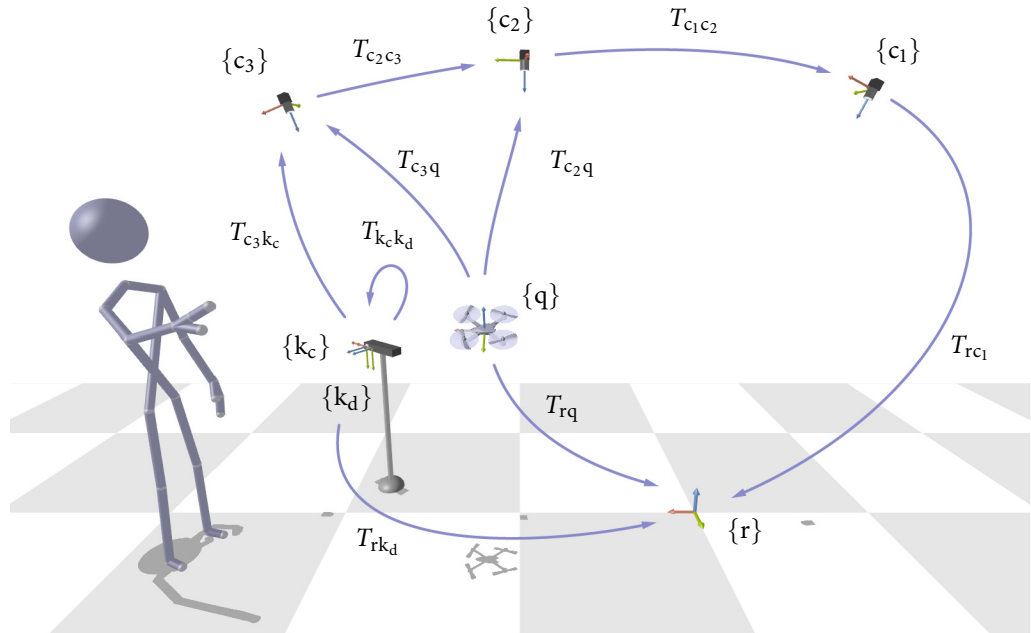


Figure 3.20: Concrete coordinate system transformations T between individual components of the testbed setup and the common reference frame $\{r\}$. The individual coordinate frames correspond to the industrial vision cameras $\{c_1\}$, $\{c_2\}$, $\{c_3\}$, a quadrotor $\{q\}$, and Kinect color and depth sensors $\{k_c\}$, $\{k_d\}$, respectively.

In order to achieve perfectly time-synchronized images for pose estimation, the cameras are triggered by an electric circuit. Such a circuit can be realized easily with the specific camera hardware used. The current setup of *ICARUS* uses three Ximea MQ013MG-ON USB3 high-speed vision cameras [XIM19] with a resolution of

1280 × 1024 pixels and Fujinon DF6HA-1B 1:1.2/6 mm lenses. The approach implemented in the testbed involves the main camera being software-triggered through the Ximea xiAPI, operating in fixed *FPS Mode* set to 100 Hz. In comparison to the *Free Run Mode*, where cameras acquire images as fast as possible, the *FPS Mode* fixes the image acquisition at the defined frame rate. The software-triggered camera then triggers the passive cameras through the electric circuit shown in Figure 3.21.

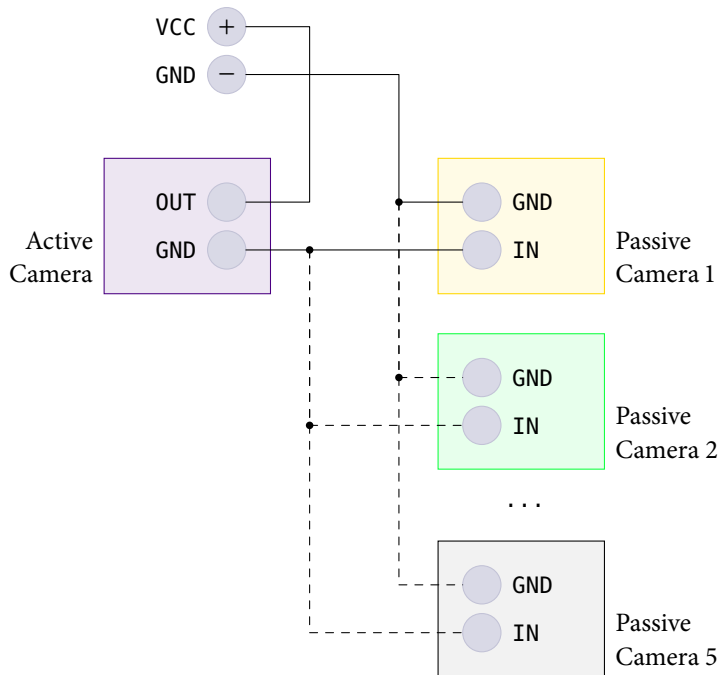


Figure 3.21: Schematic of the electrical synchronization of the cameras. The active camera is limited to a current output of 25 mA, while each passive camera consumes 5 mA, thereby limiting the maximum number of passive cameras to five [XIM19].

The pose estimation system employed [Tja19] relies on the OpenCV library and can run on a dedicated machine, if required. It captures images from hardware-synchronized vision cameras, which are then processed asynchronously using the single-core library implementation. Poses are averaged when multiple cameras detect the same marker, as described in Section 3.1.4. Then time-based pose smoothing is applied as described in Section 3.1.5. In the testbed implementation, the smoothing time interval is set to $\Delta t = 0.01$ s, and the time constant, determined through experimentation, is set to $\tau = 0.05$ s, resulting in $\alpha \approx 0.18$. This value

has been chosen as it effectively smoothens poses without introducing noticeable lag. The averaged and smoothed six DOFs poses and their corresponding time points are then multicasted as UDP messages. To reduce load, pose representation is changed from a 4×4 matrix to a vector representing the quadrotor's position and a Rodrigues vector representing its attitude. The drone's identifier, the pose's time point, and the pose itself are then serialized using the Protobuf library and broadcast as UDP messages using the Boost network libraries.

The tracking markers are aligned with the drone's frame and, thus, in \times configuration, do not point in the direction of flight. Furthermore, the testbed follows the general convention of having the z axis point upward instead of downward, as predefined by the pose estimation system. To adjust the coordinate frame, a local marker offset transformation T_{mq} is applied. This transformation also incorporates a translation that results in the pose estimation reporting the quadrotor's altitude as 0 m when it is lying on the ground. In total, the local marker offset transformation is composed of two consecutive, homogeneous rotation matrices R_x , R_z and a translation matrix T_z :

$$T_{mq} = R_x(\pi) \cdot R_z(\psi) \cdot T_z(-z).$$

$R_x(\pi)$ rotates the marker's z axis upward, R_z rotates the x axis into the desired direction of flight, and T_z moves the origin to the lowest point of the quadrotor. Using a quadrotor in $+$ configuration, the marker can be aligned with its x axis pointing in the direction of flight, and R_z can thus be omitted. For the Crazyflie in Figure 3.6, the specific parameters are $\psi = \frac{\pi}{4}$, and $z = 30$ mm.

The Kinect 2 RGBD camera, that is used for human pose estimation or "skeleton tracking" is designed with two sensors: an RGB color sensor with 1920×1080 pixels resolution and a depth sensor that has 512×424 pixels resolution. The depth sensor employs the ToF approach in order to estimate distances to objects for each pixel. This is done by measuring the RTT of an infrared light that is emitted from the device and captured by its IR sensor. The skeleton joints determined by the Kinect 2 SDK are tracked with respect to the Kinect 2's depth camera frame. In order to transform the skeleton joints into the common reference coordinate frame, the depth sensor has to be registered into the frame of one of the vision cameras that track the quadrotors and are used to define the common frame of reference.

The depth sensor of the Kinect 2 is only able to capture the pattern of a printed calibration rig at very close distances. However, the Kinect 2 must be placed at a relatively large distance from the flight area in order to capture people in their

entirety while they interact with the quadrotors within the flight area. At that distance, where a printed calibration pattern would overlap with one of the vision cameras, it is not visible anymore. This is why the Kinect's color sensor is used as an intermediary transformation, registering the Kinect 2's depth sensor into the global frame of reference through the calibration procedure described below.

Utilizing the same calibration pattern as for the extrinsic calibration of the vision cameras, at close distance, the depth sensor $\{k_d\}$ is stereo-calibrated into the frame of the color sensor $\{k_c\}$. The color sensor is then stereo-calibrated into the coordinate frame of one of the vision cameras $\{c_i\}$ whose field of view overlaps. The results of the Kinect 2 extrinsic calibration steps are the two transformations $T_{c_i k_c}$ and $T_{k_c k_d}$. The coordinate system transformation $T_{k_c k_d}$ connects the skeleton coordinate frame and the color sensor frame, while the transformation $T_{c_i k_c}$ connects the color sensor with the overlapping vision camera frame $\{c_i\}$. The transformations are shown in Figure 3.22.

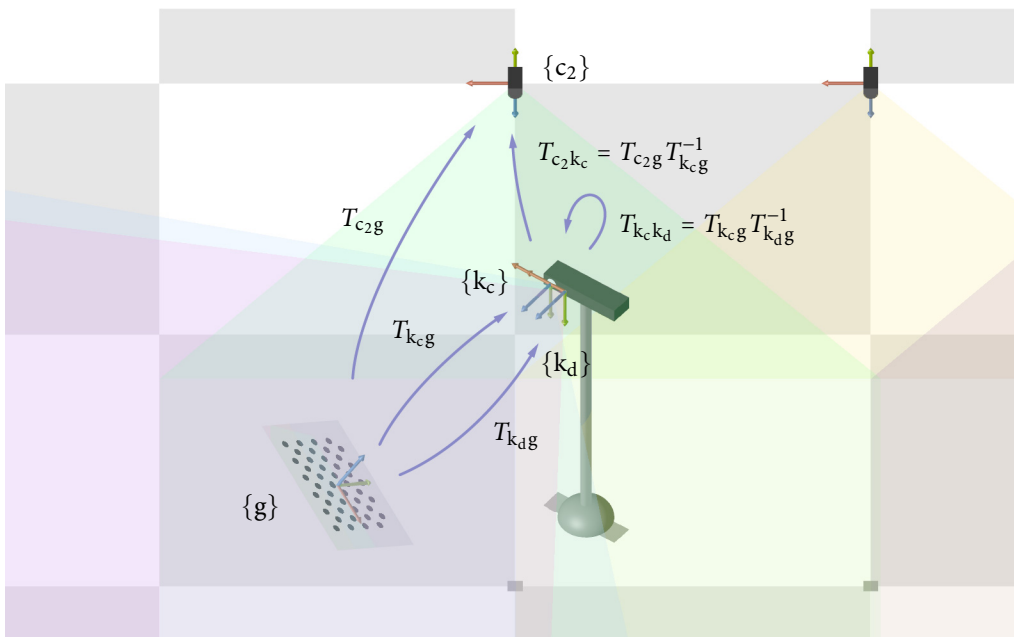


Figure 3.22: Utilizing a printed calibration rig, which spans a coordinate frame $\{g\}$, the extrinsic transformation of the depth sensor frame $\{k_d\}$ of the Kinect 2 relative to one of the computer vision cameras, e. g., with coordinate frame $\{c_2\}$, can be determined. The calibration is carried out through the color sensor $\{k_c\}$ of the Kinect, which serves as intermediate frame. The calibration from the color sensor to the depth sensor and the calibration from the color sensor to the vision camera are performed in two independent steps.

Chaining these transformations, skeleton positions can be referenced with respect to the common reference frame $\{r\}$ by the change of basis

$$T_{rk_d} = T_{rc_k} T_{c_k k_c} T_{k_c k_d}, \quad (3.6)$$

where $c_k, k \in \{1, \dots, n\}$ is one of n vision cameras whose field of view overlaps with the Kinect's color camera k_c .

3.5 USER STUDIES

For HDI user studies, which often rely on hardware and human presence to generate statistically relevant findings, a substantial number of subjects across a diverse range of demographics is necessary. Acquiring a large pool of volunteers can be challenging, particularly at smaller institutions. Often participants are limited to students and faculty members, resulting in constraints on age range, gender, and background diversity.

To reach a wider demographic despite the need for physical presence and real hardware, technologies like virtual reality (VR) can be utilized. Being able to conduct VR-based user studies saves the need to set up an outside-in tracking system, implement security mechanisms to protect study participants and hardware, and can provide all participants with identical trajectories for evaluation.

Especially since the testbed already includes simulation and an OpenGL visualization, which can be ported to VR and augmented reality (AR) devices, virtual user studies could be based on it. This presents a valuable opportunity to assess the participants' understanding of how a quadrotor should move or their ability to interpret the intentions of the robot. While both in-person and virtually conducted user studies involving drones reveal similar trends [Kun+23], certain parameters influencing the inhibition threshold are challenging to simulate in a virtual environment. Especially when evaluating proximity with aerial vehicles and how people generally react to the presence of drones, the physical size, rotor noise, and downwash generated by the rotors have an immediate effect on the participants' emotional states. Visuals and sound can be easily generated with VR, but not the downwash caused by the rotors, which plays an important role in people's acceptance of drones in their vicinity. Not being able to transport the feeling of downwash into virtual environments can negatively affect the meaningfulness of user studies, especially when evaluating psychological factors.

To improve the missing immersion of VR drone applications, a simple device that can generate rotor airflow representative of the drone may suffice. A prototype of such a device is shown in Figure 3.23. Connected to the infrastructure, it can simulate the collective thrust of a trajectory recorded for evaluation by a real drone. With this physical extension available, user studies could become statistically comparable to those conducted in reality. The main advantage, however, is that a much larger test group can be addressed, as there is no need to setup an elaborate testbed—only a VR headset and the downwash generator are required.



Figure 3.23: Prototype of a downwash generator to improve immersion in virtually conducted HDI user studies.

3.6 USER CODE EXTENSIONS AND APPLICATIONS

The *ICARUS* infrastructure has been utilized for many student theses and semester projects. The following of this section describes a selection containing an FPV simulator, light paintings created by the quadrotors' trajectories, a quadrotor collaborating with a ground robot, a SLAM porting, the inspection of unknown environments, and gesture-based interaction.

3.6.1 FPV Simulator

During an FPV drone flight, the pilot wears special goggles, that display a live video feed from a camera mounted on the drone, while using a remote control to perform *acro* (from acrobatic) maneuvers such as flips, rolls, dives, and power loops or smooth, *cinematic* flights. While more familiar consumer drones are stabilized, FPV drones are typically manually controlled, allowing for more natural, dynamic movements. A maximum total take-off weight of 0.249 kg allows flights close to

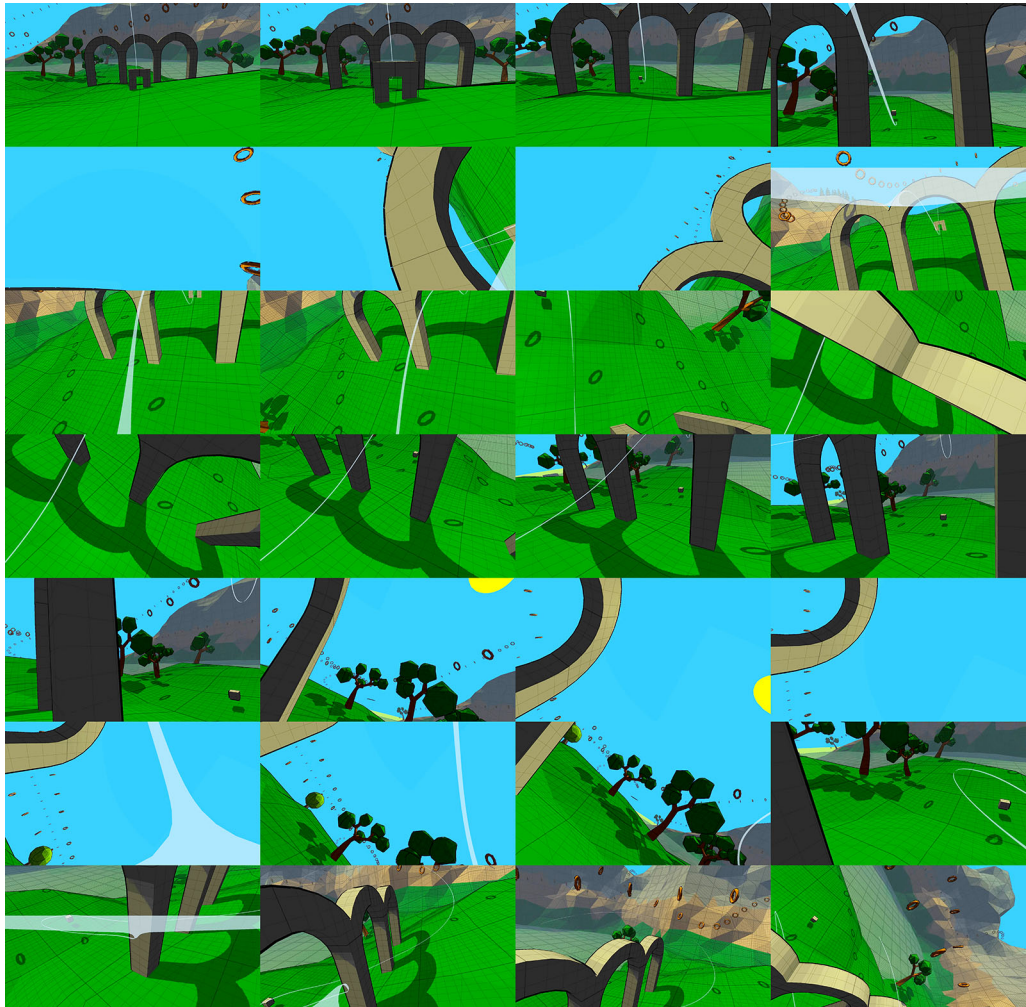


Figure 3.24: A sequence of stills from an FPV flight based on the simulation and visualization of *ICARUS*. The full video can be viewed online at <https://youtu.be/cnkKfDZKevo>.

people and within buildings. The compact design of FPV drones and manual control make it possible to capture unprecedented perspectives. Practicing in a physically correct simulation is important to avoid breaking expensive hardware and *ICARUS* already offers the necessary components for this. Figure 3.24 shows still images from an FPV flight simulation through virtual environments.

3.6.2 *Light Painting*

Light painting is a photographic technique that usually involves using a handheld light source to create light trails or patterns in a dark environment while taking a long exposure photograph. Experiments in which the handheld light source is replaced by a drone following trajectories can be seen in Figure 3.25. The resulting photograph captures the drone's movement and—in addition to the artistic appeal—provides a good feeling for the accuracy of the control system.

3.6.3 *Ground Robot Collaboration*

Especially in the field of search and rescue (SAR) operations the limited FOV of onboard sensors and cameras mounted on an uncrewed ground vehicle (UGV) can easily be expanded by collaborating with UAVs. This collaboration enables the investigation of the surrounding area from a higher point of view than the ground robot could achieve. Furthermore, a UAV can be exploited to extend the mobility of a ground vehicle, that may not be able to reach areas of interest through cluttered environments. In turn, the UGV can serve as a mobile power supply platform for the UAV, such as by charging its batteries through wireless power transfer.

With this in mind, *ICARUS* was extended by an experimental SAR application, which implements a flying periscope [And15], inspired by a system previously proposed [Fae+14]. For this project, the whole infrastructure including the pose estimation system was employed on a Raspberry Pi 3 using the Raspberry Pi camera module, operating at 50 Hz with 1280×720 pixels resolution. A prototype of the scenario is shown in Figure 3.26, where a quadrotor is controlled from an iRobot Create 2 (Roomba) development platform.

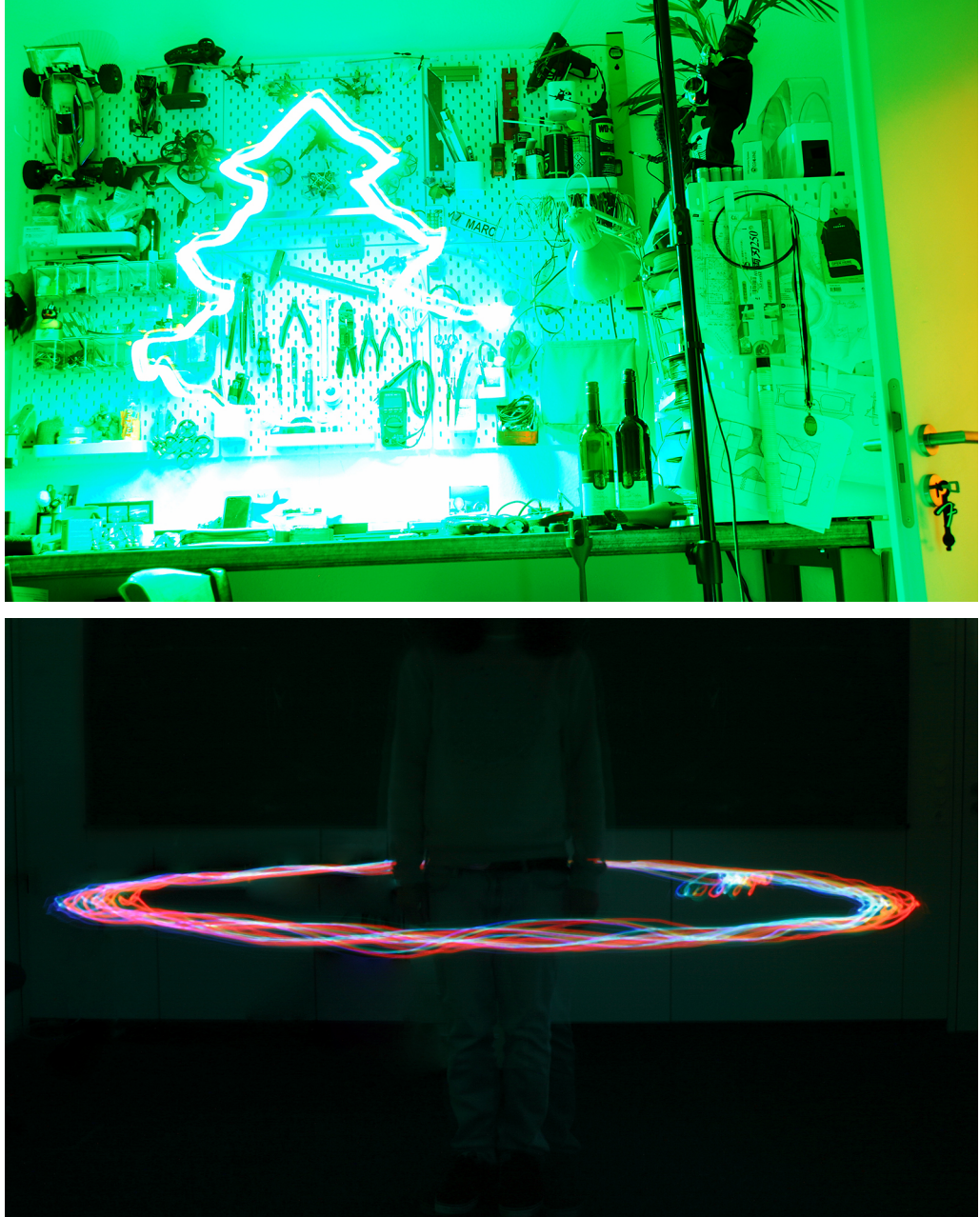


Figure 3.25: Exploring the possibilities of indoor light painting with drones.



Figure 3.26: Collaboration between a mobile ground robot and a UAV. The left image shows the setup before take-off. A Raspberry Pi equipped with the Raspberry Pi camera module (indicated by an orange circle), is employed for optical pose estimation. It tracks the marker mounted to the bottom of the drone through an acrylic pane, which serves as a landing platform. In the image on the right, the hovering drone (circled) transmits images from an onboard camera to a computer for further processing.

3.6.4 Inside-out Tracking Extension

This project [And16; Wei16] ported a visual SLAM implementation, i. e., ORB-SLAM [MMT15] to use with the infrastructure. Expanding upon the outside-in tracking approach with an inside-out tracking method enables the drone—in this case a Parrot Bebop 2 shown in Figure 3.27—to extend beyond the typical operational radius of *ICARUS*. The drone was extended by a tracking marker and a monocular camera with an 5.8 GHz video transmitter.

The problem of estimating absolute scale in monocular visual SLAM approaches was solved by synchronizing poses of both tracking methods and utilizing the

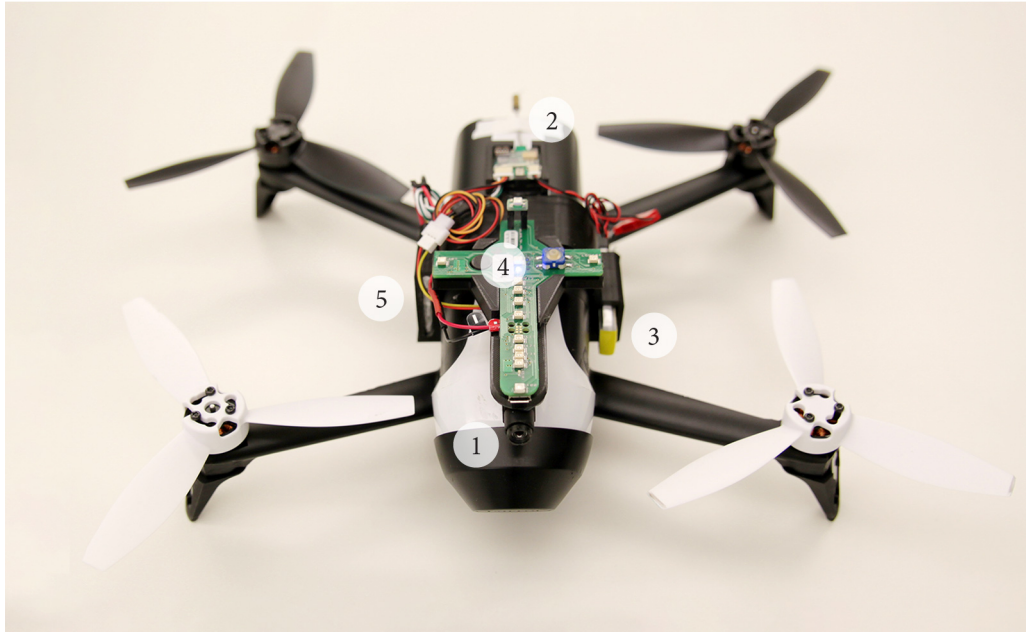


Figure 3.27: A Parrot Bebop 2 was employed to “escape” the testbed’s outside-in tracking by utilizing SLAM algorithms. An FPV hobby camera (1) sends images through a video transmitter (2), both powered by a LiPo battery (3), to the computer that runs SLAM. The tracking marker (4) design predates the circular-shaped design used throughout this thesis and is powered by a dedicated LiPo battery (5). Utilizing this pose estimation approach, which provides metrical poses, alongside SLAM, addresses the challenge of absolute scale estimation in SLAM methods.

already metrical poses delivered by the pose estimation system employed in the testbed. The calibration of the camera relative to the marker was performed by pointing the drone’s camera at a calibration pattern. During the process, both the drone’s marker and the calibration pattern had to be simultaneously visible to a testbed vision camera.

3.6.5 *Inspection of Unknown Environments*

A similar extension to ICARUS, but initially equipped with more primitive sensor technology, was the autonomous navigation of a drone through unknown environments [Rig19]. This was enabled by using expansion boards that provide

the Crazyflie with additional sensors. The flow deck⁴ is equipped with an optical flow sensor, which detects relative motion between the quadrotor and the ground and a ToF sensor, that measures the distance of the quadrotor to the ground. The multi-ranger deck⁵ adds four ToF sensors in the four horizontal directions and one pointing upward. These sensors allow for a rough reconstruction of the environment and the localization of the quadrotor in that map, as well as autonomous navigation to a target point specified by the user. An additional camera attached to the drone was used to detect objects of interest for closer examination. This was achieved by having the drone approach the object and film from various perspectives, gathering sufficient information to create a 3D reconstruction of the object.

3.6.6 Reinforcement Learning Control

Using reinforcement learning strategies, a neural network was trained to control a quadrotor in stabilization and waypoint tracking [Lar18] based on the simulation implemented in the infrastructure. The policy network is a function that directly maps the quadrotor state to rotor thrusts and is based on [Hwa+17]. This experiment showed that it is possible to unify more complex control structures by learning strategies.

3.6.7 Gesture-based Interaction

Since operating control levers is not very intuitive, *ICARUS* was extended by a gesture-based interaction approach [Gut15] to allow multiple participants to control multiple quadrotors by the motion of their hands using a Kinect 2. In the proposed interaction scenario, the user starts with both arms resting at their sides. They then point the right arm to a drone to select it. Keeping the right hand pointed at the drone, raising the left arm causes the drone to lift off. As long as the left arm is up, the user is controlling the quadrotors position with his pointing hand. The quadrotor then moves in relation to the hand. Lowering the left arm keeps the quadrotor hovering at its last position while allowing it to be selected

⁴<https://www.bitcraze.io/products/flow-deck-v2>

⁵<https://www.bitcraze.io/products/multi-ranger-deck>

by other users. If both hands are brought together at hip height, the drone lands. This scenario is illustrated with stills from the main application in Figure 3.28. A multi-user interaction video can be found on YouTube⁶.

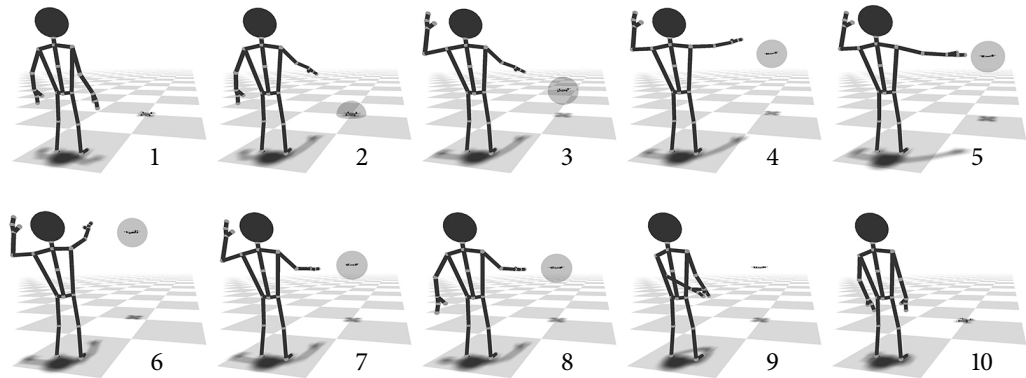


Figure 3.28: Gesture-based interaction: A user selects a quadrotor by pointing at it with the right arm (2). Lifting the left arm instructs the quadrotor to take off (3). While the left arm is up, the quadrotor imitates the motion of the user's right hand (4–7). When lowering the left arm, the quadrotor stays in hover mode and can be selected by another user (8). Bringing both hands together at hip height (9) lets the quadrotor perform a landing (10).

⁶<https://youtu.be/l0bxZJoUZrw>

4

NONVERBAL COMMUNICATION IN HUMAN-DRONE INTERACTION

The fundamental questions addressed by HRI are as old as the emergence of robots. Some of them were described in Section 1.2 and, in essence, they specifically explore how to make the interaction satisfactory for both sides, human and robot. Over time, robots change as technology evolves. Increased confrontation also shifts people's attitudes toward robots. While many questions, goals, and solutions already explored for the main part remain consistent, they may not always be transferrable to new types of robots. This can result from technical constraints or limitations that are essential to ensure the robot's functionality. In the past, most of the research was done with ground-based robots, as drones are still very young and modern drones only appeared in the consumer sector half a decade ago. Even with drones, introducing nonverbal communication channels, for example, has the potential to enhance both the interaction experience and humans' overall perception of the robot. Especially smaller drones have to make sacrifices in terms of equipment in order not to be particularly limited in their flying time. For this reason, this chapter explores interaction opportunities that can be implemented without the need for additional hardware or hardware beyond what is already required for the control and flight of the drones within the testbed.

Initially, the multifaceted research in the field of nonverbal HDI, which is the subject of this chapter, will be described using a demonstrator based on the previously developed drone infrastructure, *ICARUS*. A photograph of a live demonstration is shown in Figure 4.1. The setup comprises a single camera on a tripod, enabling the camera to be positioned at a height of approximately 3 m. A flight height of 1.1 m, which is comfortable for interacting with the drone by hand, results in an interaction area of approximately $2 \times 2 \text{ m}^2$. The interaction scenario is as follows: Due to technical limitations and safety measures, the quadrotor is only able to fly in the volume determined by the cameras. The human double-taps the quadrotor, e. g., to instruct it to perform an action. In this case, the action is to move out of the way. This is done by the drone horizontally transitioning 0.3 m away from the direction it was tapped on the frame. If the target position remains within the



Figure 4.1: Transportable exhibit of the drone infrastructure, *ICARUS*, consisting of a laptop, camera, and drone, during a live demonstration at Darmstadt University of Applied Sciences.

feasible flight volume, the drone anticipates its movement by first flying slightly in the opposite direction before proceeding to the intended position. If the target position is beyond reach because it is located outside the visible area of the camera, the drone remains stationary at its current location. In either case, the drone provides feedback to the human interaction partner. This is done through replicating human *nodding* or *shaking* of the head. Both gestures are commonly used in Western cultures to nonverbally convey agreement or disagreement¹. The trajectories of both scenarios are shown in Figure 4.2. This exhibit was presented, among others, at the open day of the Doctoral Center Applied Informatics (PZAI) and a video of the demo was published online². In addition to the visible cues, the negation gesture was augmented with acoustically perceivable information to enable an audible distinction between the two aerial feedback gestures of the quadrotor.

This chapter covers and goes beyond the individual aspects of the HDI demonstration described. It discusses available nonverbal communication channels between humans and drones and goes into detail with the selected channels, which are

¹Nodding and shaking head gestures are not innate, i. e., some cultures have different or even opposite meaning.

²<https://youtu.be/RUAcDskj8nI>

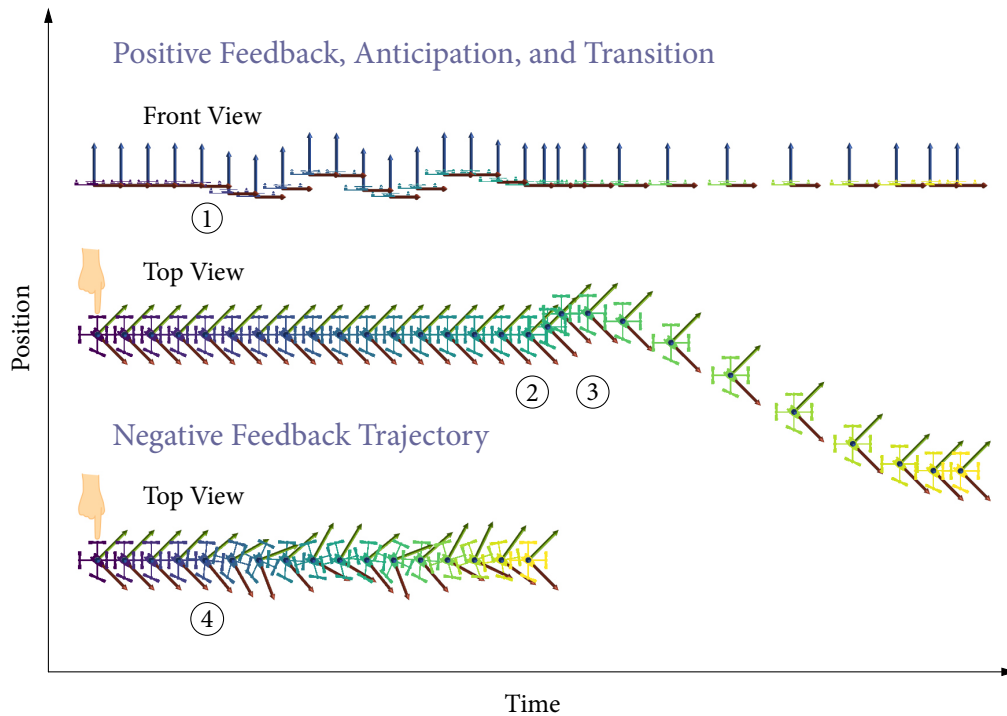


Figure 4.2: Trajectories used in the HDI demonstration: Shortly after being tapped, the drone either affirms the action (top) with a brief “nod” (1) before anticipating its movement (2) and flying toward the targeted location (3) in the direction indicated by the tapping hand, or rejects the action (bottom) by “shaking its head” (4), accompanied by acoustic feedback.

kinesics, proxemics, haptics, and vocalics, all of which are represented in the demonstration described above. In Section 4.1, the animation of trajectories is discussed and evaluated as a method to lower the inhibition threshold in humans when drones approach them. Section 4.2 examines the tolerated distances of miniature drones from four approach directions. A novel method for direct touch interaction is introduced in Section 4.3, with its implementation in a proposed interaction scenario serving as a proof of concept for virtually defined buttons. Finally, Section 4.4 explores the manipulation of a quadrotor’s rotor sound, evaluating it as a novel approach for acoustic feedback.

4.1 KINESICS

The research field of *kinesics* studies body language, gestures, and nonverbal behavior related to the movement of body parts or the body as a whole. It aims to comprehend the impact of physical movements on communication, especially in the context of human interactions. In robotics, motion is often designed to execute actions as efficiently as possible. For example, trajectories frequently describe the shortest paths between two points in the configuration space of the robot. As a consequence, the robots' movements follow calculated, linear, "mechanical" patterns, which can make it difficult for humans to interpret the robots' intents or emotions.

This issue has led to various strategies for enhancing interaction, e. g., by reducing mental stress and lowering the inhibition threshold experienced by humans. One such strategy involves introducing social features [Yeh+17] by adding a tablet displaying a face and a "happy" greeting voice to a quadrotor. However, adding physical, anthropomorphic features to miniature flying robots is often impractical due to their functional design and critical payload constraints. In such cases, robots can express their inner state, emotions, or intent through their movements.

Hoffman et al. [HJ14] distinguish between *pragmatic* and *expressive* movements. Pragmatic movements are designed by the engineers who developed the robot to fulfill specific requirements, typically achieving physical goals with maximum efficiency, reducing energy consumption, and minimizing costs. In contrast, expressive movements are designed to convey traits, states, or intentions to human interaction partners. For drones, which typically lack moving parts other than rotors, this expressiveness can be achieved through their trajectories. Studies have examined the potential to convey affect through a quadrotor's flight path [Sha+13] or to acknowledge a person, e. g., by wiggling [JHK18]. However, much of this research either lacks detailed implementation and parametrization [Cau+16] or remains conceptual [Den+18].

When the external appearance of robots remains mechanical for functional reasons, kinesic approaches can still effectively communicate intentions. For anthropomorphic robots, this is often achieved through gestures such as nodding or head-shaking, which are easily incorporated due to their human-like physical features. Designing robots to resemble humans helps them fit into environments tailored for human use and fosters familiarity, increasing comfort during interactions.

However, transferring human gestures to robots that lack human-like shapes, such as quadrotors, presents additional challenges. Unlike anthropomorphic robots, quadrotors cannot directly mimic human gestures due to their distinct appearance and operational mode. Nevertheless, the introductory demonstration (Figure 4.2) illustrates that the transfer of basic gestures is possible. To effectively convey messages through movement, established principles from kinesics and character animation can be adapted for quadrotor trajectories.

4.1.1 *Animation Principles*

Disney’s so-called “Nine Old Men” established a well-known set of animation principles [TJ81]. These principles are often referenced in the fields of character animation and robotics design. Some principles have already been adapted for robots, such as designing a music-listening companion that moves to the beat [HJ14]. The application of animation principles to a robot’s facial expressions enhanced human comprehension of the robot’s emotions [RP12], while applying these principles to a virtual ground-based robot improved its readability [TDJ11]. The concept of anticipatory movement, i. e., movement that prepares humans for an upcoming action, was transferred from Disney’s principles of character animation to both ground-based robots [GT11] and drones [SMF14]. Compared to more conventional approaches of indicating motion intent, such as blinking LEDs [SMF15; GC23] or using AR [Wal+18], anticipatory motion is more expressive, thus adding liveliness.

Some principles may require adaptation or modification to align with the specific design, mode of operation, and capabilities of quadrotors. Due to the drone’s adherence to the laws of physics and a desired pragmatic configuration that limits moving parts to rotors, certain principles of character animation are not directly applicable to quadrotors, or at least not in their entirety.

The *straight ahead action and pose to pose* principle involves two different approaches of animation techniques. Straight ahead action involves animating a scene frame-by-frame in a linear fashion, while pose to pose involves planning out “keyframes” before filling in the in-between poses. As detailed in Section 2.3.2, the trajectories employed in this thesis are defined by quintic polynomials, with their boundary states serving as keyframes. This approach is similar to the pose to pose principle, which is already determined by the employed trajectory representation.

On the contrary, some principles can be easily applied to quadrotors by setting the appropriate boundary conditions of the linear system (2.24).

The *slow in and slow out* principle is used to create realistic acceleration of objects. Using quintic polynomial trajectories also simplifies this process, as quadrotor acceleration can be adjusted to physically meaningful values through the keyframe parameters.

The *arcs* principle can be implemented using velocity information and generate arched trajectories resulting in more natural motion. In contrast, a pragmatic robot motion would follow a straight line, i. e., the shortest path, between two points.

Timing is a powerful tool for evoking emotions and can be manipulated through the boundary conditions of the trajectories. Velocity and acceleration between keyframes can convey significant information about a character's personality and emotions. For example, slow movements can indicate a sense of relaxation or lethargy, while quick and jerky movements can convey nervousness or agitation. The idea that a character's personality is more defined by their movements rather than their appearance [TJ81] supports the notion of using natural motion to distract from the mechanical design of quadrotors.

A *secondary action* is often used in animation to add emphasis to the main action and give more life to a scene. If applied to quadrotor motion, this could involve yawing the drone's heading toward the next keyframe, creating a more natural and lively impression as it focuses on a fixed target as a human or animal does. A tornado-like, fast yaw-spinning drone as a secondary action throughout parts of a trajectory could convey playfulness, or play into the animation principle of *exaggeration*. This principle involves portraying character features in an exaggerated manner to overcome the limitations of static drawings that closely resemble reality.

The *staging* principle involves positioning of characters in a way that makes their action clear. This can be as simple as the drone moving out of the way or staying in place, when a user taps it like in the introductory example of this chapter.

As already mentioned, *anticipation* helps the audience preparing for actions. Furthermore it makes movements appear more natural. Anticipation is achieved through pre-emptive motion, similar to how a bowstring is drawn back in anticipation of releasing an arrow.

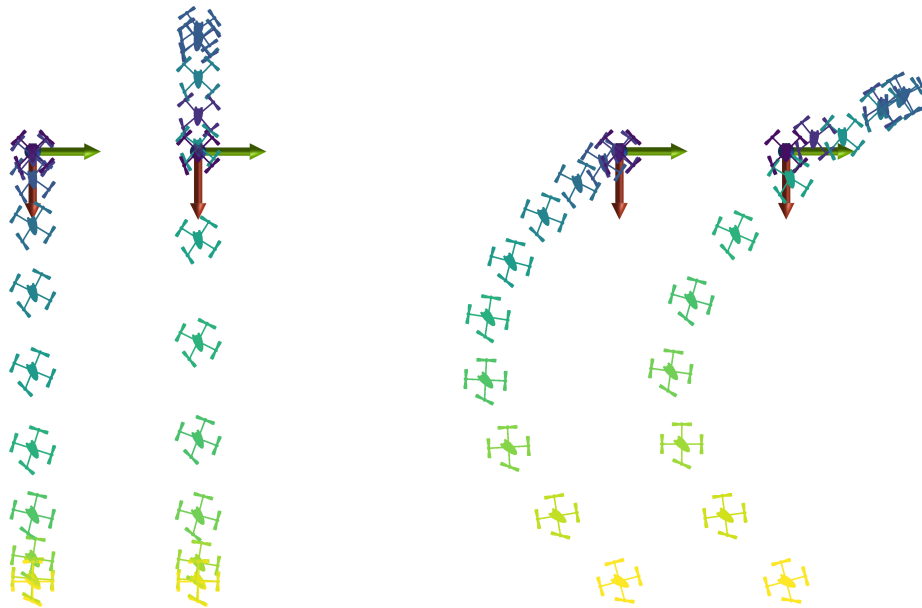


Figure 4.3: This figure illustrates two examples of adding anticipatory motion to trajectory segments. The keyframes for all four trajectories share the same positions. However, the end keyframes of the right two trajectories include velocity information. Each pair of trajectories, left and right, differs by incorporating anticipation of motion intent.

The principle of *anticipation* was applied in the design of quadrotor motion to avoid surprising or frightening participants with sudden movements, which could potentially intimidate them during interaction. The principles of *slow in and slow out* and *arcs* were also employed in this design. The resulting motion is depicted in Figure 4.3, where the quadrotor's motion is anticipated by extrapolating the first piecewise polynomial by a predefined coefficient, creating a counter-movement before the actual motion begins. Using polynomial extrapolation instead of linear extrapolation provides velocity and acceleration information in addition to information about the next keyframe's position. It is, however subject to greater uncertainty and poses a higher risk of yielding meaningless results. The extrapolation of the heading angle adds a counter rotation to the motion.

The principle *follow through and overlapping action*, where loose parts of a character still move after the body has stopped in order to follow the laws of physics and where parts of the body move at different rates, is not applicable to quadrotors. The well-known *squash and stretch* principle, which expresses physical parameters like weight, flexibility, or impact, can not be transferred to quadrotors as they are real-

world rigid objects and thus cannot be compressed. The principles *solid drawing*, where characters are created with a sense of weight, volume, and dimensionality in order for them to appear more realistic cannot be implemented due to its drawing technique-related nature. The same applies to the *appeal* principle, which involves creating characters visually appealing for the viewer to keep them in mind.

In a scenario where the quadrotors maintain a mechanical appearance, their trajectories are capable of diverting attention from their design and expressing intent or specific emotions, thereby enhancing the interaction experience for humans. The flight of a drone from a starting point toward a human can be designed in various ways, as illustrated in Figure 4.4. A mechanical or pragmatic flight (Figure 4.4a) would be the most energy-efficient approach. To prevent overtaking the human, the actual motion can be anticipated (Figure 4.4b) through a counter-motion. For a more expressive interaction, the quadrotor could approach in a rhythmic pattern (Figure 4.4c).

4.1.2 User Study

To evaluate preferred approach flight characteristics, a user study was conducted, that consisted of two parts, in which a pragmatic flight and an expressive flight were demonstrated to participants. After each flight, a questionnaire was presented to the participants containing 5-point Likert scales to evaluate the flight and to assess their emotional states through a self-assessment manikin (SAM) survey [BL94], a widely used rating scale to assess self-reported affective states. It consists of a set of pictorial representations of a human figure with different poses illustrated in Figure 4.5, indicating the dimensions of valence (negative–positive), arousal (calm–excited), and dominance (submissive–dominant). Participants are asked to select the figure, that best represents their current emotional state along each dimension. The questionnaire can be found in Appendix A.3.1 under Experiment 2 (Versuch 2) and Experiment 3 (Versuch 3), while the key research data is listed in Appendix A.3.2.

The user study involved a total of $N = 32$ participants, consisting of 11 females (34.4 %) and 21 males (65.6 %) with ages ranging from 19 to 42 years old ($\bar{x} = 25.90$, $\sigma = 5.44$). Most of the participants were students from RheinMain University of Applied Sciences and studying various fields of Computer Science (20), Media

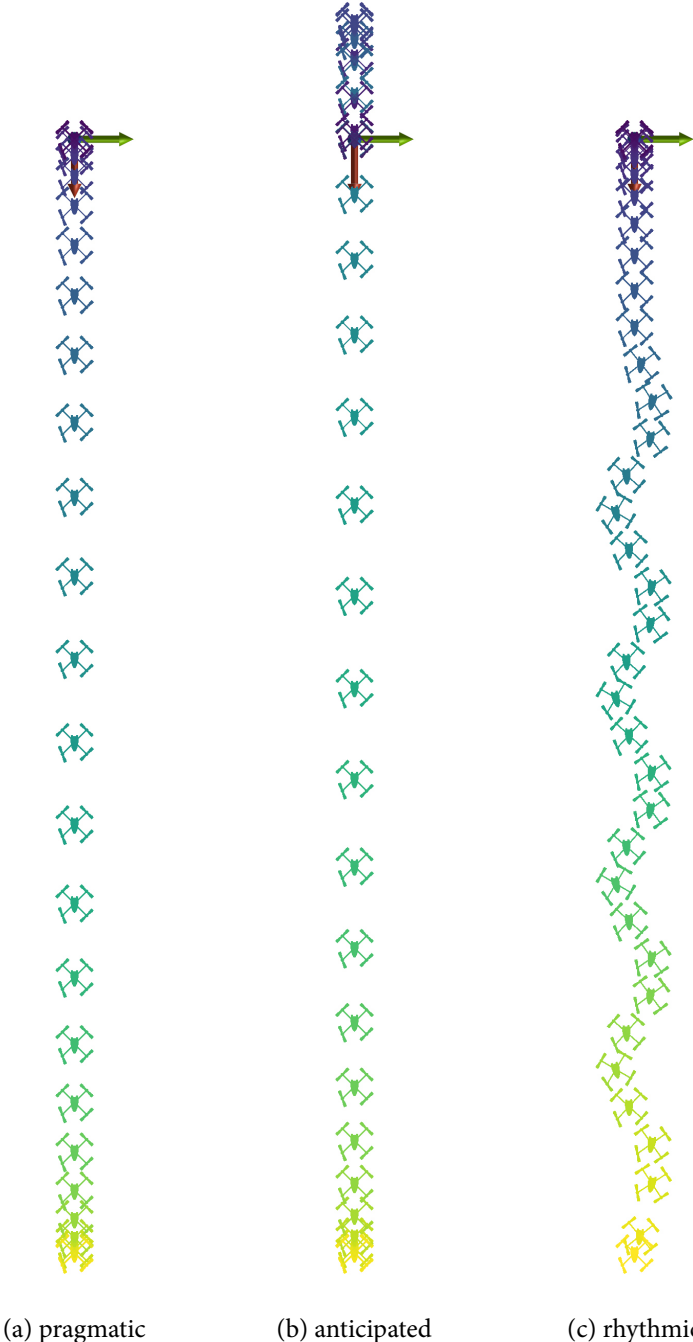


Figure 4.4: Different approach trajectory characteristics that were evaluated in the conducted user tests, where seated test subjects were approached by a miniature drone flying pragmatic (a), anticipated (b), and rhythmic (c) trajectories.

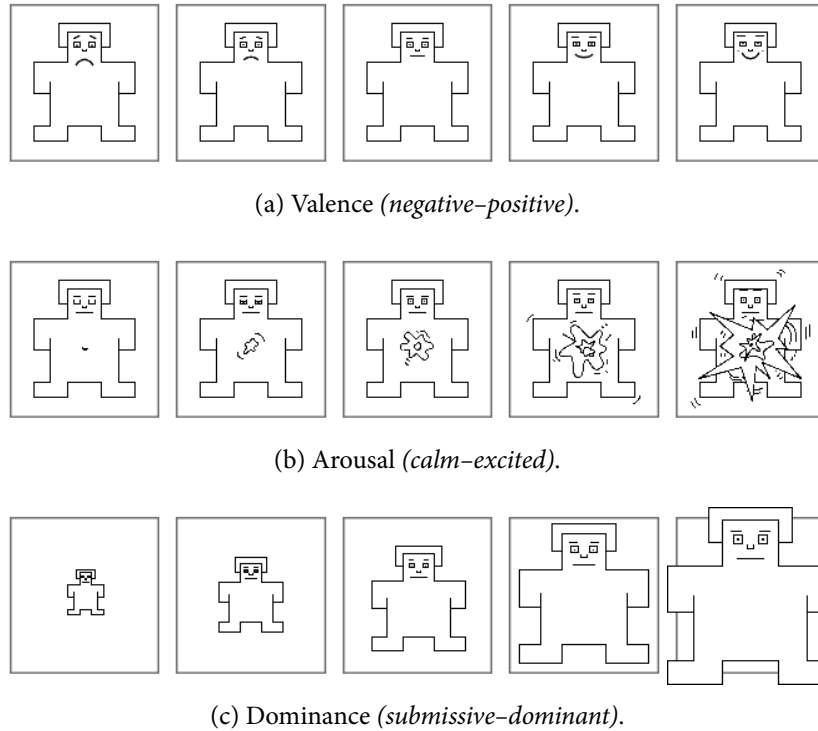


Figure 4.5: Pictorial representations of the human figure used in the self-assessment manikin rating scale [BL94].

Management or Media Conception & Production (7), and Mobility Management (1), or were research assistants (4).

Anticipated Trajectory

Since indicating motion intent is already subject of research in robotics, the anticipation principle of character animation was picked for this experiment. The participants were first shown a *pragmatic* flight (Figure 4.4a), and then a flight with *anticipation* (Figure 4.4b). The anticipation extrapolation coefficient for this flight was determined experimentally and set to $k_a = 0.3$, achieving the desired effects. A paired *t*-test resulted in the following, statistically significant ($p < 0.05$) findings: Participants felt more positive (valence) and calmer (arousal) when the quadrotor approached with anticipation. They also identified the anticipation as

motion intent ($p < 0.1$). The confirmation that anticipated movement is more positively received than pragmatic flight is supported by statistically significant responses, with participants reporting they felt “very comfortable” ($p < 0.05$) after the anticipated flight.

Rhythmic Trajectory

Participants were presented with two types of trajectories: a *pragmatic* trajectory (Figure 4.4a) and a *rhythmic* trajectory (Figure 4.4c). Out of the participants, 69 % preferred the rhythmic trajectory, while the remaining 31 % favored the more pragmatic trajectory. Statistically significant results ($p < 0.05$) from a paired *t*-test indicate that participants reported feeling calmer after experiencing the rhythmic trajectory. However, they perceived the flight as highly mechanistic and devoid of emotional or human characteristics.

4.1.3 Discussion

Kinesics are a powerful tool to convey intent, especially motion intent, as the study shows. Much like other aspects of nonverbal HRI, creators in visual storytelling, such as those in comics or animated films, have introduced ideas about robot behavior to their audiences. The ability of rhythmic movement to conceal the mechanical appearance of the robot could not be confirmed in this study. However, a dancing drone was preferred over the straight trajectory. Introducing animated trajectories could enhance the interaction experience, as participants reported feeling significantly calmer with anticipated movement. This improvement in emotional response is especially important in scenarios where discomfort might otherwise arise, as it helps lower the inhibition threshold and fosters more comfortable human-robot interactions.

4.2 PROXEMICS

The research field of proxemics focuses on how individuals perceive and use space during social interaction. It was first introduced by the anthropologist Edward T. Hall [Hal66] and has since evolved and found its way into a wide variety of disciplines. These include psychology, sociology, architecture, and HRI. Proxemics explore the way in which people use personal space and distance to communicate and establish social relationships, e. g., by claiming space to establish their identity, status, and ownership. For example, if a standing person talks to a seated person, it can give the impression of dominance. When a person is turned away from the communication partner, it can convey a feeling of disinterest. When someone (or something) unknown enters another person's personal or even intimate space, it can cause discomfort. Proxemics also investigate how arrangements of objects like furniture can affect interpersonal relationships and group dynamics in a given space. This helps designers create spaces that are conducive to effective communication, encourage positive social interactions, and respect cultural norms.

One of the key concepts of proxemics is the notion of interpersonal distances. These are defined as imaginary bubbles of space surrounding a person and describe interpersonal boundaries. Depending on the social status, role, or relationship, people use different distances to interact. While different cultures have different ideas of appropriate interpersonal distances, for Western cultures, these have been divided into four zones [Hal66]: intimate space, up to a distance of 0.45 m, personal space up to 1.2 m, social space up to 3.6 m, and public space up to 7.6 m or more.

Proxemics research also contributes to the understanding of human behavior, perception, and communication, and helps to better understand how individuals use and perceive space in their daily lives. These concepts, which were initially studied in an interpersonal context, have been adapted by HRI researchers for use in robotics, where the goal remains the same: understanding the effects of distances in communication to create positive interaction between humans and robots. Earlier user studies evaluated tolerated distances toward ground-based robots [Wal+05b; Wal+05a; Hue+06; Wal+07; TP09; MM11; FM12; Bha+19], while more recent ones assess drone proxemics [DM13; Yeh+17; ABD17], and compared real studies with virtually conducted user studies [Li+19; Kun+23].

Interactions predominantly occur within personal space (0.45–1.2 m). In personal space and especially in intimate space, a natural inhibition threshold exists that humans experience when it comes to interaction. Even more this is the case with robots, whose intents are hard for humans to understand. The inhibition threshold is higher in regards to flying and mechanical-looking robots compared to ground-based robots or robots with anthropomorphic features [ABD17; Yeh+17].

4.2.1 *Tolerated Distances*

To evaluate the tolerated and—depending on the type of interaction—preferred distance between a robot and a human subject, a commonly utilized process involves letting the robot approach the subject. If the subject starts to feel uncomfortable, they signal the experimenter, who then stops the robot and measures the remaining distance. This approach is common in proxemics studies with ground-based robots and has been similarly adopted for drones. Figure 4.6 illustrates tolerated distances identified in previous drone proxemics studies.

An early human-quadrotor proximity user study was conducted by Duncan et al. [DM13]. The largest quadrotor platform used in the discussed tests is the AirRobot AR100-B with a wheelbase of approximately 0.55 m and enclosed by a safety ring with a diameter of 1.0 m. The quadrotor was mounted to a moveable platform at the ceiling and the minimum distance allowed between the participants and the quadrotor was set to 0.6 m. This minimum distance was maintained by a mechanical stop in the rail to which the moving platform was attached.

In the user study conducted by Yeh et al. [Yeh+17], an off-the-shelf drone (DJI Phantom 3, wheelbase 0.35 m) approached participants as is (“non-social”), and provided with a face (“social”). The non-social drone was stopped at a distance of 1.35 m, while the social drone was able to approach at a distance of 1.06 m. To prevent the drone from drifting and moving unpredictably, it was suspended from a zip line instead of physically flying toward the participants.

The emotions of the participants are heavily influenced by moving rotor blades, the sound of the rotors, and the generated downwash [CCC17]. If studies on the psychological effects of robots on humans lack important factors, such as those resulting from the physical flight of drones, the research results may not be very meaningful.

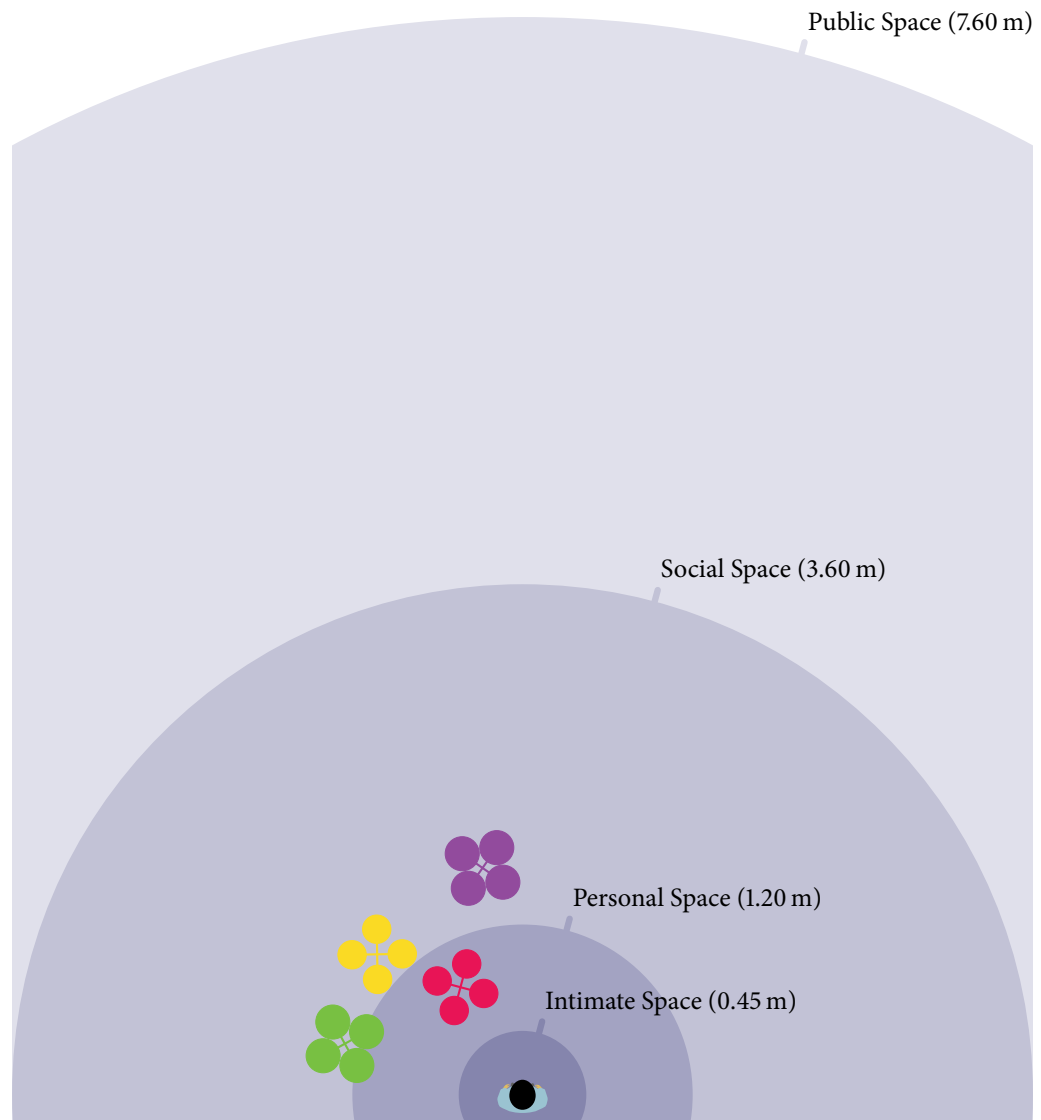


Figure 4.6: Comparison of results from related quadrotor proxemics studies with a person at the center. Bubbles represent interpersonal distances as defined by Hall [Hal66], their specified dimensions are radii. The quadrotors are presented in their actual sizes to enable a direct comparison with the other proxemics studies that involve quadrotors. The distances maintained between the humans and the quadrotors are referred to as “stop distances”, which were determined based on the participants’ instructions to halt the quadrotor from approaching further by saying “stop”: [AscTec Hummingbird](#) [ABD17], “social” [DJI Phantom 3](#) [Yeh+17], [Parrot AR.Drone 2.0](#) [Yeh+17], and [regular DJI Phantom 3](#) [Yeh+17].

In the user study conducted by Acharya et al. [ABD17], a professional motion capture system was used to ensure the reproducibility of the trajectories and thus the repeatability of the experiments. In their study, a mechanical-looking quadrotor (AscTec Hummingbird, wheelbase 0.34 m) was compared to a ground-based robot. The quadrotor was stopped at a distance of 0.65 m, which resulted in the participants experiencing a statistically significant increase in mental stress. In contrast to the previous studies, a motion capture system was used to fly the drone toward the subject in a precise and repeatable manner. However, the motion capture system used is costly and requires maintenance, posing challenges for smaller research institutions to conduct such studies.

The flight characteristics that an autonomous drone should employ when approaching a person were evaluated by Wojciechowska et al. [Woj+19]. In this experiment, four parameters were investigated: speed, direction, altitude and final proximity to the participants. The preferred conditions resulted in the quadrotor approaching at a constant speed of 0.5 m s^{-1} from the front, at a constant height of 1.75 m with the quadrotor stopping in the participant's personal space at 1.2 m distance. The study's authors noted a variation in the trajectories of the quadrotor.

Tolerated distances in proxemics user studies can be influenced by factors such as the size of the quadrotor and its rotors, the presence of rotor guards, or the inclusion of social features. As a result, comparing the previously described user studies is challenging. However, in all studies, the quadrotors were either too distant [Yeh+17; Woj+19] to enable close-proximity physical interaction scenarios, such as the one proposed later in Section 4.3, or an observed increase in mental stress [ABD17] may discourage participants from engaging in closer interaction when given the opportunity.

The lack of reproducible trajectories in the majority of proxemics studies conducted, missing implementation details, the increasing confrontation with flying robots, their further miniaturization, as well as possible delays caused by stopping the drone by proxy, necessitate a re-evaluation of the tolerated distances.

4.2.2 *User Study*

Since no studies have yet been conducted on the tolerated distance to miniature drones, this will be investigated through a user study. The preferred approach

direction of the drone will also be investigated. Considering the prospect of several service drones that could operate in the workplace of the future, the study will be conducted with participants in a seated position. Additionally, the tendency of users to interact with real, flying miniature drones using their hands will be examined.

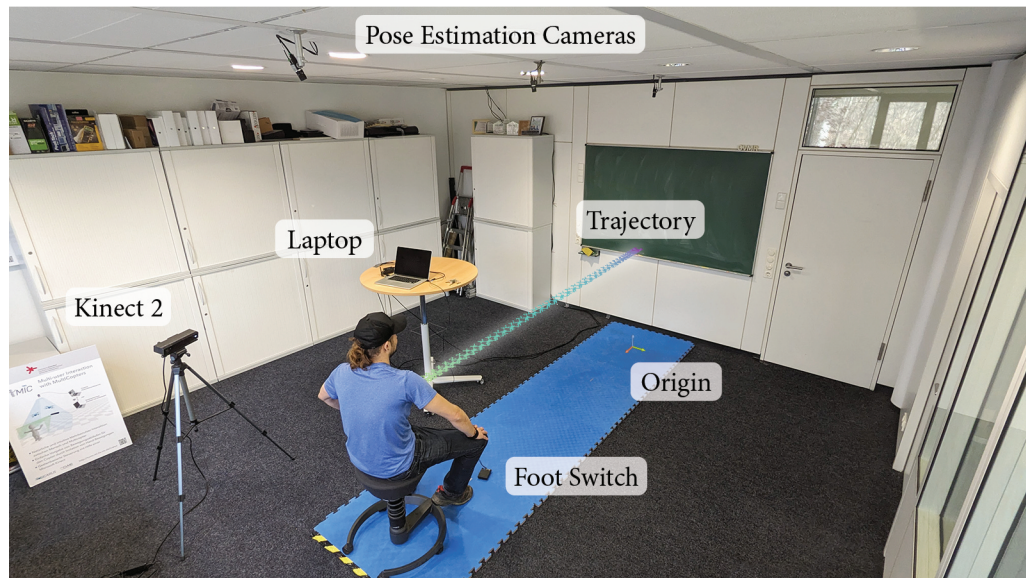


Figure 4.7: Proxemics user study overview: A participant sits at the end of the superimposed trajectory and can manually stop the quadrotor’s flight at their preferred distance using either a foot switch or their hand. The distance between the drone and the participant’s head is continuously monitored via skeleton tracking, running on a separate computer (not shown). This data is transmitted to a laptop running the *ICARUS* main application. The same laptop operates the pose estimation application, which processes images from three ceiling-mounted cameras. The calculated poses are sent to the main application, which controls the drone’s movement.

For this experiment, the *ICARUS* testbed described in Chapter 3 was set up in a room of size $6.8 \times 5.9 \times 2.7 \text{ m}^3$. Three cameras were adjusted to capture a flight area of approximately $4.5 \times 2.0 \times 1.4 \text{ m}^3$. A photograph of the experiment setting can be seen in Figure 4.7.

This user study was part of a larger study with $N = 32$ participants who were students and employees of RheinMain University of Applied Sciences, with their demographics described in Section 4.1.2. Relevant for this session, seven of the participants reported owning robots at home (six vacuum cleaners and one educa-

tional robot companion), while 16 had previous experience with drone, and two reported owning a quadrotor themselves.

Setting

At the beginning of each session, participants were given written instructions to ensure they all had the same basic knowledge before the experiment started. They were asked to carefully read the instructions, that included the photograph of the quadrotor shown in Figure 4.8, in which tactile interaction points are marked with arrows. A quadrotor identical to the one used in the experiment was lying

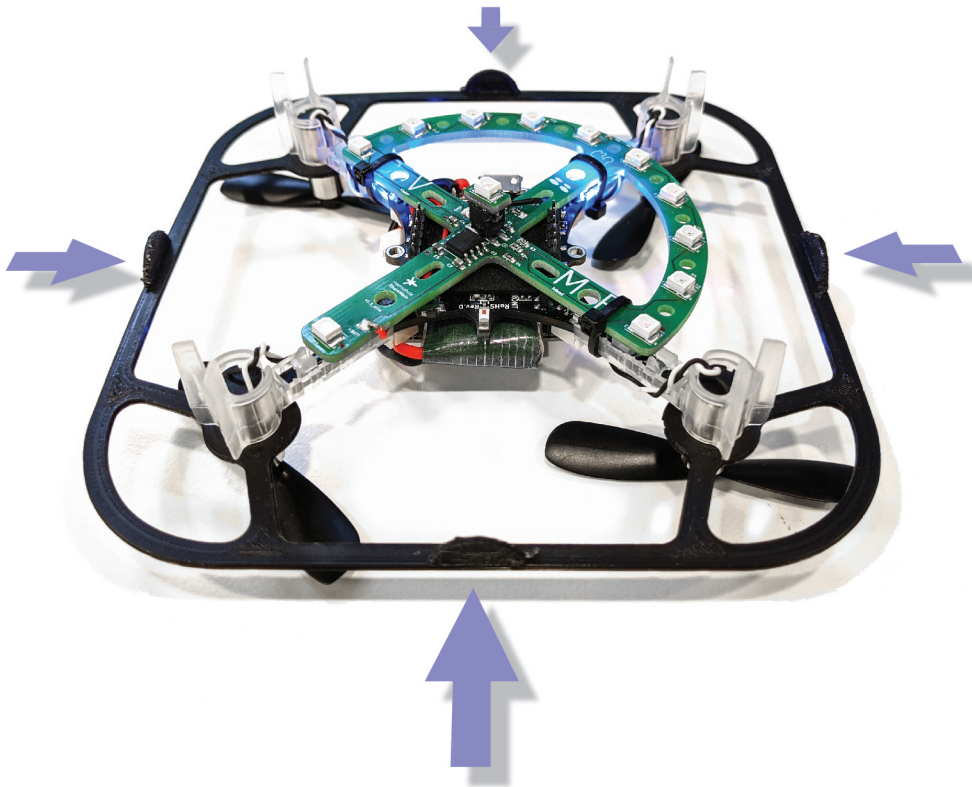


Figure 4.8: The Bitcraze Crazyflie 2.1 used in the presented proxemics study is equipped with a 3D-printed frame that offers four touch interaction points (marked with arrows) and provides protection to participants from touching the rotors.

on the table with the instructions. Participants were informed that a quadrotor

would approach them at moderate speed and constant height from four different directions (front, back, left, and right) in a randomized order. They were also informed that they could stop the approach of the quadrotor at any time if they felt uncomfortable, either by a foot switch or by tapping the quadrotor's frame directly at one of the interaction points. This way, participants were left to decide if they wanted to stop the quadrotor and, if so, how they wanted to do it. Additionally, the foot switch allowed participants to have both hands free to stop the quadrotor by hand if they chose to do so. Participants were not informed that the trajectory would end right in front of their chair and that the distance between their head and the quadrotor would be continuously monitored for safety reasons. The watchdog implementation would stop the trajectory and switch to the position controller if the center-to-center distance measured in the horizontal plane would fall below 0.3 m.

Before the test, the participants completed an initial form that included questions about their demographics, personality [SG08; Joh91], and previous experiences with robots, and drones in general. Additionally, to assess the participants' emotional states, they completed an initial SAM survey (Figure 4.5), provided as a 5-point Likert scale.

Finally, after completing the initial forms, the participants were accompanied to the flight laboratory to carry out the test. After each flight, the SAM survey was repeated along with further questions about their wellbeing (ranging from *not well at all* to *very well*) and perceived proximity of the quadrotor at its final position (ranging from *too close* to *too far*), both provided using a 5-point Likert scale. The complete questionnaire is included in Appendix A.3.1. A session took about 15 minutes and was closed with a short interview.

Hypotheses

The primary hypothesis of this proxemics study is that

people show different levels of affective responses and comfort with the quadrotor approaching from different directions.

This was assessed by the participants' affective responses and comfort levels toward different approach directions of the quadrotor by analyzing their responses to questionnaires and the distance at which they decided to stop the quadrotor from

advancing further. The chosen stopping method should also provide insight into the participants' willingness to physically interact with a flying quadrotor by touch.

Trajectory

The height of the trajectory was fixed at 1.1 m. When standing or sitting, this provides a comfortable height for hand interaction. In a seated setting, participants looked slightly downward at the quadrotor (head height $\bar{x} = 1.27$ m ($\sigma = 0.09$ m)) and only had to bend their elbow to raise their hand and stop the quadrotor through a tap to its frame (shoulder height $\bar{x} = 1.04$ m ($\sigma = 0.09$ m)).

Procedure

The quadrotor was positioned at the point of origin before each flight, where it lifted off and ascended to an altitude of $z = 1.1$ m, the starting point of the trajectory. The flights were initiated only after the participants had confirmed their readiness verbally. The quadrotor then flew to the endpoint at $x = 3$ m, positioned directly in front of the study participant. The duration of the flight was 6 s. The reference trajectory is shown in Figure 4.9. A three-dimensional visualization of the trajectory is included in the photograph of the study setting presented in Figure 4.7. The quadrotor would decrease its velocity toward the end of the trajectory and eventually stop right before the participant. The flight could only be intervened either by the participant manually stopping the drone or by the flight control software, in case its distance to the participant's head would fall below the safety threshold.

Results

The results presented in this section are based on the research data collected, consisting of the participants' responses on the questionnaire and parameters logged from the drone controller software. The full key research data is provided in Appendix A.3.2.

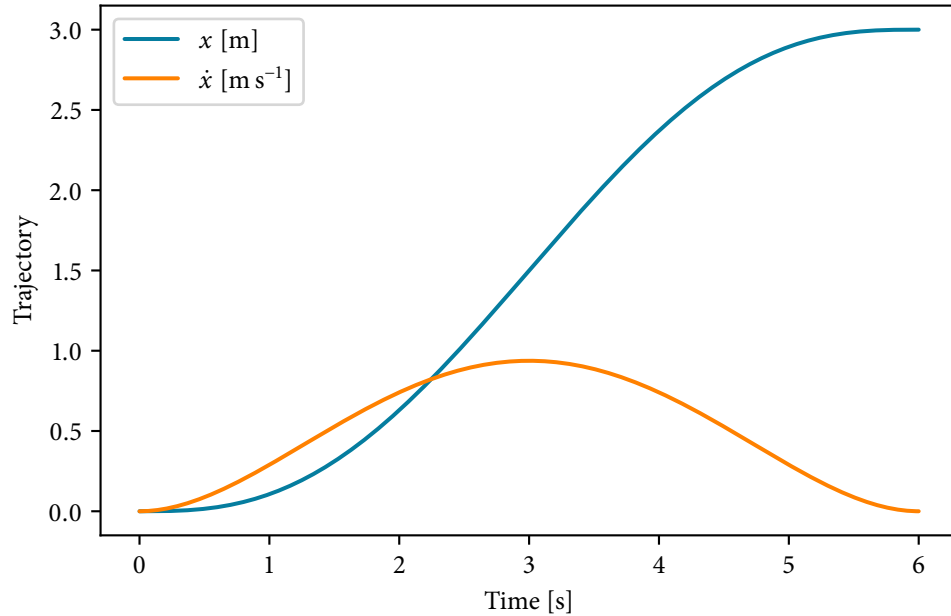


Figure 4.9: Position and velocity along the x coordinate of the reference trajectory.

Of the 128 flights conducted, safety measures intervened in 27 cases (21.0 %), preventing the quadrotor from further approaching. In 10 of these cases, involving three male participants, the quadrotor was not manually stopped, as the participants were curious to see what would happen if they did not intervene. In the remaining 17 cases, participants initiated a stop immediately following the safety stop intervention. As the quadrotor was already in the process of stopping in these cases, the exact intended stop distance could not be determined. Therefore, all 27 flights were excluded from the numerical distance evaluation.

Evaluation of the *approach stop proxemics* revealed that the quadrotor was stopped in social space in 8.1 % of the flights, in personal space in 58.6 % of the flights, and in intimate space in 33.3 % of the flights, regardless of whether the safety measure intervened or not. The boxplots in Figure 4.10 depict all 101 flights that were stopped by user intervention, for all four directions, including all samples color-coded by the stop method. The average stop distance was $\bar{x} = 0.63$ m ($\sigma = 0.33$ m). The average foot and hand stop distances were $\bar{x} = 0.73$ m ($\sigma = 0.36$ m), and $\bar{x} = 0.47$ m ($\sigma = 0.09$ m), respectively. The average stopping distances based on direction and stopping method are shown in Figure 4.11. The means and standard deviations of the stop distances in each direction by stop method are presented in Table 4.1.

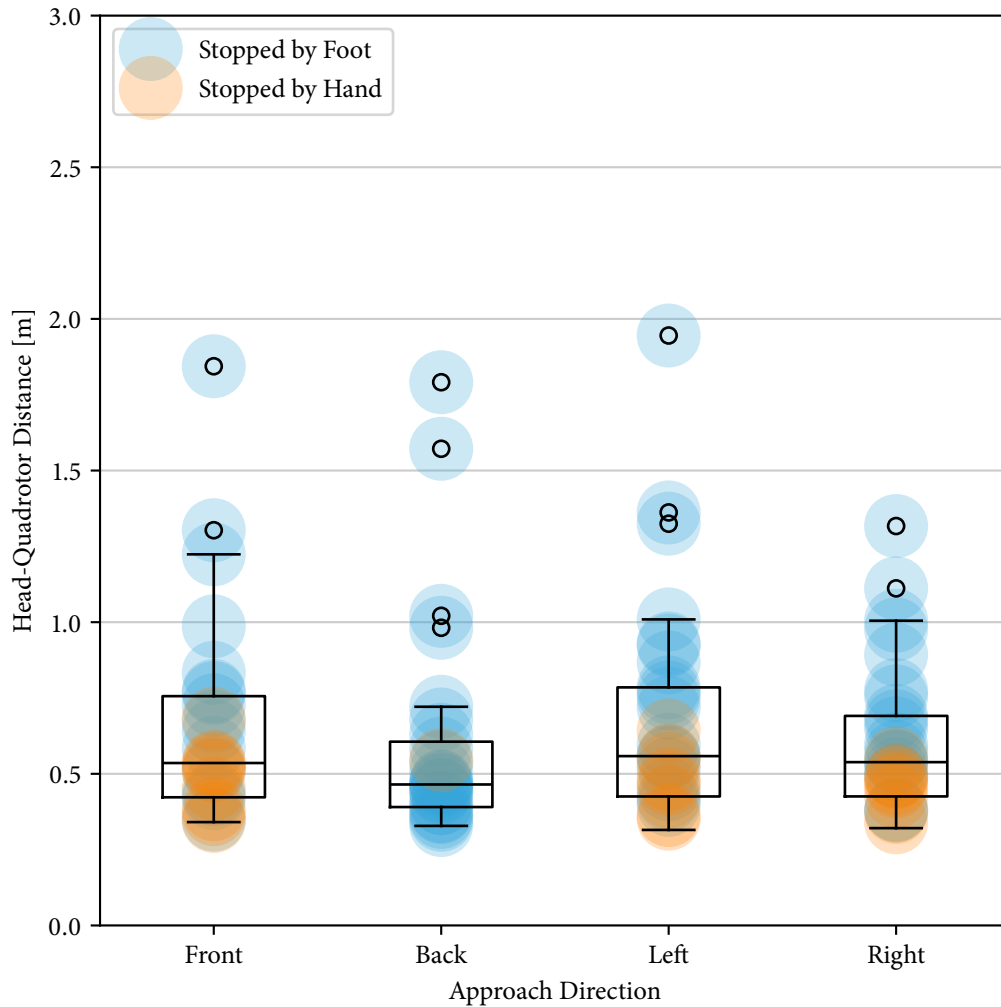


Figure 4.10: Boxplots depict the statistical evaluation of the numerical approach stopping distance experiment for each approach direction and stopping method. They are generated based on blue and orange sample points, representing the distance at which participants halted the drone's further approach. Blue samples denote the use of the foot switch to stop the quadrotor, while orange samples indicate physical intervention by tapping the quadrotor's frame.

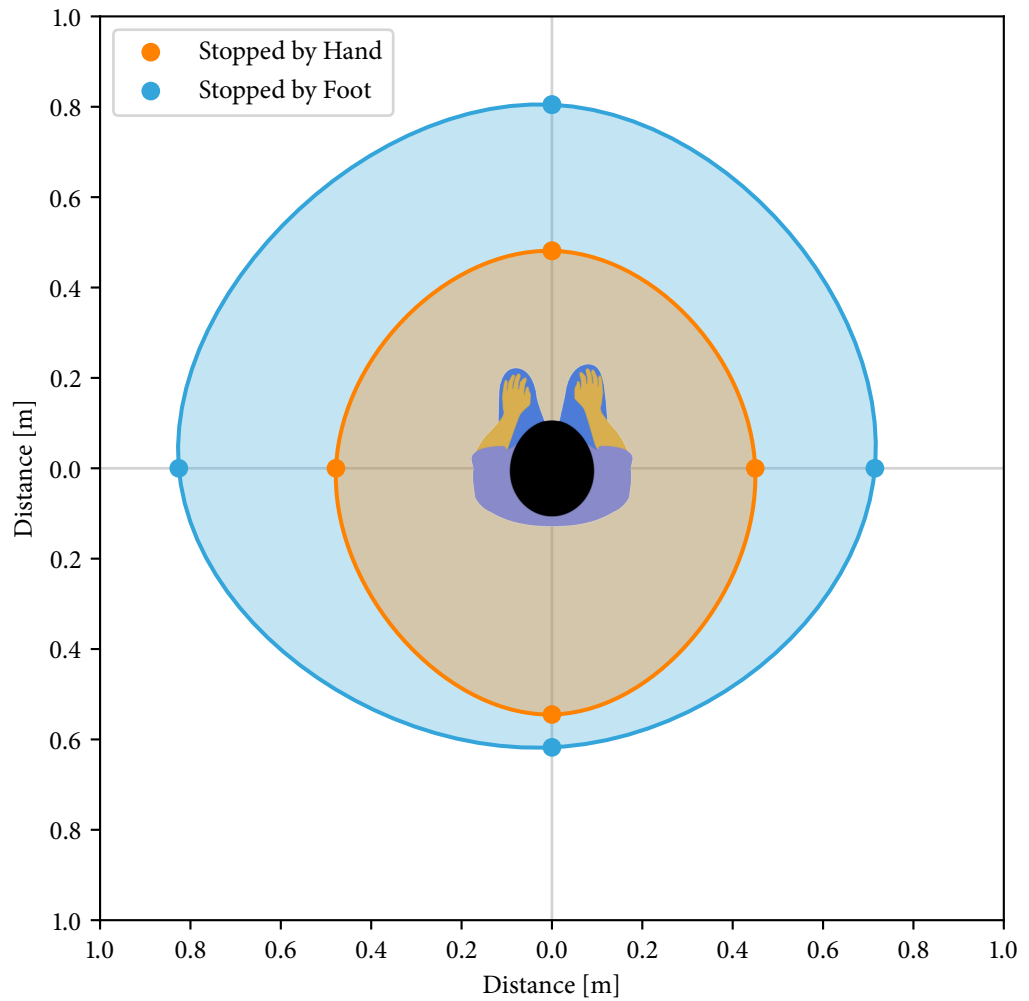


Figure 4.11: Average stopping distances based on direction and stopping method.

Assessing the *approach stop method* unveiled that out of the 128 flights conducted, the quadrotor was stopped in 68.0 % of the cases by using the foot switch, in 24.2 % of the cases by hand, and in 7.8 % of the cases it was not stopped at all. A total of 15 out of 32 participants attempted to stop the quadrotor by hand at least once. All female participants (100.0 %) stopped the quadrotor at least once by foot and 55.0 % at least once by hand. Among male participants, 81.0 % stopped the quadrotor at least once by foot and 81.0 % at least once by hand.

Table 4.1: Average and standard deviation of the stop distances given in meters for each approach direction.

Statistics		Approach Direction			
		Front	Back	Left	Right
Foot	\bar{x}	0.80	0.62	0.83	0.71
	σ	0.38	0.37	0.38	0.25
Hand	\bar{x}	0.48	0.54	0.48	0.45
	σ	0.10	0.00	0.09	0.06

Table 4.2: The results of a binomial test for the proportion of hand stops in manually stopped flights overall and for each approach direction. The quadrotor was found to be significantly less often stopped by hand than by foot ($p < 0.05$, underlined) in flights approaching from the back, from the left side, and overall across all flights.

	Approach Direction				
	Front	Back	Left	Right	All
Hand-stop ratio	0.40	<u>0.07</u>	<u>0.28</u>	0.31	<u>0.26</u>

Of the flights that approached from the front, 56.3 % were stopped by foot and 37.5 % by hand. Flights approaching from the back were stopped in 87.5 % of the cases by foot and 6.3 % by hand. Flights from left and right were stopped by foot in 65.6 % and 62.5 % of cases, respectively, and by hand in 25.0 % and 28.1 % of cases, respectively.

The proportions of hand-stopped flights to all manually stopped flights for all approach directions and in total can be found in Table 4.2. In total, the quadrotor was significantly less frequently stopped by hand than by foot.

Out of the 15 participants who used their hand to stop the quadrotor, eight (53.3 %) had prior experience with drones. On the other hand, six participants (21.4 %) from the group that used their foot to stop the quadrotor stated that they owned a service robot or a drone. Among the hand-stopping group, 40.0 % (six participants) reported owning a robot or a drone. There were no significant correlations found

Table 4.3: Means and standard deviations of participant emotions assessed through the 5-point Likert scale SAM surveys. Post-flight surveys are compared to the initial surveys using a Wilcoxon signed-rank test; the significant results ($p < 0.05$) are underlined.

Statistics		Post-Flight				
		Initial	Front	Back	Left	Right
Valence	\bar{x}	3.91	4.09	3.69	3.94	4.03
	σ	0.80	0.76	1.07	0.83	0.85
Arousal	\bar{x}	2.59	2.91	<u>3.28</u>	2.88	<u>2.94</u>
	σ	0.78	1.04	1.04	0.89	0.90
Dominance	\bar{x}	2.56	<u>3.12</u>	2.62	2.69	2.81
	σ	0.79	0.78	0.93	0.73	0.81

between previous experience or ownership of a robot or drone and the stop method or distance. Most of the participants (14 out of 15) who stopped the quadrotor by hand tapped the front of the quadrotor's frame. The remaining participant tapped the quadrotor from the right.

The impact of the *approach direction* on the participants was analyzed through self-assessment using the SAM scales. Emotional states were evaluated by comparing participants' initially reported states to those reported after each approach direction. This evaluation was conducted using a Wilcoxon signed-rank test. The results indicated that participants felt a higher level of dominance ($w = 5.0$, $p = 0.002$) when the quadrotor approached from the front. Furthermore, the participants' arousal levels were increased when the quadrotor approached from the back ($w = 23.5$, $p = 0.002$) and from the right ($w = 41.0$, $p = 0.037$). However, there were no statistically significant results for the other directions and emotional states. Table 4.3 shows the evaluation results including means, standard deviations, and statistically significant results from the Wilcoxon signed-rank test.

The participants' wellbeing after each flight was assessed using a 5-point Likert scale and compared between one approach direction and the other three directions using an independent two-sample t -test. It was found that the participants felt less comfortable when the quadrotor approached from the back ($t = -2.13$, $p = 0.035$). Additionally, after the flight approaching from the back was completed and

participants were able to turn around and look at the quadrotor, they perceived the final position of the quadrotor to be too close ($t = -3.23$, $p = 0.002$).

4.2.3 Discussion

Human-quadrotor proxemics studies are heavily influenced by parameters such as the quadrotor's size, which affects factors like weight, rotor size, noise, and downwash—all of which have a significant impact on participants' emotional states. The stationary state of the quadrotor—whether attached to a rail on the ceiling, actively flying, or equipped with rotors protected by a frame—contributes to varied individual emotional reactions. Additionally, the size and design of the room influence perceived proxemics. The variability in room conditions makes direct comparisons even more challenging. The wide range of non-standardizable parameters of the drone proxemics studies discussed in Section 4.2.1 highlights the difficulty in achieving comparability across current user studies in HDI.

Room

In this study, although a seating scenario was targeted to simulate environments commonly found at home or work, no context-specific furnished room was provided. However, as the primary focus of this study was on estimating distances without introducing a specific context, it was conducted directly in the flight laboratory. It is acknowledged that room size and design can influence the emotions of subjects, and future studies should consider adapting these aspects to the specific context under investigation.

Approach Stop Distance

The average distances at which participants stopped the quadrotor using their foot or hand are displayed in Figure 4.12. In comparison to related quadrotor proxemics studies, the average stop distance found in this study is the closest to the participants. In comparison to the user study conducted by Acharya et al. [ABD17], no increase in mental stress was observed for the quadrotor approaching from

the front. The absence of mental stress, as well as the closer distance, can likely be attributed to the relatively small size of the quadrotor used in this study.

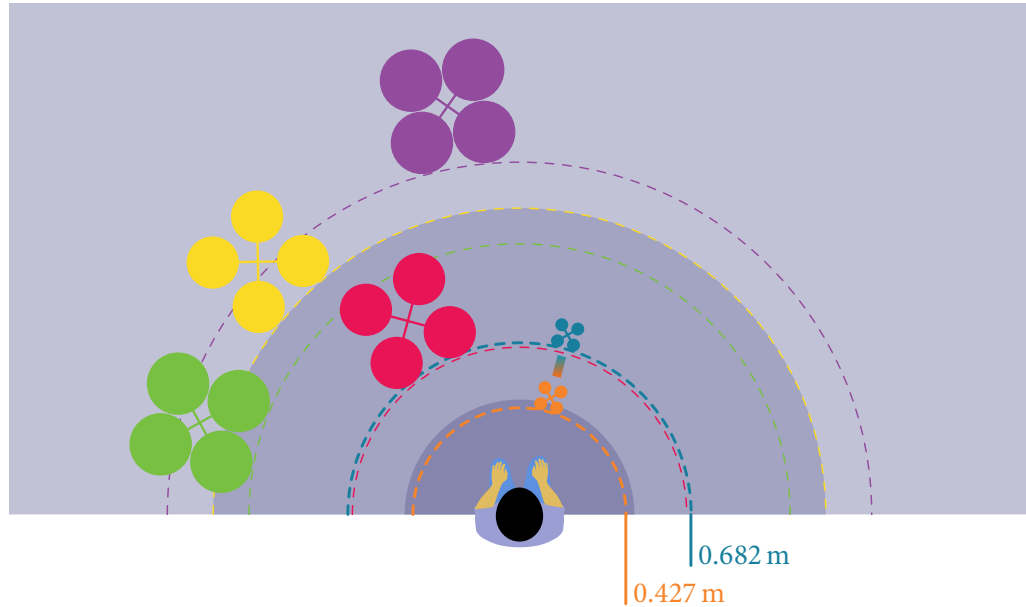


Figure 4.12: This figure illustrates the results of the numerical proxemics evaluation, i. e., the average stopping distances for a Crazyflie 2.1 when **stopped by hand** and **by foot switch**.

Approach Stop Method

Most participants perceived the foot switch as a safer option to stop the quadrotor from approaching further. As they stated, they did not want to touch the rotors and risk causing a crash. None of the participants was scared of the quadrotor due to its small scale, only one female participant stated, that the quadrotor would be uncanny. Some participants preferred to stop the quadrotor by hand as they believed it was a more straightforward, direct method. Some participants continued tapping the quadrotor's frame because they found it enjoyable.

Approach Direction

The findings of this study, which include increased dominance when the quadrotor approached from the front and decreased comfort and increased arousal when

approaching from the back, are consistent with a previous study by Wojciechowska et al. [Woj+19], which used flights from the front, front-right, and back. However, in the study conducted here, participants were unable to observe the quadrotor approaching from the back. Although participants stopped the drone themselves, they estimated the final stopping distance to be too close when approaching from behind, suggesting that they cannot realistically estimate the distance without looking at the drone.

Skeleton Tracking

During experiments with the Kinect 2 skeleton tracking and the testbed pose estimation system, a constant horizontal offset of the head position ($x = 62.4$ mm, $y = -31.1$ mm) was observed. This horizontal error of 69.6 mm is consistent with the mean error determined by Wang et al. [Wan+15], who compared the skeleton tracking accuracy of the Kinect 1 and Kinect 2 with a professional motion capture system. Depending on the angle between the user and the Kinect, the mean error of the estimated head position ranged from 62–79 mm at the distance where the participants were seated.

During the user study, only the participants' head positions were monitored using the Kinect 2. The horizontal offset was applied solely to the logged head position for the evaluation. All measurements reported include the horizontal mean error offset. However, the head safety measure during the user study was determined without the offset, which caused the quadrotor to stop earlier than intended. As a result, 17 flights (where the safety measure interrupted participants who were just on course to stop the flights) could have been used for the numerical distance evaluation. This would have resulted in the quadrotor's stopping distance being even closer to the participants' intimate space, namely at $\bar{x} = 0.59$ m ($\sigma = 0.33$ m).

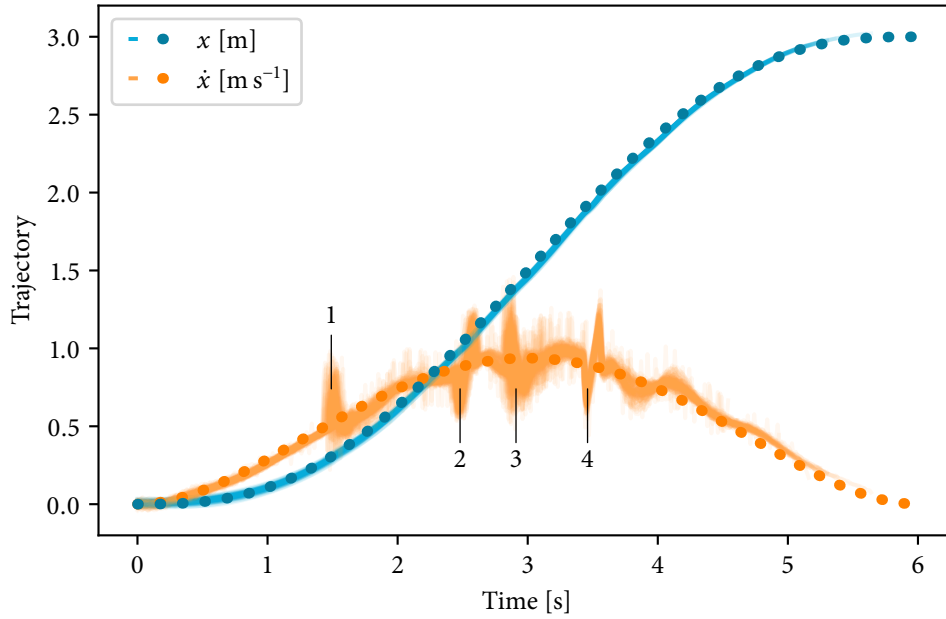
The participants' head positions at the time when the quadrotor came to a halt have been evaluated, but all skeleton data throughout all flights has been collected. The analysis of this data over time could provide further insights into participant behavior, e. g., whether they backed away as the drone approached.

Quadrotor Pose Estimation

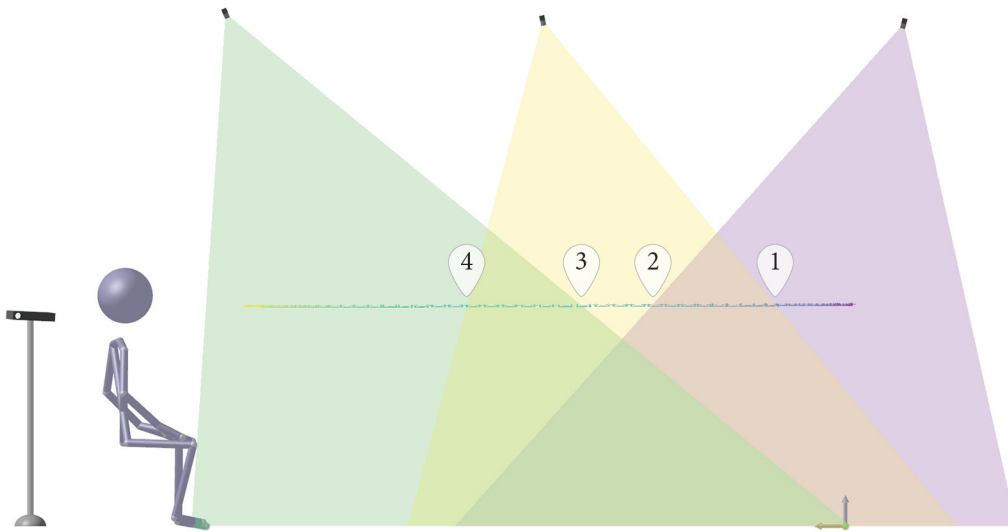
The root-mean-square error (RMSE) of all flights was 49.5 mm. For the x , y and z coordinates, the RMSE was 28.6 mm, 13.9 mm and 37.9 mm, respectively. Slight noise and twitching of the drone were observed in certain trajectories. This issue is noticeable in Figure 4.13a, which illustrates the x positions and velocities of all flown trajectories, including the reference trajectory. During transitions between the view frustums of the pose estimation cameras, errors in velocity arise. These errors occur because the quadrotor's velocity is numerically derived from positional pose estimation data. At these transitions, optical pose estimation errors are more pronounced at the edges of the camera image due to lens distortions.

As outlined in Section 3.1.1, all cameras are intentionally overlapped at the edges of their view frustums. This arrangement facilitates pairwise stereo-calibration into the coordinate frame of the reference camera. The estimated poses of two cameras at their overlapping image borders become more contradictory during these transitions, as the corresponding errors are larger in that region.

Despite employing sophisticated calibration routines, hardware synchronization of the pose estimation cameras, and time-dependent averaging of poses, small errors in pose estimation lead to inaccuracies in the numerically derived velocity of the quadrotor. Consequently, this results in slight instabilities during camera transitions, as indicated in Figures 4.13a and 4.13b. These issues were subsequently addressed and resolved using the algorithm described in Section 3.1.4.



(a) Positions and velocities along the x coordinate for all 128 flights conducted in this experiment, depicted as continuous lines, and the reference trajectory, represented by dotted lines. The y and z coordinates remain constant and are not displayed.



(b) Rendered user study session with the reference trajectory of a quadrotor approaching from the front. Initially, the quadrotor is only visible to the right camera until it moves into the view frustums of the right and middle cameras between marks 1 and 2. From marks 2 to 3, it is solely seen by the middle camera, and at mark 3, it enters the view frustum of the left camera. Finally, at mark 4, it exits the view of the middle camera and is only visible to the left camera.

Figure 4.13: Trajectory evaluation reveals numerical issues in velocity estimation (a) and shows transitions between camera frustums (b) where these issues occur. The numbers indicate transitions between view frustums and correspond across figures.

4.3 HAPTICS

Haptic interaction involves nonverbal communication through touch, including gestures like handshakes and pats on the shoulder. These tactile interactions convey diverse social and emotional messages, from greetings to expressing joy and offering support. Touch plays a crucial role in building social bonds and is an early-developing sense, making it one of the most instinctive and intuitive forms of interaction [CWE99].

To facilitate such interactions between humans and drones, it is necessary to lower the inhibition threshold and develop innovative techniques that feel more natural than conventional methods such as radio remote controls or smartphones. Even with drones, participants will naturally employ touch as a mode of interaction [Abt+17], which could be supported by the findings of the proxemics experiment described in Section 4.2.2.

Inspired by a way of interaction that does not require additional equipment such as sensors, a novel approach to control quadrotors is employed: utilizing 3D touch interaction through virtual buttons. Quadrotors are equipped with IMUs, an essential component in the guidance and control of uncrewed vehicles. IMUs are electronic devices that combine accelerometers and gyroscopes to measure linear accelerations and angular velocities, respectively, for each of the three principal axes. Sometimes IMUs include a magnetometer determining magnetic North used as a heading reference. In this study, the accelerometer data from the IMU integrated into the flight control PCB of the drone is utilized to facilitate physical interaction between humans and quadrotors without the need for intermediary devices.

By incorporating 3D touch interaction capabilities, it becomes possible to apply familiar human gestures, e. g., to last-mile UAV delivery scenarios. The act of tapping the side of a truck twice to signal the driver that they can drive off could be adapted for UAV delivery. This gesture indicates that the vehicle has been loaded or unloaded and is ready for departure. Similarly, the gesture of patting someone's shoulder as a way of acknowledging a job well done can also be transferred to flying robots. This is particularly relevant as studies have observed that people tend to interact with UAVs in a manner similar to how they interact with humans or pets [Cau+15].

4.3.1 Touch Detection

In order to ensure smooth and continuous motion of a vehicle, trajectory generators typically aim to minimize jerk, which refers to the third derivative of the position vector or the rate of change of acceleration over time. Consequently, under normal flight conditions where external forces are minimal, jerk is rarely observed. If jerk is observed, however, it may be due to a collision or a user input, for example. In the following, the method used to detect touch input in the accelerometer data of a commercially available IMU is described.

A mechanical accelerometer consists of a proof mass on a spring-damper system within a frame. Acceleration can be measured by observing spring compression when forces act on the frame, as schematically illustrated in Figure 4.14. When the

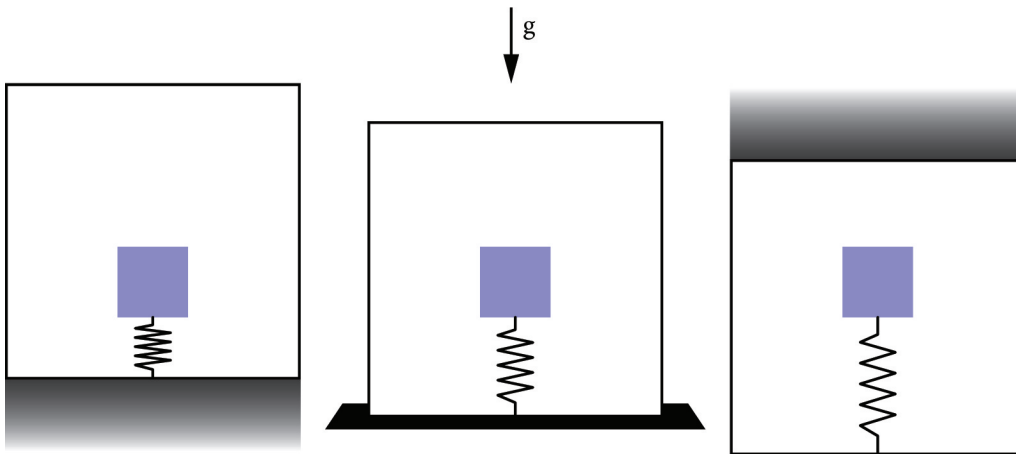


Figure 4.14: The mechanical accelerometer comprises a frame and a proof mass on a spring-damper system. When the frame is accelerated upward (left), the springs compress due to the inertia of the proof mass. At rest (center), the springs are compressed by gravity acting on the proof mass. Accelerating downwards (right) results in elongated springs.

frame is accelerated upward, the springs compress due to the inertia of the proof mass. During downward acceleration, the springs elongate. At rest—meaning the sensor is placed on a surface with no external forces acting on it—the springs are compressed due to gravity acting on the proof mass, similar to the compression observed during upward acceleration.

Accelerometers provide measurement data in G, representing gravitational acceleration, “G-force”, measured relative to freefall. Thus, during freefall, the accelerome-

ter registers 0 G. At rest it records 1 G straight upward, which on Earth is equivalent to approximately $9.806\ 65\ \text{m s}^{-2}$.

To measure accelerations in three dimensions, triaxial assemblies of accelerometers are employed. Modern accelerometers, such as those used on drones, are micro-electromechanical system (MEMS) that detect mechanical changes and convert them into electrical information. Excluding external disturbances, a hovering drone is also in a stationary state, and the accelerometer registers 1 G, just as it does when at rest.

As with all electrical sensors, accelerometer data are subject to measurement noise, which is typically caused by electromagnetic interference (EMI). To minimize the effects of measurement noise, a low-pass filter is applied to the accelerometer data, i. e.,

$$\ddot{\mathbf{x}}'_k = \alpha \ddot{\mathbf{x}}_k + (1 - \alpha) \ddot{\mathbf{x}}'_{k-1}, \quad \alpha = \frac{\Delta t}{\tau + \Delta t},$$

where $\ddot{\mathbf{x}}_k$ represents the acceleration vector at a specific discrete time k , while τ denotes an empirically determined time constant for two consecutive measurements separated in time by Δt . The jerk vector $\ddot{\mathbf{x}}_k$ at time step k is then obtained by deriving it from the filtered acceleration vector:

$$\ddot{\mathbf{x}}_k = \frac{\ddot{\mathbf{x}}'_k - \ddot{\mathbf{x}}'_{k-1}}{dt}.$$

When the magnitude of the jerk vector exceeds a predetermined threshold t_{jerk} , it indicates the detection of a potential tap, i. e., if

$$\|\ddot{\mathbf{x}}_k\| > t_{\text{jerk}}.$$

The direction of the jerk vector at the start of the acceleration is consistent with the tap direction, as jerk is the change in acceleration. At the end of the acceleration, the jerk vector points in the opposite direction. To detect the jerk only at the beginning of the acceleration, a cool-down time period t_{cool} is introduced in which the detection is suspended to prevent the algorithm from double detection or detecting the wrong direction.

To distinguish different tap directions, e. g., to trigger different reactions of the quadrotor, a virtual button metaphor is introduced. For this, a set of m unit vectors is defined in the quadrotor's body frame

$$\mathcal{R} = \{\mathbf{r}_i, \in \mathbb{R}^3 : \|\mathbf{r}_i\| = 1, i = 1, \dots, m\}.$$

As an example, Figure 4.15 shows four virtual buttons defined along the directions of the \times -shaped frame of a quadrotor, along with acceleration data resulting from several interactions.

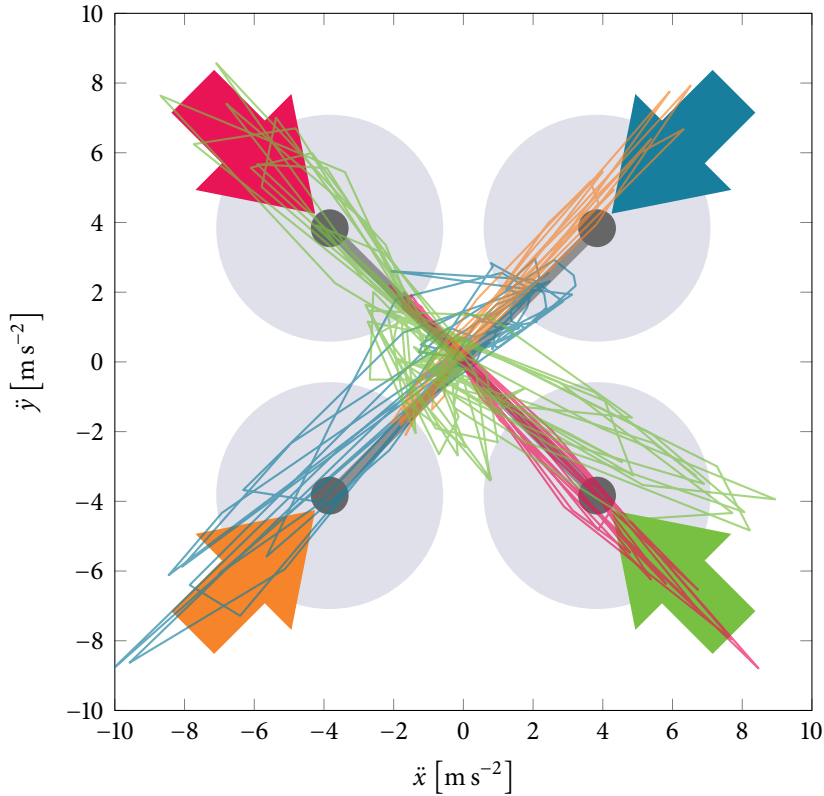


Figure 4.15: Unfiltered accelerometer data from several operations of four designated virtual buttons, represented by arrows indicating their normal vector of operation. The color coding links the accelerometer data to the operated, virtual button.

The identification of the operated virtual button is accomplished by maximizing the cosine similarity between the normalized detected jerk direction vector and the set of virtual buttons, i. e.,

$$\mathbf{r}_k = \arg \max_{\mathbf{r} \in \mathcal{R}} \mathbf{r} \cdot \hat{\mathbf{x}}_k, \quad \hat{\mathbf{x}}_k = \frac{\ddot{\mathbf{x}}_k}{\|\ddot{\mathbf{x}}_k\|}.$$

This identifies the button that is “closest” to the tap direction, but may still deviate from the actual tap direction. To narrow down the selection more precisely, an additional similarity threshold t_{cos} is introduced. If $\mathbf{r}_k \cdot \hat{\mathbf{x}}_k$ exceeds this threshold, a tap with intensity of $\|\ddot{\mathbf{x}}_k\|$ is detected along \mathbf{r}_k .

4.3.2 Applications

The kind of interaction the proposed virtual buttons can be used for is arbitrary and versatile. Similar to single or double clicks on a computer mouse, different inputs can be distinguished. To achieve this, the jerk detection can be extended accordingly. Concrete examples include basic double-tap detection and dodging in the detected direction, both of which are parts of the demonstrator scenario described at the beginning of this chapter. To better contextualize the proposed interaction method within a framework of enhancing human-robot companionships, a playful multi-tap interaction scenario is created and described thereafter.

Double-tap Interaction

The introductory example from Chapter 4 described the scenario in which double-tapping the quadrotor should prompt it to dodge in the corresponding direction. If the corresponding flight is possible, i. e., if the desired target position is within the tracking volume, it should confirm and execute the movement. If the flight is not possible, the drone should deny and hover in place. This scenario is once again depicted in more detail in Figure 4.16. For detecting double-taps, the time points $t_i, i \in \{0,1\}$ of two consecutive taps to the same button must satisfy the condition

$$t_{\text{cool}} < t_1 - t_0 < t_{\text{delay}},$$

where the maximum delay t_{delay} required for the taps to be interpreted as a double-tap was set to $t_{\text{delay}} = 1.0$ s. The cool-down period, introduced in Section 4.3.1 to prevent false jerk detection, was set to $t_{\text{cool}} = 0.2$ s. Both parameters were determined through experimentation.

The displaced target position \mathbf{x}_r in the reference frame $\{r\}$ is given by

$$\mathbf{x}_r = \mathbf{p}_r + \mathbf{q}_{rb} \odot \mathbf{d} \mathbf{r}_k,$$

where \mathbf{p}_r and \mathbf{q}_{rb} are the quadrotors current position and attitude, \mathbf{d} is a displacement constant set to $\mathbf{d} = 0.3$ m, and \mathbf{r}_k is the normal vector of the operated virtual button.

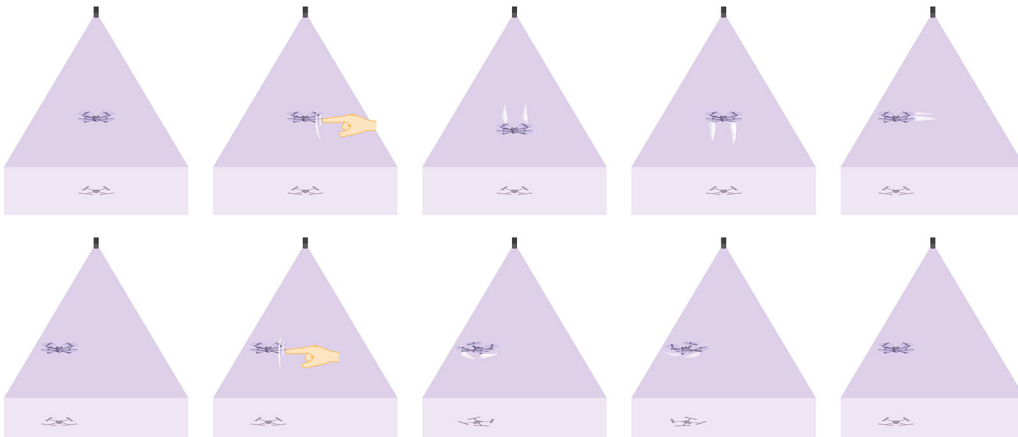


Figure 4.16: Tap interaction: A quadrotor hovers at the center of the camera’s view frustum when the user double-taps its frame from the right to instruct the quadrotor to move. The quadrotor acknowledges the command with a downward and upward motion before proceeding to fly in the opposite direction of the tap. At this position, the quadrotor is double-tapped again. However, since the targeted trajectory would extend beyond the camera’s view frustum, the quadrotor denies the command with a yaw motion and remains at its current position.

Multi-Tap Interaction: MetroDrone

Socially expressive ground robots enjoy wide acceptance due to their movements being easily comprehensible to humans. Music listening companions such as the Sony Rolly [KKK09] and the Keepon [KMN09] have found applications in research, therapy, and entertainment. Studies have demonstrated that when robots respond to musical experiences, humans tend to attribute human-like characteristics to the robot and perceive it as more similar to themselves [HV13]. Moreover, the shared experience of emotions evoked by music serves to strengthen the emotional bond between humans [Koe13; HBV16]. The interaction scenario proposed hereafter is named *MetroDrone* and is designed specifically for emotionally connecting a human and a flying robot through a musical experience. In Figure 4.17, a brief sequence of the suggested interaction scenario is depicted, wherein the user taps the quadrotor’s frame four times to a beat, and in response, the quadrotor flies a trajectory that synchronizes with the beat.

Rather than relying on software to detect the song or the beat, this scenario focuses on creating an interactive experience where humans and quadrotors can enjoy music together. To achieve this, the proposed approach utilizes close proximity

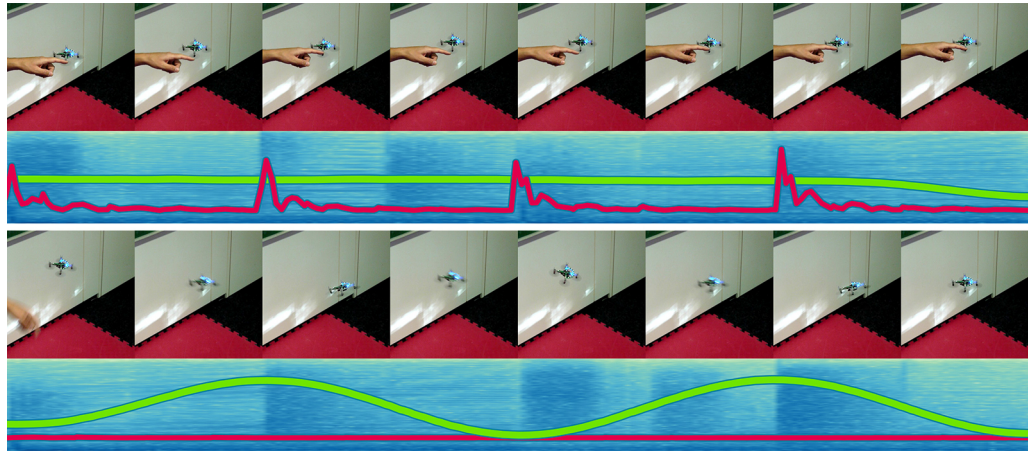


Figure 4.17: Tactile HDI sequence with *MetroDrone*: The user taps the frame of the airborne vehicle several times to the beat of the music. The quadrotor’s onboard accelerometer measures changes in acceleration and an algorithm detects their frequency and direction. Based on this information, a trajectory is generated to synchronize a the quadrotor’s movement with the rhythm.

interaction through touch, eliminating the need for intermediary devices. This allows the user to teach the quadrotor the song’s tempo, measured in beats per minute (BPM). Consequently, the quadrotor performs a brief dance, creating the impression of an emotional response to the music. This approach adds a touch of human-like behavior to the quadrotor, making it seem less mechanical and aiming to lower the inhibition threshold.

The following are insights into the application and adaption of the jerk detection, a description of the beat detection, and the generation of the specific trajectories. In the example, a Crazyflie 2.1 drone is used, which is equipped with a Bosch Sensortech BMI088³, a six-axis inertial sensor. The processing computer receives the quadrotor’s accelerometer data via the Crazyradio PA at a frequency of 100 Hz. For jerk detection, the jerk peak threshold was set to $t_{\text{jerk}} = 400 \text{ m s}^{-3}$ and the low-pass filter time constant to $\tau = 0.0025$. The detection suspension threshold was specified at $t_{\text{cool}} = 0.2 \text{ s}$ and allows the algorithm to detect beats of up to 300 bpm. A cosine similarity threshold of $t_{\text{cos}} = \cos(\pi/8)$ was used. All parameters were determined through experimentation.

³<https://www.bosch-sensortec.com/products/motion-sensors/imus/bmi088>

Four variations of rhythmic trajectories depicting repetitive patterns were generated and are depicted in Figure 4.18. A basic, scalar periodic trajectory \mathcal{K}_1 that

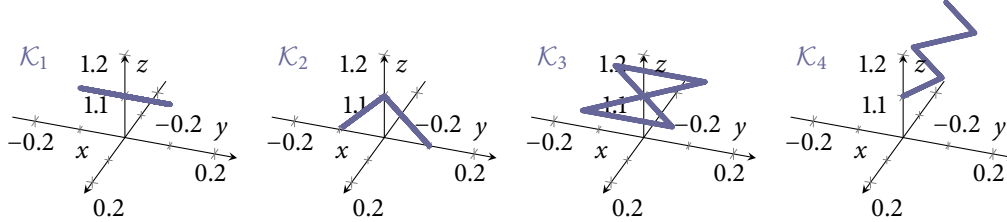


Figure 4.18: Plots showing coordinates of four different periodic trajectories (all dimensions are in meters).

implements a side-to-side motion at $f = 2 \text{ Hz}$ (120 bpm) is constructed using the keyframes

$$\mathbf{s}_i(t_i) = \begin{pmatrix} (-1)^i \cdot d \\ 0 \\ 0 \end{pmatrix}, \quad t_i = \frac{i}{f}, \quad i \in \{1, \dots, n\}$$

where the displacement was set to $d = 0.1 \text{ m}$.

To emphasize the beat, it is important for the quadrotor to come to a rest at the keyframes. Therefore, velocities and accelerations are set to zero at these time points. The trajectory comprises eight keyframes (divided into $n = 7$ segments) to ensure it aligns with the 4/4 time signature. In order to match the update rate of the flight controller, the trajectory is sampled at a rate of $f_s = 100 \text{ Hz}$. Similarly, the other trajectories depicted in Figure 4.18 and their projections in Figure 4.19 are constructed using distinct keyframes while following a similar approach. Their flights are initiated by the beat detector, depending on which button the user presses.

The implemented beat detector collects the time points t_i of detected taps, i. e., jerk values exceeding the defined peak threshold t_{jerk} . These taps are detected in the four specified normal directions \mathbf{r}_i , which point toward the quadrotor’s center of mass, as illustrated in Figure 4.15. The intervals between successive time points are converted into BPMs.

If the standard deviation of the BPM values within a discrete time window falls below a specified threshold, the beat detector triggers one of the four rhythmic trajectories. For the *MetroDrone* scenario, a standard deviation of 3 bpm and a

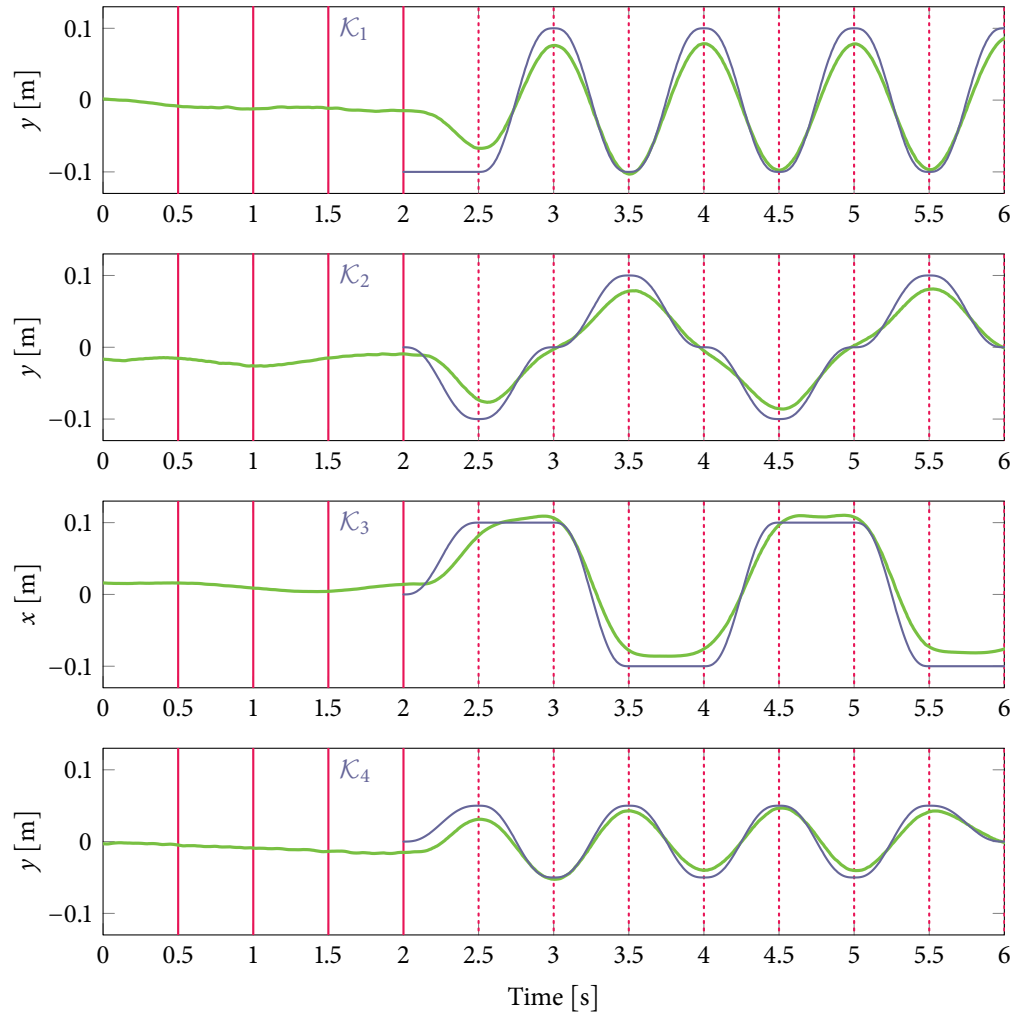


Figure 4.19: *MetroDrone* interaction sequence: Four taps to a beat at 120 bpm in 4/4 time signature (red), the propagated beat (dotted) to which one of the trajectories \mathcal{K}_i , $i \in \{1, \dots, 4\}$ (purple) is initiated, and a selected coordinate of the quadrotor's position (green) during the tracking of the generated trajectory.

window size of four is used. This window size thus matches 4/4 time signatures, also known as “common time”, as it is the most common time signature in music. It indicates that there are four beats in each measure, and the quarter note (1/4) receives one beat. When the user taps four times, each tap corresponds to one of the four beats in a measure. Subsequently, the trajectory is initiated, synchronizing with the beginning of the following measure.

Upon detecting the fourth tap, the trajectory is generated and executed precisely in time for the upcoming beat. To synchronize the start of the trajectory, the time it takes for trajectory calculation is subtracted. Due to the short duration of the generated trajectories that enable a more frequent interaction, there is no need for synchronization that exceeds the initial start of the trajectory. The interaction sequence of the side-to-side response is depicted in Figure 4.17. Videos of *MetroDrone* interaction sequences are published online from first person⁴ and third person⁵ perspective. For a more comprehensive exploration of synchronizing quadrotors to music, i. e., continuously aligning a quadrotor’s motion with sudden changes in reference amplitude and frequency, refer to Schöllig et al. [Sch+10].

4.3.3 Discussion

The presented interaction possibilities serve as a proof of concept for basic physical interaction with quadrotors. In the multi-tap interaction scenario *MetroDrone*, four virtual buttons positioned in the quadrotor’s horizontal plane, as depicted in Figure 4.15, were proposed and implemented as they align with the default frame. They limit the quadrotor’s interaction to taps within the drone’s xy plane. When external vertical forces are applied to points that do not coincide with the quadrotor’s center of mass, the thrust vector tilts, leading to instability in its attitude. For instance, if a downward force is applied to one of the frame’s arms, the quadrotor (including its thrust vector) would tilt and move toward the user’s finger, which is clearly undesirable. While the four proposed buttons can be used with most quadrotors, it is possible to define buttons virtually anywhere as long as a suitable frame or housing is available (e. g., the frame proposed in Figure 4.8).

⁴<https://youtu.be/fiCwqzmSuAg>

⁵<https://youtu.be/HPDFXabEjbI>

Using the cosine similarity between the jerk direction at the detected peak and a set of predefined normal vectors provided in the quadrotor's body frame, several interaction modes can be distinguished. These interaction modes again can be subdivided by grading tap intensities, that have not been taken into account in the implementation. If it is desired to react to the intensity of the button operation, e. g., by wider movements, the magnitude of the jerk can be taken into account.

In addition to the proposed interaction scenarios, however, a single detected jerk can also mean collision with an obstacle. On this basis, it would be possible to dodge in the opposite direction of the hit or mark the locations of the collision in a map. Based on a single jerk detection, the quadrotor could react, e. g., by dodging or moving away from a collision source, similar to the early basic vacuum cleaner autonomy and to create a map from hits.

4.4 VOCALICS

While visual cues, such as gestures embedded in trajectories, are more prominent and have been extensively researched, acoustic cues remain largely unexplored, despite their potential to expand nonverbal communication channels in HDI. One likely reason for this is the rotor noise that is already emitted as a result of the drone's operation, referred to as its *consequential sound*. Since humans perceive the consequential sound as unpleasant noise, it negatively impacts their interaction experience [Cau+15; Jon+16; Col+17; CCC17; Kni+18]. Adding a speaker to a drone to add natural sounds, such as birdsong or rain, increases the pleasantness of the interaction, but also increases the perceived loudness [Wan+23], as the added sound must compete with the drone's consequential sound

In order to reduce noise pollution for humans and animals, the reduction of the acoustic signature of rotorcraft is the subject of ongoing research. Measures against noise pollution include the development of active noise cancellation (ANC) solutions [Nar+20] and the exploration of new rotor blade geometries such as the toroidal rotor [SS20]. However, neither of these technologies has reached the consumer stage as of yet.

While the majority of people may not find value in the consequential sound of a drone, there are groups that can benefit from the rotor sound, such as visually impaired people who can be aurally navigated using the rotor sound of a quadrotor [AFH15; Al +16; Avi+17; AF18]. For this reason, the augmentation or shaping of a drone's rotor sound should be exploited and evaluated for acoustic communication. Utilizing this pre-existing noise source resulting from the operation of rotary-wing drones avoids the additional payload that even a small loudspeaker entails. Additional audiovisual and sensory equipment is not always desired or practicable as it adds to the payload of a drone and decreases flight time. But the rotors themselves offer potential for nonverbal communication. With this minimalist approach, virtually any drone would be capable of using this form of communication. The addition of vocal cues to commonly used nonverbal channels, such as gestures, provides the opportunity for humans to understand the intent of the drone without having to keep it in sight during the interaction. Sometimes the drone is not even intended to be in the operator's line of sight, e. g., in current scenarios where drones are used to film athletes such as joggers.

This section proposes a vocalic approach by adding a higher-frequency oscillation to a quadrotor's consequential sound while maintaining the visually perceived flight characteristics. This method can be used to acoustically enhance drone flights for better differentiation of aerial gestures or, in its simplest form, to attract attention of the person accompanied by the drone. Just as a shaky voice adds auditory cues to a human voice, the proposed method adds information to a trajectory. Using onboard means, this method contributes to HDI by providing a new nonverbal communication channel, especially in situations where the person does not have the drone in sight or cannot see it, for example due to lack of daylight or visual impairment.

Several options come to mind for how a quadrotor could utilize the sound generated by its rotors as part of an interaction, such as using it to signal a basic yes or no. Probably one of the most intuitive approach may be to extract cues of the human voice and transfer this information to a quadrotor. One of the most present cues is pitch. Greetings typically have a rising intonation, while farewells feature a falling intonation; even infants recognize a sentence with a higher-pitched ending as a question [QS12]. The frequencies generated by the rotors of a quadrotor vary depending on the platform. For the Crazyflie 2.1 used in this study, the hover frequency is approximately 640 Hz. The human voice, however, typically falls within the range of 300–3000 Hz. As such, the rotors of this particular quadrotor cannot effectively reconstruct a voice signal but could be used for SFU, where limited frequency range is sufficient. Furthermore, they can still be utilized effectively for playing musical notes by modulating rotor speed to produce discrete pitches for musical expression.

As described in Section 3.3.3, the acoustically dominant frequency can be modelled as a quadratic function of rotor speed. A fitted function of quadrotor thrust input to the generated frequency, as well as its inverse function, can be seen in Figure 4.20. To play a music piece on a quadrotor, a musical instrument digital interface (MIDI) controller, such as a musical keyboard, can be used. With the function

$$f(n) = (\sqrt[12]{2})^{n-49}(440 \text{ Hz}),$$

the desired note frequency based on the key number n is determined [Wei05]. Through the second-order polynomial fit from Figure 4.20, the thrust to be generated can then be determined using

$$f(x) = 9.833 \times 10^{-7}x^2 + 5.595 \times 10^{-4}x - 1.157 \times 10^{-2}.$$

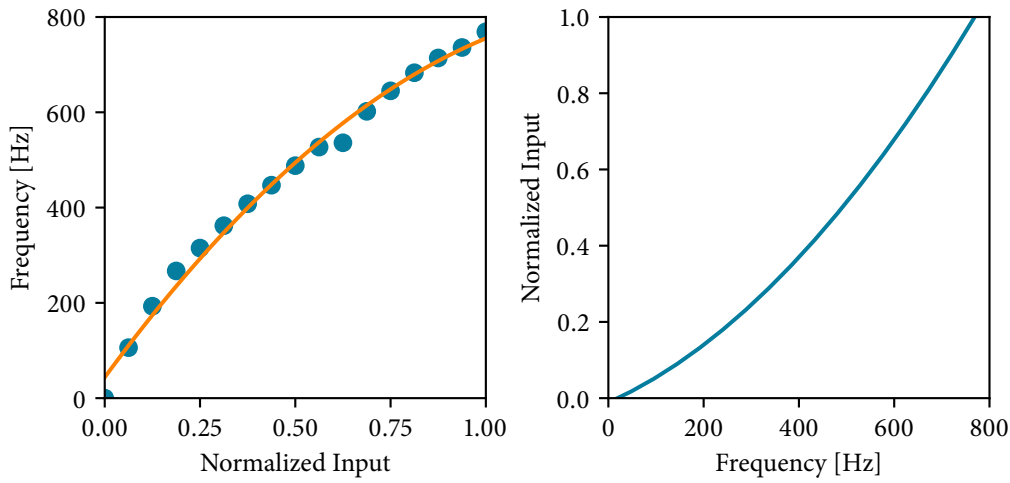


Figure 4.20: Second order polynomial fit of the normalized quadrotor thrust input to the generated audible frequency (left) and its inverse function (right).

The rise or fall of the drone, resulting from changes in rotor speed, constitutes a considerable side effect, making this type of interaction less effective, as the motion may distract from intended messages. To achieve stable altitude, the frequencies would need to be pulsed in a way that the quadrotor maintains its altitude, which would require the implementation of a special controller, which is left for future work.

However, it is conceivable to enhance an existing trajectory by temporarily superimposing a frequency, which can be accomplished with the given control algorithms. This enables a hovering drone to, e. g., communicate using Morse code in very complex cases or—since Morse code is hardly intuitive—just to signal yes or no, accompanying an aerial gesture.

Two fundamental human gestures used to communicate positive and negative feedback in Western cultures are nodding and shaking one's head, as described with the introductory example from Chapter 4. Head shaking is widespread to indicate disagreement, denial, rejection, disapproval, upset, while head nodding is often used for affirmation or approval. Transferring these movements to a drone through simple periodic up-and-down motions and alternating changes of the heading angle results in flights that sound similar, despite their opposite intent. Identifying them based on acoustic cues alone can be challenging. Especially in

scenarios where a person is unable to visually track the drone, being able to audibly distinguish its intent could help improve the interaction experience.

The remaining part of this section is organized as follows: The basic positive and negative airborne feedback gestures, mimicking human head nodding and shaking, are described in Section 4.4.1, along with their augmentation using the proposed method to enhance acoustic distinguishability. In Section 4.4.2, a detailed account of the conducted user study is provided, evaluating the described method. Finally, Section 4.4.3 discusses the study's results and limitations.

4.4.1 Airborne Gestures

As described in Section 1.3.2, quadrotors are underactuated systems, i. e., translational motion in the vertical plane is coupled to rotational motion about their roll and pitch axes. For this reason, mimicking human nodding as positive feedback is achieved with a simple up-and-down motion rather than by tilting about the drone's pitch axis [Cau+15; JHK18], as the latter, though closer to human nodding, would introduce horizontal motion. Mimicking human head shaking as negative feedback is done by repeatedly rotating the drone clockwise and counterclockwise about its yaw axis.

Both aerial gestures are based on a harmonic motion

$$h(t, f, p) = \sin(2\pi f \cdot t + p)$$

with time t , frequency f , and phase shift p . Mimicking head nodding as an up-and-down motion for the positive feedback trajectory, this function is scaled to model an altitude offset

$$dz(t) = 0.03 \cdot h(t, 1, \pi).$$

This offset is then applied to the drone's current altitude z . To start the nodding motion downward first, a phase shift of $p = \pi$ is applied. The negative feedback trajectory, which mimics head shaking, has the same characteristics and is scaled to

$$d\psi(t) = \frac{\pi}{6} \cdot h(t, 1, \pi).$$

It describes a heading angle offset to be added to the current heading ψ .

Flying these trajectories using a Crazyflie 2.1 (Figure 3.6) results in audible sounds that are hard to distinguish from each other. This similarity is evident in the STFT of a microphone recording of the performed flights, as shown in Figures 4.21a and 4.21b. The displayed frequencies correspond to those of the rotors. When

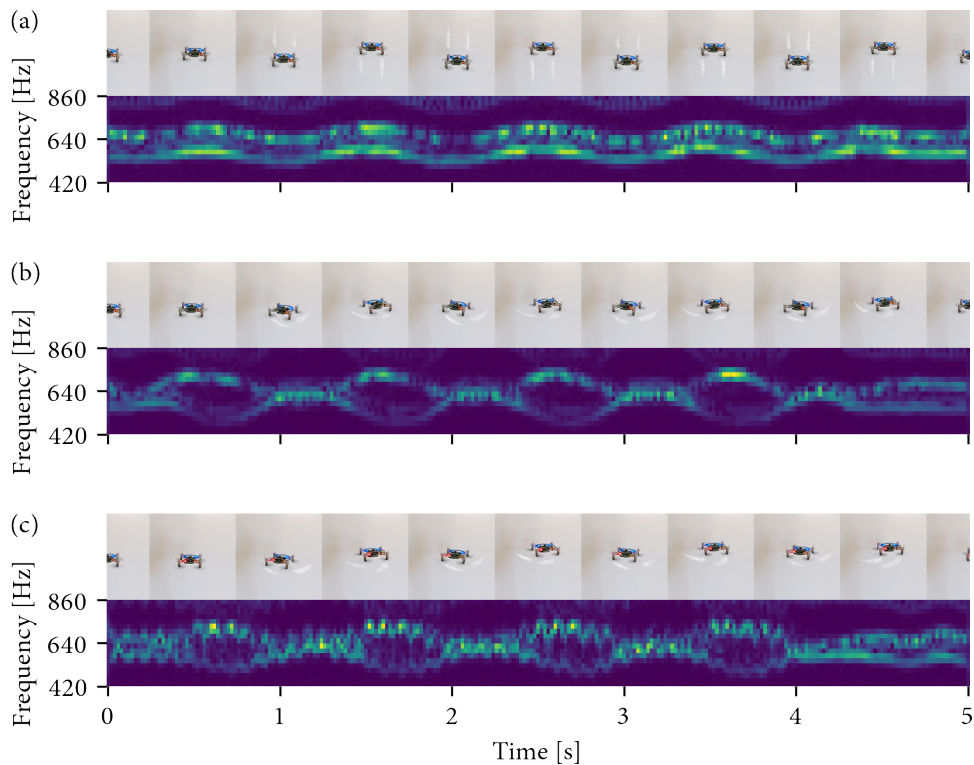


Figure 4.21: Stills from trajectories communicating positive feedback (a), *ordinary* negative feedback (b), and *vocalics* negative feedback (c) including relevant sections of the short-time Fourier transform of their microphone recordings. The motion of the drone is visually emphasized by illustrated tracks. Since the drone is equipped with two-bladed rotors, the measured frequency is twice the rotor speed.

a quadrotor performs a translational motion along its vertical axis, such as the implemented nodding aerial gesture, thrust is applied equally to all four rotors, causing upward acceleration and increasing their frequencies. When a quadrotor rotates about its vertical axis, as in the implemented head-shaking aerial gesture, more thrust is applied to one opposite pair of rotors, while less thrust is applied to the other pair.

Due to the similarity in thrust applied, both flights sound similar. To enhance the acoustic distinction between the negative feedback trajectory and the positive feedback trajectory, an additional 10 Hz harmonic motion

$$d\psi_{\text{voc}}(t) = d\psi(t) + \frac{\pi}{3} \cdot h(t, 10, 0)$$

is superimposed onto the negative feedback trajectory, resulting in the *vocalics* negative feedback trajectory, with the corresponding STFT of the sound recording shown in Figure 4.21c.

The relevant coordinates logged from the pose estimation system for all three trajectories are shown in Figure 4.22. The coordinates of the *ordinary* negative feedback trajectory (Figure 4.22b) and the *vocalics* negative feedback trajectory (Figure 4.22c) demonstrate that the visual impression of the flight remains largely unchanged, despite the superimposition of the clearly audible higher-frequency signal.

4.4.2 User Study

To evaluate whether the vocalic extension of gestures can add value in HDI by helping to audibly differentiate similar sounding trajectories, a drone performing the aerial gestures of the described positive feedback trajectory, the *ordinary* negative feedback trajectory, and the *vocalics* negative feedback trajectory were recorded. Video and audio recordings were presented separately in an online survey to two groups of volunteer participants. The study was implemented with oTree [CSW16], an open source framework for developing behavioral experiments that allows researchers to design, conduct, and analyze experiments in a web-based environment. The design, implementation, and analysis of the user study is described in the following sections.

Multimedia and Synchronization

The videos of the aerial gestures were recorded using a Google Pixel 5 with default camera settings. Audio was captured using a Shure SM57 instrument microphone with a noise filter and a Focusrite Scarlett 2i2 studio-quality interface. Audio and video of each aerial gesture were synchronized using a signal tone played in mid-air

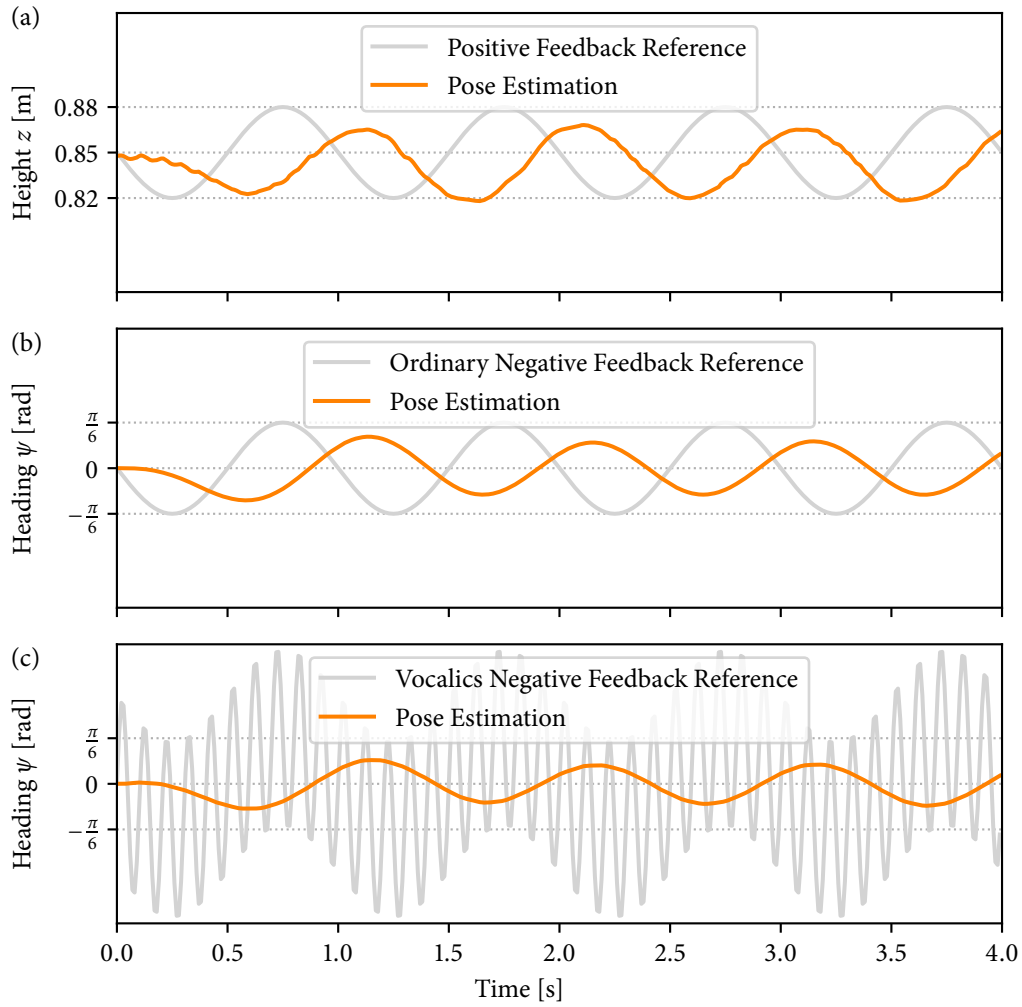


Figure 4.22: Relevant reference parameters and parameters tracked by the pose estimation system of the three trajectories: The altitude z for the positive feedback trajectory (a), and the drone's heading angle ψ for both the *ordinary* negative feedback trajectory (b) and the *vocalics* negative feedback trajectory (c). The reference parameter is shown in the background, while the tracked parameter provided by the pose estimation system is shown in orange.

just before the gesture began. For later analysis, the software controlling the drone logged key variables at 100 Hz, including control input, position, attitude, battery voltage, and control method. The control method changes when switching from hover control to trajectory control for the flight gestures. This switch is used to synchronize the log data with the audio recording and signal tone.

Hypotheses

The alternative hypothesis is that

the group whose negative feedback gestures were acoustically augmented by vocalics can better link the sounds to the videos

implying higher audible information content in the drone's *vocalics* negative feedback trajectory, as it visually closely resembles the *ordinary* negative feedback trajectory. To reject the null hypothesis that

the additional information will not increase the chance of linking the sound correctly to the gestures

the common significance level of 5 % is chosen.

Study Design

Participants were given a brief written introduction to the user study and a consent form, both of which are included in Appendix A.4.1, along with screenshots of the full survey and tables of key research data collected. It was explained that participants would first view two short videos, each featuring a drone performing a specific movement, followed by listening to two audio files. They were also informed that having a device with audio output was a prerequisite for participation, and that the media sound should be turned on. After agreeing to the consent form, age, gender, and occupation were asked before the actual experiment began.

The user study was divided into two groups: the *ordinary* trajectory group (A) and the *vocalics* trajectory group (B). Both groups would see two videos and then listen to two audio files of the previously presented videos of aerial gestures: the positive feedback trajectory and either the *ordinary* negative feedback trajectory or

Table 4.4: Media orders of subgroups where media index 1 is the positive feedback trajectory and media index 2 is the negative feedback trajectory.

Subgroup	Survey Media Order			
	1	2	3	4
	Video	Video	Audio	Audio
1	1	2	1	2
2	1	2	2	1
3	2	1	1	2
4	2	1	2	1

the *vocalics* negative feedback trajectory. The videos shown to participants in the first part of the study, from which the audio for the second part of the study was extracted, are available on YouTube⁶. To ensure impartiality, the participants in both groups were again divided into four subgroups each so that the media could be presented in the pseudorandom sequences shown in Table 4.4.

Participants could watch the videos multiple times or listen to the sounds multiple times, but they could not navigate back to the previous media item. The buttons to move to the next page were disabled until media playback was complete. The time spent on each media page was determined by taking the time the page finished loading and then calculating the time difference until the button to move to the next survey page was pressed. This ensured that the participants played the media in its entirety. Data from participants who did not reach the last page of the study has been deleted.

After each of the two audio files, participants were asked to associate the sound they had just heard with one of the two previously shown videos. They could choose between *Video 1* (1), *Video 2* (2), and *Not sure* (0). Both sounds had to be correctly assigned to the corresponding videos to be considered *correctly linked* for statistical analysis.

⁶<https://www.youtube.com/playlist?list=PLPhCA5Y9lesmAPD5TRXISybvMxNve0jA>

Table 4.5: Contingency table presenting the results of the conducted user study.

Correctly Linked	Ordinary (Group A)	Vocalics (Group B)	Combined Response
Yes	44	64	108
No	52	32	84
Total	96	96	192

Participants

The user study was conducted with $N = 192$ participants, of which 99 were male (51.6 %), 90 were female (46.9 %), and three were diverse or undisclosed (1.5 %). Participants were aged 19 to 69 years (mean $\bar{x} = 36.0$, standard deviation $\sigma = 7.8$). They were divided into 24 subgroups. All volunteered for the study and were recruited via social media.

Results

Of the 96 participants in the *ordinary* group, 44 (45.83 %) were able to correctly associate both sounds with the gestures, while for the *vocalics* group, 64 of the 96 participants (66.67 %) were able to do so. The collected data is presented in the 2×2 contingency table Table 4.5, which was analyzed using Barnard's exact test [BAR47].

Under the null hypothesis, that the additional information introduced by vocalics does not increase the probability of correctly linking the sounds to the aerial gestures, the probability of obtaining test results at least as extreme as the observed data is approximately $p = 0.0019$. As the p value is below the chosen significance level of 0.05, there is sufficient evidence to reject the null hypothesis in favor of the alternative hypothesis, meaning that participants are better able to link acoustic information to aerial gestures when these are augmented by the proposed vocalics superimposition.

The complete key research data collected, on which this evaluation is based, is presented in Appendix A.4.2.

4.4.3 Discussion

The results of the experiment suggest that the acoustic augmentation of the *vocalics* trajectory, achieved by superimposing a frequency onto the flight sound, positively impacted participants' ability to connect the flight sounds with the aerial drone gestures, making it easier for them to associate and differentiate the two, which led to a higher success rate in the *vocalics* group compared to the *ordinary* group. This underlines that the idea of extending the consequential sound of drones is capable of conveying information and thus opens up a previously unexplored channel of nonverbal communication in HDI.

Methodology

Online studies make it easier to reach a larger number of people across a broad demographic spectrum. More importantly, the effects typically associated with drone proxemics, such as increased psychological stress, result from the physical presence of the drone. An online study ensured that the results were free from these influences. Participants were able to fully focus on the visual and auditory content of the study in an environment of their own choosing.

Participants conducted the online study independently without supervision. Control over the playback of multimedia content was restricted. However, measures were programmed into the survey to ensure that participants could not return to the previous media content, and that they could not proceed to the next media item until the current item had been played to completion. Browsers, though, do not have access to system settings such as device volume. It is assumed that participants, all of whom participated voluntarily, actually turned on the sound as they were asked to do on the first slide of the survey. There was also no control for whether participants used their device speaker, headphones, or external speakers. It is presumed that participants are familiar with their own devices and consume media through them, as they were acquired through social media. If they did not change their surroundings between individual aerial gestures, there are no concerns in this regard for this study as they were meant to compare sounds with each other; it is only important that they would hear them through the same device and in the same surrounding.

According to the page times database generated by oTree, which stores a timestamp for when a survey page is completed, there were two participants (standard score > 2) who took more than the average 2.7 min for the four pages containing with media (namely 44 min and 264 min). Unfortunately, the database is missing page time entries for eight participants, so the average was calculated using only 184 participants. For the two outliers, it could be doubted that they had consumed all media in the same environment, and for eight people missing time information, it is simply unknown. However, excluding these ten participants from the statistical analysis further reduces the p value to $p = 0.0011$.

Drone Vocalics Limitations

The augmentation of drone rotor sounds as part of HRI is limited to ranges within earshot. Loud environments, where ambient noise drowns out the consequential drone sound are also unsuitable for drone vocalics as an isolated communication channel. On larger drone platforms with slower rotor speeds, it remains to be investigated how the proposed method can be implemented without affecting flight characteristics. The effect of superimposing the trajectory with oscillations on the quadrotor battery also remains uncertain and requires further evaluation.

Technical Limitations

The yaw motion of the drone induced a noticeable change in altitude in both the *ordinary* and *vocalics* negative feedback trajectories (Figures 4.21b and 4.21c). Rotation about the yaw axis is induced by a torque imbalance. In a quadrotor, opposing rotors spin in the same direction: one pair rotates clockwise and the other pair rotates counterclockwise. To initiate yaw, the speed of the rotors rotating in one direction is increased while the speed of the rotors rotating in the opposite direction is decreased. This results in a net torque, causing angular acceleration in the desired direction. The sum of forces generated by the individual rotors during yaw motion can momentarily exceed the thrust necessary to maintain the current altitude as rotors spin up faster than they spin down. In addition, the flight controller can only reduce the rotor speed to a minimum idle speed, which contributes to this effect.

There is a latency in the flown trajectories compared to their reference trajectories, as shown in Figure 4.22. Since the participants view trimmed video recordings of the flights, this has no impact on the study or its results. However, the latency should be further reduced by optimizing the smoothing of the pose estimation system and controller parameters.

To calculate the reference negative feedback trajectory, the total angle of the head shake motion was set to 60° . Because the controller used does not take into account the mathematical model parameters of the specific drone and the feasibility of trajectories, the *ordinary* negative feedback trajectory only reaches a full head shake range of 42° . The superimposed signal in the *vocalics* negative feedback trajectory further reduces this range to 32° and could be compensated by adjusting the trajectory controller.

Proposed Approach

The proposed approach modifies the trajectory rather than the controller parameters. As a result, more complex signals, such as SFUs mapped from selected nonphonemic properties of speech, can be easily superimposed on the drone's trajectory. However, with the proposed method, trajectories must be planned in advance, whereas a dedicated controller implementation could have the potential to dynamically add vocalics to a trajectory that the drone is already flying.

Humans may interpret vocalic communication as a plausible character cue. Therefore, compared to synthetic acoustic feedback from drones, such as buzzers or sounds played through loudspeakers, the proposed vocalics approach has the potential to provide more naturally appearing drone feedback. By integrating vocalics with common nonverbal communication channels like kinesics, a social drone character could be created that resembles well-known animated film characters, enabling even richer interactions.

5

DISCUSSION

This chapter consolidates research findings of this thesis and offers a comparative analysis to related work. The exploration extends to the general implications of the findings for the field of HDI. Additionally, the discussion delves into the limitations observed and contemplates potential future directions.

5.1 INTEGRATION OF FINDINGS

From preferences for trajectory types and emotional responses to tolerated distances, observed haptic engagement, and the positive influence of consequential sound on gesture recognition, each facet contributes to HDI. The research findings across kinesics, proxemics, haptics, and vocalics are summarized below.

KINESICS

- Rhythmic trajectories are preferred over pragmatic trajectories.
- Participants feel more positive (valence) and calmer (arousal) when the quadrotor anticipates its movement.
- After anticipated flights, participants feel very comfortable, in comparison to pragmatic flights.

PROXEMICS

- People allow miniature, mechanically-looking UAVs into their personal space, even close to the transition to intimate space, specifically at an average stopping distance of 0.63 m (recalling that intimate space starts at 0.45 m).
- No significant correlations were found between prior experience or ownership of a robot or drone and the chosen stop method or distance.

- Subjects feel more dominant when the drone approaches from the front.
- Their arousal increases when it approaches from the back and right.
- Participants feel less comfortable when a drone approaches from the back.
- Participants tend to misjudge the distance when the drone approaches from behind, outside their line of sight.

HAPTICS

- People tend to voluntarily engage in haptic interaction. About a quarter of the user study flights in the proxemics study were stopped by hand; almost half of the participants stopped the quadrotor at least once by hand.
- The concept of *MetroDrone* is reinforced by the preference of the participants in the user studies for rhythmic flight paths and their instinctive stopping of the drone by touch.

VOCALICS

- The augmentation of a drone's consequential sound positively influences participants' ability to distinguish aerial gestures when hearing the sound without seeing the gesture.

5.2 COMPARATIVE ANALYSIS

Principles of character animation have previously been incorporated into HDI. This thesis discussed the established animation principles and explored possibilities for their application in the context of drone communication and trajectory generation. A basic mathematical approach utilizing linear extrapolation was introduced to enrich quintic quadrotor trajectories with anticipation, and its influence on human emotional states was evaluated through a user study. To date, these aspects have not been addressed by the existing publication on this matter.

Compared to related proxemics user studies, the seated setting and flights approaching from different directions are experimental designs that have not been explored before. Tolerated distances of drones of such small size have also not

yet been evaluated. Assessments of proxemics in HDI heavily rely on achieving repeatable results through the actual flight of the drones. In many comparable studies, the quadrotors used either lacked the ability to fly autonomously or did not involve physical flight at all. Instead, they were either manually controlled, moved, or navigated along trajectories that are difficult to replicate due to unstable navigation relying on onboard sensors. Providing the study participants with methods to stop the drone's approaching using either foot or hand eliminates potential delays that can arise from having to instruct an operator to stop the quadrotor, as is common in many proxemics user studies [ABD17; Woj+19; DM13; Yeh+17]. Only a limited number of studies utilized pose estimation systems to ensure smooth and reproducible trajectories, providing more reliable findings. Research groups that conducted experiments without ensuring repeatable results identified this as a problem.

The presented method of virtually defined buttons eliminates the need for physical buttons on drones, as utilized in previous experimental studies. The differentiation through button press angles also allows for a multitude of buttons, which would be challenging to implement with physical buttons. Another advantage is that in case of a failure of physically attached buttons, virtual buttons could still be used as a fallback solution. The actuation of a virtual button is not confined to finger precision, eliminating the need for precise targeting while the drone is in motion. Since the proposed method exclusively evaluates the direction of operation, the entire palm could be utilized with an appropriately designed drone frame. The conducted inhibition threshold user study revealed that people are comfortable with smaller-sized drones approaching them closely. Some participants playfully interacting with the drone by stopping it with their hands in the kinesics user study or tapping its frame repeatedly just out of joy and curiosity could confirm that humans naturally employ touch interaction with drones [Abt+17].

The utilization and intentional modification of the consequential drone sound, extending beyond the sound itself (such as for navigation purposes), introduces a completely novel approach to nonverbal communication between drones and humans. In comparison to synthetic acoustic feedback from drones, such as buzzers, the proposed vocalics approach holds potential for more naturally appearing drone feedback, as humans may interpret vocalic communication as a plausible character cue.

In general, reproducibility of user studies in the fields of HRI and HDI is crucial for ensuring the reliability, applicability, and credibility of research. The inability

to replicate experiment settings and potential findings undermines the validation, generalization, and overall progress of scientific inquiry. Particularly within the field of HDI, achieving reproducibility in conducted experiments proves challenging, as different hardware and test conditions are frequently employed. Beyond the spatial setting, robots exhibit differences in size, appearance, operation of flight, and trajectories. Some drones are technically incapable of executing repeatable trajectories, or they are manually controlled, resulting in non-reproducible flights. In other cases, drones are suspended from ceiling rails in experiments that do not reflect realistic scenarios.

To address these challenges, the detailed description of the *ICARUS* testbed in this thesis aims to help researchers with similar endeavors. While replicating an entire infrastructure is challenging, the studies that accompany this thesis provide specific details, including trajectory parameters, trajectory generation methods, and tuning parameters for the controllers that track these trajectories. These measures facilitate precise reproducibility of trajectories and user studies across different testbeds and institutions. Explicitly providing such details enhances the potential for exact replication, a crucial aspect in fostering robustness and comparability of research results across studies.

5.3 IMPLICATIONS FOR HUMAN-DRONE INTERACTION

Throughout the endeavor to enhance HDI, minimalist strategies without the need for complex external sensors or mechanisms utilizing onboard capabilities were favored for several reasons. By foregoing even simple changes in appearance, such as equipping a drone with displays that show faces or disguising it as a pet, creative processes are initiated that can use expressive movement as a very powerful communication tool in the context of kinesics. Exploring novel interaction contexts in proxemics studies, such as seated settings and directional approaches, provides valuable insights for designing drones that prioritize human comfort and usability in diverse, future environments. The intuitive nature of touch interaction, akin to interactions with animals, is implemented through virtual buttons using the onboard IMU. This method can also be regarded as a last resort, particularly in situations where other sensors, such as cameras, may fail when the drone is in close proximity. As an acoustic communication channel, utilizing the consequential sound of the drone does not add to the already unpleasantly perceived volume. It

can also be employed when the drone is out of sight (but within earshot). Vocalics directly result from the flight path design or the used controller—i. e., kinesics are closely intertwined with vocalics. They can deliberately reinforce intentions by using vocalics as an additional nonverbal communication channel, or, if not properly designed, convey contradictory information. Similar to the haptic interaction channel, this can also be used as a fallback method. Each of the proposed and implemented nonverbal HDI channels remains effective as long as the drone is operational and able to fly.

5.4 LIMITATIONS AND FUTURE DIRECTIONS

In future endeavors, there are several aspects of the proposed interaction, that can be further explored and developed. Drone kinesics, involving communication through expressive movements, may affect battery capacity and consequently, flight duration. It necessitates visual contact to a drone and may generate sounds incongruent with the intended message. Evaluating improvement of interaction quality by adding further animation principles other than *anticipatory* motion intent, is left for future work. Future endeavors should also include exploration of methodologies to transfer whole characters from real life or animated motion pictures into drone motion.

By incorporating tap intensities, e. g., the quadrotor could respond with varying degrees of movement, ranging from broader to more reserved motions. Introducing a spherical housing would overcome the limitation of virtual buttons being restricted to the horizontal plane as all tap directions could be directed toward the center of mass of the quadrotor. This enhancement would allow for taps in arbitrary directions, expanding the range of interaction possibilities. Furthermore, a housing would provide a safety measure required by larger drones. Further improvement of the directional tap interaction could be achieved by implementing the detection of angular accelerations initiated by the user, thus enabling angular tap detection. This can also be accomplished by utilizing sensor data from the onboard IMU, expanding the range of the drone's "perceptual abilities".

Expanding the *MetroDrone* beat detection method to work with different time signatures and rhythms would contribute to a more comprehensive and engaging user experience. Additionally, the song that plays could be identified using appropriate services. With the song information, databases could be queried to determine the

genre or even a danceability score of the song. This information could then be utilized to generate genre-specific trajectories, enhancing the connection between the music and the quadrotor's movements. These potential directions for future work aim to enhance the interaction capabilities and overall performance of the system, providing participants with more nuanced control and a richer experience with the quadrotor.

Building on the vocalics approach presented, new ways to further improve HDI can be explored, e. g., by developing new interface designs that include both visual and auditory cues. Especially for individuals with visual impairments, this could make drone technology more accessible. In future scenarios, where drones and their sounds may have become more commonplace, vocalics could be an additional channel for attracting attention alongside the currently available possibilities, or to communicate with pedestrians. This study served as a proof of concept; future studies should focus on exploring a wider variety of tones, communicating more meaningful messages and emotions, and qualitatively assessing them. The proposed method can be expanded by superimposing different frequencies, encoding sequences of varying signal durations, such as Morse code, or mimicking bio-inspired sounds like the pulsed vibration used by honey bees for communication.

CONCLUSION

To enable research in HDI, the drone testbed *ICARUS* was created as part of this thesis, exclusively utilizing commercially available hardware, making it highly cost-effective. At its smallest scale, it can be run with just a drone, a camera, and a laptop, making it suitable for applications such as lectures or demonstrations. The system is highly versatile due to the implementation of several common hobbyist radio protocols. The use of miniature drones eliminates the need for elaborate safety measures. With the integration of NUI sensors, it is particularly well-suited for conducting HDI experiments, which constitute the core of this thesis, involving investigations and experimentation with new interaction possibilities.

In the area of drone *kinesics*, the well-known character animation principles of the Disney artists were discussed and examined for possible transfer to drone trajectories. While design principles from animators have been used to inspire robot motion in the past, this is yet an area of little research for drones. The most promising of these principles in terms of reducing the inhibition threshold, anticipation, was selected and implemented through a straightforward approach, namely by extrapolating quintic polynomial trajectories. An experiment was conducted, involving participants' preferences for different trajectories (pragmatic, anticipated, and rhythmic) and their effects on the participants' emotional states. The rhythmic trajectory was preferred over the pragmatic trajectory, and the anticipated trajectory proved less intimidating to participants than sudden movement.

Until recently, *proxemics* studies were mostly performed with large drones or were often not repeatable due to lack of accurate pose estimation. A corresponding experiment was conducted and its findings discussed. It evaluated tolerated distances in tactile HDI and the participants' inclination to physically interact with a quadrotor. The study was carried out with participants seated and from four directions, employing a new experimental design. The trajectories were fixed at a height suitable for comfortable touch interaction. The participants had the choice to stop the quadrotor by a tap to its frame or using a foot switch. The aim of this study was to simulate scenarios that may occur in future home or workplace envi-

ronments where social drones are used to provide services to humans. The study showed that humans allow miniature UAVs, that were not modified by adding social features, into their closer personal space without an increase in mental stress when approaching from the front. The group that chose to stop the quadrotor by tapping its frame allowed it to enter their intimate space. The fact that nearly half of the participants were unafraid to use their hands for interaction could not be correlated with past experiences with drones or ownership of robots. The group of participants who enthusiastically tapped the drone's frame served as proof that an efficient touch is a relevant interaction method with drones. This also emphasizes the importance of incorporating playful elements into HDI, such as rhythmic trajectories or repeated tapping on the frame.

Virtually defined buttons have been introduced for drone *haptics*. This represents a novel approach to HDI, leveraging the basic human instinct of touch. Utilizing the accelerometer data from the quadrotor's built-in IMU, this minimalist approach eliminates the need for extra intermediary devices. It allows for the detection of user taps based on peaks in the accelerometer signal. As proof of concept, an example of double-tap functionality was introduced. In an additional scenario employing multi-tap interaction, a quadrotor responds to repetitive user taps to its frame by initiating various rhythmic trajectories, depending on the operated button. This approach aimed to create a playful bond between humans and robots by sharing a musical experience, thereby reducing the inhibition threshold to HDI. In certain interaction scenarios, physical communication is preferred, and touch interaction may even be necessary in close proximity, especially when conventional user interaction sensors are ineffective, e. g., when cameras are too close to capture human gestures. Touch interaction, for example, can serve as a safety fallback measure to stop a drone from advancing, as demonstrated in the conducted proxemics user study. Although rotor guards are commonly used to ensure user safety, adding such features was intentionally avoided to emphasize the minimalist approach in the *MetroDrone* proof of concept. For studies involving external participants, frames with predefined points of interaction were used to ensure their safety. Moreover, any additional weight significantly reduces the flight time of a small-sized quadrotor. Since the minimalist nature of the proposed approach incorporates the drone's IMU instead of physical buttons or capacitive sensors, this method can be implemented on nearly any quadrotor system.

The augmentation of the consequential sound produced by a drone was investigated, resulting in *vocalics* trajectories, i. e., aerial gestures with additional acoustic

informational value. A user study was conducted to evaluate its potential. Visually clearly differentiable positive and negative feedback trajectories mimicking human gestures were generated and tracked by a quadrotor. However, relying solely on auditory information, such as in situations where it is not possible to maintain visual contact with the drone, or for people with visual impairments, makes it difficult to acoustically distinguish between these two gestures. Superimposing a higher-frequency signal to the negative feedback trajectory did not affect the visual impression of the flight, but added enough acoustic information to be better distinguishable from the positive feedback trajectory. Next to the results and methodology of the user study, the limitations of the approach and the technical implementation were discussed. The proposed vocalics approach represents an initial step toward integrating vocalics into HDI by adding auditory cues. As with the other proposed approaches, drone vocalics were motivated by capturing existing resources that are already part of the drone.

HRI endeavors to incorporate as many nonverbal channels as possible to enhance the interaction experience and ensure satisfaction for all participants. This thesis aims to equip the HDI community with a new set of tools for employing effective communication channels, thereby making future interactions even more fruitful. Minimalist concepts for the nonverbal communication channels of kinesics, proxemics, haptics, and vocalics were developed, implemented, and evaluated. Their implementation did not rely on additional sensory equipment, emphasizing a minimalist design that is also efficient and economical, and can be used as a fallback method when dedicated sensors fail. They provide a robust foundation for the ongoing development and improvement of interaction experiences in the emerging field of HDI, and offer the potential to elevate a social robot like Vector to an authentic implementation of a flying social companion.

This thesis aims to inspire researchers, character designers, and interaction designers within the HRI and HDI communities. It encourages them to use and exploit available tools, using the proposed approaches as a basis to integrate various pre-existing nonverbal communication channels of drones that have been further shaped or newly created in this thesis. The objective is to elevate the interaction, striving for a rewarding and rich experience that ultimately results in a satisfying nonverbal human-drone interaction.





USER STUDIES

This appendix includes an ethics statement, providing detailed information on the ethical considerations related to the conducted user studies. It also contains the survey pages and questionnaires used during the studies. Additionally, the collected key research data, which is crucial for evaluating the study's findings, is presented.

A.1 ETHICS STATEMENT

During the time of research, RheinMain University of Applied Sciences did not have a formal ethical review board in place. However, it was assured that established ethical principles were diligently adhered to, which can be summarized as follows:

- Participants had to be older than 18 years, which is the legal age of majority in Germany.
- The responses were subjected to thorough anonymization to prevent any linkage to specific individuals. During online surveys, neither IP addresses, names, locations, nor any other identifying data were recorded. Because participants in the physical survey were compensated for their participation, information about their payment was collected separately and could not be correlated with their questionnaires.
- Participants were not allowed to participate more than once. For the online survey, this was ensured by the survey tool.
- Participation was entirely voluntary, and participants could withdraw at any point during the survey.
- Data to be collected was disclosed to the participants before the studies and is summarized in Appendix A.2.

- Because the survey was conducted at a German university, the collected anonymized data was transferred to the university's server in Wiesbaden, Germany.
- Participants were informed that the results of this study might be presented at scientific conferences, published in scientific journals, or shared with the scientific community in an anonymized form.

The study procedure and the above information were provided to participants in advance to which the voluntary and informed consent of participants was obtained.

The following additional established ethical principles were also diligently adhered to:

- There were no associated physical, psychological, social, or economic risks for participants.
- The drone studies held no potential for physical harm. Appropriate safety measures were taken for the physical studies, including a frame to protect humans from contact with the drone's rotors, continuous monitoring of the head-drone distance, and a physical switch that immediately shuts down the drone. The online study was conducted remotely through videos and sound recordings of drone flights on devices owned by the participants, such as mobile phones or computers.
- The studies were conducted without any involvement in deceptive practices.
- The studies carry no potential for dual use or exploitation by criminal or terrorist groups.
- The samples did not target any vulnerable populations, such as ethnic minorities, patients undergoing treatment, or minors.
- There exist no conflicts of interest among the authors that could compromise the research's integrity.

A cookie consent form was not shown in the online study because, according to the EU General Data Protection Regulation (GDPR), the use of first-party cookies only requires consent if they are not considered strictly necessary for the functioning of the website.

A.2 DATA SUMMARY

The key research data of experimental and statistical nature was collected through online surveys, questionnaires, and numerical data logged by the drone control software. This data led to the scientific findings presented in this dissertation and consists of the following:

- For all experiments
 - Unique identifier
 - Age or year of birth
 - Gender identity
 - Occupation
- Proxemics and trajectory experiments
 - Skepticism toward drones
 - Ownership of service robots
 - Contact with drones
 - Ownership of drones
 - Initial SAM test
 - 15 questions to determine personality traits
- After each proxemics flight
 - SAM test
 - How comfortable they felt
 - How they would judge the final distance to the quadrotor
- After pragmatic vs. anticipated flight
 - SAM test
 - How comfortable they felt
 - Whether they could anticipate the quadrotor's intent
- After pragmatic vs. expressive flight
 - SAM test
 - Whether the flight was emotional, mechanical, human, or sympathetic

- Which of the flights they liked better
- Vocalics experiment
 - Browser name and version
 - Unix time of access
 - Experiment group
 - Media presentation order
 - Assignment of audio to video files
 - Time spent on individual media pages
 - Unix time of individual survey page access
 - Index of the last survey page visited

Data collected beyond this list can be found in the questionnaire, but was not used for further statistical analysis.

Gender identity is reported as female (f), male (m), diverse (d), or undisclosed (u). Answers to Likert scale questions are given as values from 1 to 5, answers to binary questions are reported as 0 or 1.

A.3 KINESICS AND PROXEMICS

This section contains the questionnaire from the kinesics and proxemics user studies described in Sections 4.1 and 4.2, respectively, which were conducted in one session. The questionnaire and provided data include all three experiments, namely Experiment 1 (Versuch 1) for proxemics and stop method evaluation, Experiment 2 (Versuch 2) for pragmatic vs. anticipated trajectory evaluation, and Experiment 3 (Versuch 3) for pragmatic vs. expressive trajectory evaluation.

A.3.1 *Questionnaire*

This section contains the questionnaire as presented to the participants.

Miniatur-Quadcopter Experiment

Instruktionen

Herzlich Willkommen zu diesem Experiment und vielen Dank für Ihre Teilnahme! Bitte lesen Sie diese Anweisungen, die für alle gleich sind, aufmerksam durch. Dieses Experiment wird bis zu 30 Minuten dauern und Sie erhalten als Entlohnung für Ihre Zeit 10 Euro am Ende des Experiments ausgezahlt, vorzugsweise digital per Paypal oder per Banküberweisung. Bitte schalten Sie Ihr Mobiltelefon auf lautlos. Sollten Sie nach dem Durchlesen der Instruktionen Fragen haben, bitten wir Sie, diese gleich vor Beginn des Experiments an den Experimentator zu richten. Dieser begleitet Sie während des gesamten Ablaufs durch das Experiment und steht auch währenddessen für aufkommende Fragen zur Verfügung. Die Auswertung des Experiments erfolgt anonym und fließt in eine zur Veröffentlichung geplante Statistik ein. Es werden keine Foto-, Film- oder Tonaufnahmen angefertigt. Bitte lesen Sie zunächst den folgenden Versuchsablauf aufmerksam durch und füllen Sie dann die beiliegende COVID-19-Erklärung aus und unterschreiben Sie diese. Auf dem Formular muss auch die gewünschte Auszahlungsart mit Paypal-Account oder IBAN angegeben werden.

Ablauf

Ihre Aufgabe im Rahmen des Experiments besteht darin, mit einem Miniatur-Quadcopter, dargestellt in Abb. 1, zu interagieren. Es handelt sich dabei um einen fliegenden Roboter, der in der Diagonale 92 mm misst und 36,7 g wiegt.

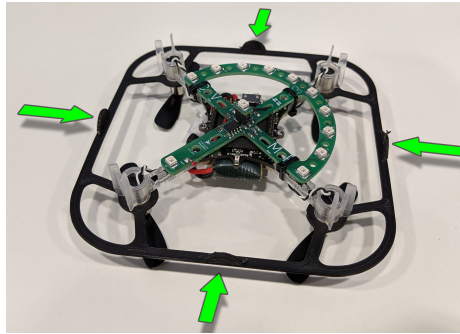


Abb. 1: Verwendeter Miniatur-Quadcopter und möglichen Stellen, an denen dieser in optional angetippt werden kann.

Insgesamt werden Sie an drei Versuchen teilnehmen. Bei jedem Versuch sitzen sie am Ende des Flugbereichs auf einem Hocker. Jeder Versuchsdurchgang wird mit einem kurzen Fragebogen abgeschlossen. Im folgenden sind die drei Versuche kurz beschrieben.

Versuch 1

Der Miniatur-Quadcopter wird mit 3 m Abstand von Ihnen in die Luft steigen und dort auf einer Höhe von 1,1 m schweben. Sobald Sie bereit sind, startet der Experimentator den Flug; der Miniatur-Quadcopter wird sich dann mit moderater Geschwindigkeit (0,5 m/s) auf Sie zu bewegen. Die Höhe bleibt dabei konstant. Sie können den Quadcopter zu einem beliebigen Zeitpunkt per Betätigung des Fußschalters oder durch Antippen an einem der in Abb. 1 dargestellten Seiten zum Stillstand bringen.

Dieser Versuch wird vier mal durchgeführt, dabei schauen Sie jeweils in eine andere Richtung des Raums, in dem Sie sich befinden. Auf jeden Versuch erfolgt das Ausfüllen eines kurzen Fragebogens.

Versuch 2

Der Quadcopter wird sich zwei mal auf unterschiedliche Art, aber immer mit moderater Geschwindigkeit (0,5 m/s) auf einer konstanten Höhe von 1,1 m auf Sie zu bewegen und von selbst zum Stillstand kommen. Bitte beobachten Sie jeden Flug aufmerksam. Im Anschluss an jeden Flug wird ein kurzer Fragebogen ausgefüllt.

Versuch 3

Der Miniatur-Quadcopter wird wieder zwei mal mit moderater Geschwindigkeit (0,5 m/s) und einer konstanten Höhe von 1,1 m auf Sie zu fliegen und selbstständig vor Ihnen zum Stehen kommen. Bitte beobachten Sie jeden Flug aufmerksam. Im Anschluss an jeden Flug wird ein kurzer Fragebogen ausgefüllt.

Fragebogen (allgemein)

Über Ihre Person

1. Ihr Geschlecht:	<input type="checkbox"/> männlich	<input type="checkbox"/> weiblich	<input type="checkbox"/> divers
2. Ihr Geburtsjahr:	_____		
3. Ihr Studiengang/Beruf:	_____		

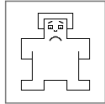
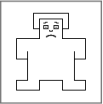
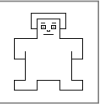
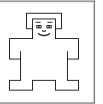
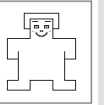
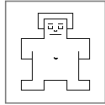
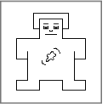
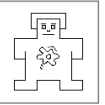
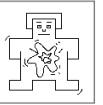
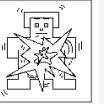
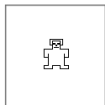
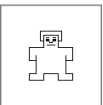
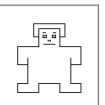
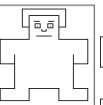
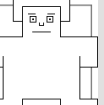
Ich bin jemand, der...

4. gründlich arbeitet	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
5. kommunikativ, geschwätzig ist	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
6. manchmal etwas grob zu anderen ist	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
7. originell ist, neue Ideen einbringt	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
8. sich oft Sorgen macht	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
9. verzeihen kann	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
10. eher faul ist	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
11. aus sich herausgehen kann, gesellig ist	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
12. künstlerische Erfahrungen schätzt	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
13. leicht nervös wird	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
14. Aufgaben wirksam und effizient erledigt	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
15. zurückhaltend ist	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
16. rücksichtsvoll und freundlich mit anderen umgeht	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
17. eine lebhaft Phantasie, Vorstellungen hat	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
18. entspannt ist, mit Stress gut umgehen kann	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu

Ihre Erfahrung mit Technologien und Robotern

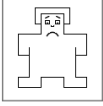
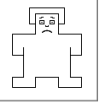
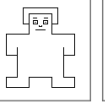
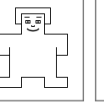
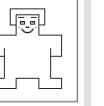
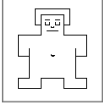
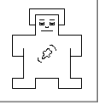
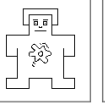
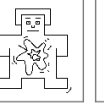
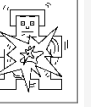

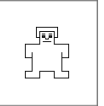
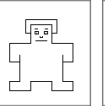
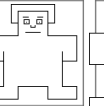
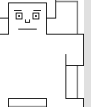
19. Ich bin an neuen Technologien interessiert	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
20. Ich verwende innovative technische Geräte	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
21. Ich kann intuitiv mit Technik umgehen	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
22. Technischem Fortschritt stehe ich kritisch gegenüber	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
23. Drohnen stehe ich kritisch gegenüber	trifft überhaupt nicht zu	<input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/>	trifft voll zu
24. Besitzen Sie einen Service-Roboter?	<input type="checkbox"/> ja, Typ: _____	<input type="checkbox"/> nein	
25. Kamen Sie in der Vergangenheit bereits mit Drohnen in Kontakt?	<input type="checkbox"/> ja	<input type="checkbox"/> nein	
26. Besitzen Sie selbst eine Drohne?	<input type="checkbox"/> ja	<input type="checkbox"/> nein	

Wie fühlen Sie sich?

27. Freude					
28. Erregung					
29. Dominanz					

Fragebogen Versuch 1 (1)

Wie fühlen Sie sich?

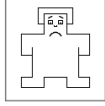
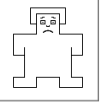
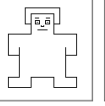
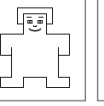
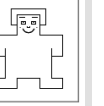
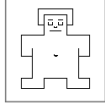
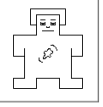
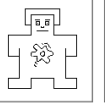
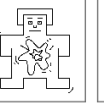
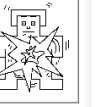

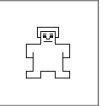
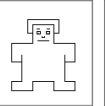
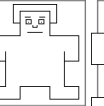
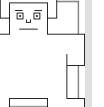
1. Freude					
2. Erregung					
3. Dominanz					

Ergänzend:

4. Wie wohl fühlen Sie sich?	überhaupt nicht wohl <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> sehr wohl
5. Wie beurteilen Sie die Distanz zu dem Quadcopter als er zum Stillstand kam?	zu nah <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> zu weit weg

Fragebogen Versuch 1 (2)

Wie fühlen Sie sich?

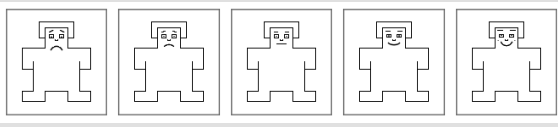
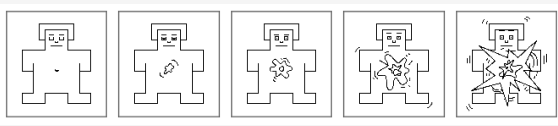
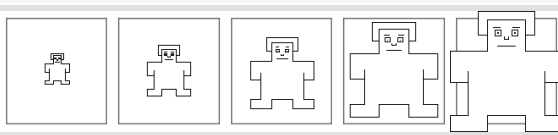
1. Freude					
2. Erregung					
3. Dominanz					

Ergänzend:

4. Wie wohl fühlen Sie sich?	überhaupt nicht wohl	□—□—□—□—□	sehr wohl
5. Wie beurteilen Sie die Distanz zu dem Quadcopter als er zum Stillstand kam?	zu nah	□—□—□—□—□	zu weit weg

Fragebogen Versuch 1 (3)

Wie fühlen Sie sich?

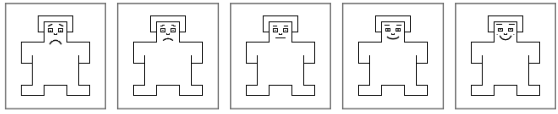
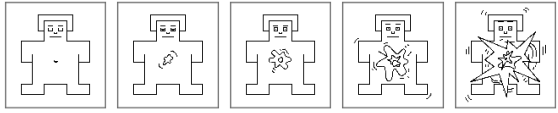
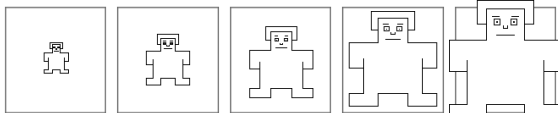
1. Freude	
2. Erregung	
3. Dominanz	

Ergänzend:

4. Wie wohl fühlen Sie sich?	überhaupt nicht wohl <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> sehr wohl
5. Wie beurteilen Sie die Distanz zu dem Quadcopter als er zum Stillstand kam?	zu nah <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> zu weit weg

Fragebogen Versuch 1 (4)

Wie fühlen Sie sich?

1. Freude	
2. Erregung	
3. Dominanz	

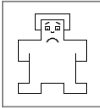
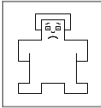
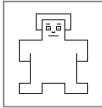
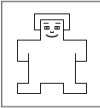
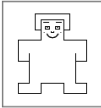
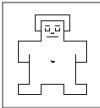
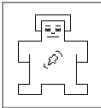
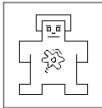
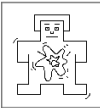

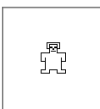
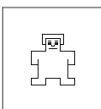
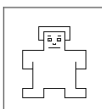
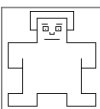
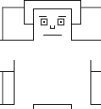
Ergänzend:

4. Wie wohl fühlen Sie sich?	überhaupt nicht wohl <input type="checkbox"/> — <input type="checkbox"/> — <input type="checkbox"/> — <input type="checkbox"/> — <input type="checkbox"/> sehr wohl
5. Wie beurteilen Sie die Distanz zu dem Quadcopter als er zum Stillstand kam?	zu nah <input type="checkbox"/> — <input type="checkbox"/> — <input type="checkbox"/> — <input type="checkbox"/> — <input type="checkbox"/> zu weit weg

Fragebogen Versuch 2

Flug 1

Wie fühlen Sie sich?

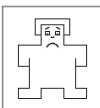
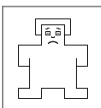
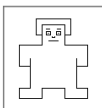
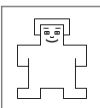
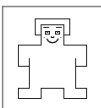
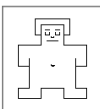
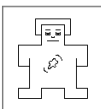
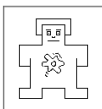
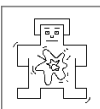

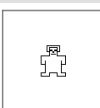
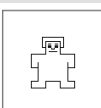
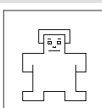
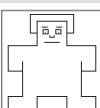
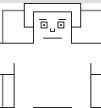
1. Freude					
2. Erregung					
3. Dominanz					

Ergänzend:

4. Wie wohl fühlen Sie sich?	überhaupt nicht wohl <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> sehr wohl
5. Ich konnte das Vorhaben des Quadcopters erkennen	trifft überhaupt nicht zu <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> trifft voll zu

Flug 2

Wie fühlen Sie sich?

1. Freude					
2. Erregung					
3. Dominanz					

Ergänzend:

4. Wie wohl fühlen Sie sich?	überhaupt nicht wohl <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> sehr wohl
5. Ich konnte das Vorhaben des Quadcopters erkennen	trifft überhaupt nicht zu <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> - <input type="checkbox"/> trifft voll zu

Fragebogen Versuch 3

Flug 1

Wie fühlen Sie sich?

1. Freude					
2. Erregung					
3. Dominanz					

Der Flug wirkte...

4. emotional	trifft überhaupt nicht zu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	trifft voll zu
5. mechanisch	trifft überhaupt nicht zu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	trifft voll zu
6. menschlich	trifft überhaupt nicht zu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	trifft voll zu
7. sympathisch	trifft überhaupt nicht zu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	trifft voll zu

Flug 2

Wie fühlen Sie sich?

1. Freude					
2. Erregung					
3. Dominanz					

Der Flug wirkte...

4. emotional	trifft überhaupt nicht zu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	trifft voll zu
5. mechanisch	trifft überhaupt nicht zu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	trifft voll zu
6. menschlich	trifft überhaupt nicht zu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	trifft voll zu
7. sympathisch	trifft überhaupt nicht zu	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	trifft voll zu

Abschließend

1. Welcher Flug hat Ihnen besser gefallen?	<input type="checkbox"/> Flug 1	<input type="checkbox"/> Flug 2
--	---------------------------------	---------------------------------

A.3.2 *Research Data*

This section contains the key research data from the kinesics and proxemics user studies described in Sections 4.1 and 4.2, respectively. The presented data contains answers from the questionnaires and numerical data recorded during the experiments by the drone control software.

Tables A.1 and A.2 contain the participant's demographic data and personality traits answers collected for both the kinesics and proxemics experiments. In Tables A.3 and A.4 the data collected to compare pragmatic with anticipated and expressive flights in the kinesics experiment are presented. Table A.5 contains the self-assessed affect and feelings in response to the drone flights of the proxemics experiment. In Table A.6 the stop distances and stop methods used are listed.

Table A.1: Key research data collected from demographic and drone-related questions.

User	Question					
	1.	Age	3.	23.	24.	25. 26.
1	m	21	media management	2		1 0
2	f	22	media conception and production	1		0 0
3	m	23	mobility management	2		0 0
4	m	24	media computer science	1		1 0
5	m	25	media computer science	3		1 1
6	f	22	media computer science	3		0 0
7	m	22	media computer science	1	vacuum cleaner	1 0
8	f	23	media management	3		0 0
9	m	21	computer science technical systems	1		0 0
10	f	26	media computer science	1	vacuum cleaner	0 0
11	m	28	applied computer science	3		1 0
12	f	24	media management	1		0 0
13	f	25	media computer science	3		0 0
14	m	35	media computer science	3		0 0
15	f	34	media computer science	1	anki cozmo	0 0
16	m	27	media management	2	vacuum cleaner	1 0
17	m	21	business informatics	2		0 0
18	f	22	media conception and production	2		1 0
19	m	19	media computer science	1		0 0
20	f	21	media conception and production	3		0 0
21	m	24	media computer science	2		1 0
22	f	25	business informatics	3		0 0
23	m	25	applied computer science	2		0 0
24	m	21	business informatics	2	vacuum cleaner	1 0
25	m	20	business informatics (dual)	2		1 0
26	m	27	computer science	1	vacuum cleaner	1 0
27	m	30	computer science	2		1 0
28	m	38	research assistant	2		1 0
29	m	29	research assistant	1		1 0
30	m	30	research assistant	4		0 0
31	f	42	computer science	3	vacuum cleaner	1 0
32	m	33	research assistant	2		1 1

Table A.2: Key research data collected from personality trait questions.

User	Question																
	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.		
1	4	4	3	3	4	5	2	5	4	2	4	2	4	5	3		
2	4	4	4	4	4	4	3	4	4	4	4	3	4	2	3		
3	4	3	4	4	4	4	4	3	3	2	3	3	4	3	4		
4	3	5	1	4	1	5	3	5	4	3	3	2	5	4	5		
5	2	1	3	3	2	4	3	1	2	3	3	4	2	2	2		
6	4	4	2	4	2	5	3	4	5	2	4	2	5	4	3		
7	4	5	3	4	5	5	3	4	5	3	5	2	5	5	3		
8	5	5	2	4	4	4	3	4	4	3	5	2	5	4	3		
9	2	3	4	4	2	4	5	3	4	2	2	3	3	4	4		
10	5	5	1	5	3	3	4	5	5	2	5	3	4	5	4		
11	4	3	2	5	5	5	2	3	5	2	4	4	5	4	4		
12	5	4	2	5	5	3	3	3	5	4	5	3	5	5	3		
13	5	2	2	3	5	2	2	2	5	5	3	5	4	5	2		
14	4	4	3	4	2	3	2	4	4	3	4	3	4	5	3		
15	4	4	2	4	5	5	2	3	4	5	3	3	5	5	2		
16	3	3	4	4	2	5	4	3	4	4	3	3	4	4	3		
17	4	4	3	4	3	4	1	2	2	3	4	3	4	4	5		
18	4	5	2	5	4	4	3	5	5	4	4	2	5	5	3		
19	4	4	2	3	4	5	3	3	5	4	3	4	5	5	2		
20	5	4	2	5	5	4	3	5	5	2	3	2	5	5	2		
21	3	3	4	4	3	5	3	3	5	2	3	4	4	5	5		
22	4	3	2	3	5	5	4	2	3	4	4	4	5	4	2		
23	4	3	2	1	3	3	4	4	3	5	3	3	3	4	2		
24	4	5	2	4	3	4	3	5	5	3	4	2	5	4	4		
25	3	4	3	4	4	4	3	4	5	3	3	4	5	4	2		
26	4	5	2	4	3	5	2	5	2	2	4	2	4	2	2		
27	4	5	2	4	3	4	3	5	5	2	4	2	4	5	4		
28	4	3	4	3	4	4	3	4	5	3	3	3	4	4	2		
29	5	3	2	4	3	4	4	3	2	2	3	4	4	3	3		
30	4	4	3	3	4	4	2	4	2	2	4	3	4	2	2		
31	4	3	2	3	3	4	2	4	4	3	4	4	4	3	3		
32	4	2	1	3	4	3	4	2	4	4	3	4	5	3	4		

Table A.3: Key research data collected from Experiment 2 (pragmatic vs. anticipated flight).

User	Question									
	Flight 1					Flight 2				
	1.	2.	3.	4.	5.	1.	2.	3.	4.	5.
1	3	3	4	5	5	3	4	3	5	1
2	4	3	1	5	2	4	3	1	4	4
3	4	1	3	5	5	4	1	3	4	3
4	3	3	3	5	4	3	3	3	5	2
5	3	2	3	5	5	3	2	2	4	5
6	5	1	1	5	4	5	1	2	5	3
7	4	3	2	3	3	2	4	1	2	1
8	3	2	2	5	4	3	2	2	5	2
9	5	2	5	5	5	5	2	5	5	5
10	5	4	4	4	5	3	4	3	3	4
11	5	2	3	5	5	5	3	3	5	5
12	5	2	2	5	5	4	4	1	3	2
13	2	3	2	2	2	2	3	2	2	1
14	4	4	2	3	2	4	4	2	3	2
15	5	4	3	5	1	5	4	3	5	3
16	4	2	4	5	4	4	3	4	4	4
17	3	1	4	4	4	3	2	3	2	3
18	3	3	3	4	3	3	4	2	3	4
19	3	3	2	4	4	3	3	2	4	5
20	4	4	2	2	2	3	2	3	1	4
21	5	1	5	5	5	5	2	4	5	4
22	4	4	2	4	5	4	4	2	4	4
23	3	2	3	3	4	2	3	1	2	3
24	4	3	3	4	5	4	3	3	3	3
25	4	4	2	5	5	4	4	3	4	4
26	5	3	3	5	5	5	3	3	5	5
27	5	1	3	5	4	5	1	3	5	5
28	3	1	2	3	3	3	1	2	3	3
29	3	4	3	4	3	3	4	3	3	2
30	4	3	3	4	4	4	3	3	4	3
31	3	2	3	4	4	3	2	3	4	5
32	4	2	3	5	1	4	2	3	5	3

Table A.4: Key research data collected from Experiment 3 (pragmatic vs. expressive flight).

User	Question														1.	
	Flight 1							Flight 2								Concluding
	1.	2.	3.	4.	5.	6.	7.	1.	2.	3.	4.	5.	6.	7.		1.
1	3	3	5	3	1	4	4	5	3	4	5	1	5	5	2	
2	5	1	1	2	3	4	4	5	1	1	3	2	5	5	2	
3	4	1	3	3	4	2	4	5	1	3	3	3	4	4	2	
4	4	4	3	4	2	2	4	5	4	3	5	1	4	5	2	
5	3	2	3	2	2	2	2	5	3	3	4	1	2	4	2	
6	5	1	2	2	4	3	3	5	3	2	3	2	5	3	1	
7	4	2	3	4	4	2	5	2	4	1	4	2	4	1	1	
8	3	2	2	1	5	2	3	3	2	2	3	2	4	2	1	
9	5	3	5	4	4	1	5	5	2	5	5	2	4	5	2	
10	5	4	4	4	5	2	2	4	4	3	5	3	3	3	1	
11	5	3	3	4	5	3	3	5	3	3	4	4	4	4	2	
12	5	4	2	4	4	3	5	5	5	2	5	2	5	5	2	
13	2	3	2	5	1	4	5	2	3	2	5	3	3	2	1	
14	4	4	2	4	4	2	2	4	4	2	4	3	2	2	2	
15	5	4	3	1	4	1	3	5	4	3	2	2	4	5	2	
16	4	3	3	3	2	1	2	3	3	3	3	1	2	2	1	
17	4	2	4	4	2	3	4	3	2	3	2	5	4	2	1	
18	4	2	4	4	2	4	4	4	2	3	4	1	5	5	2	
19	4	2	3	2	3	2	5	4	2	3	4	2	3	2	1	
20	4	2	3	2	5	4	2	4	3	4	4	2	5	4	2	
21	5	3	4	2	4	2	2	5	3	4	3	3	3	4	2	
22	5	4	2	3	4	2	5	5	4	2	4	3	2	5	2	
23	3	2	2	3	5	2	4	4	2	4	4	4	3	4	2	
24	5	3	3	4	2	5	4	5	4	3	3	4	2	2	1	
25	4	4	3	3	4	1	4	4	4	3	4	3	3	2	2	
26	5	3	3	1	5	3	3	5	3	3	4	4	4	3	1	
27	5	1	3	3	4	2	4	5	1	4	4	1	4	5	2	
28	4	1	2	2	4	2	2	4	2	1	4	2	4	4	2	
29	4	3	3	2	4	1	1	5	4	3	4	3	2	4	2	
30	4	3	3	2	4	2	3	4	3	3	4	2	3	5	2	
31	3	2	3	1	5	1	1	3	2	3	4	3	2	4	2	
32	4	2	3	1	5	1	1	4	2	3	3	3	2	4	2	

Table A.5: Research data assessing the participants' emotional states initially and after each flight in Experiment 1 (Versuch 1). The approach directions of the drones were randomized; here, the provided data is already assigned to the flight direction.

User	Question																						
	Initial			Front					Back					Left					Right				
	1.	2.	3.	1.	2.	3.	4.	5.	1.	2.	3.	4.	5.	1.	2.	3.	4.	5.	1.	2.	3.	4.	5.
1	5	3	3	4	4	5	5	4	3	3	2	3	3	3	4	2	3	2	4	4	4	4	3
2	4	3	2	4	2	2	5	3	2	4	2	3	2	4	2	1	5	5	4	1	2	5	3
3	4	2	2	4	1	3	4	3	4	2	2	4	2	4	2	2	3	2	4	1	2	3	3
4	4	3	2	5	5	3	5	3	5	5	3	4	1	4	4	3	4	4	4	3	3	5	3
5	4	3	3	5	4	4	5	3	4	4	3	4	4	5	4	3	4	2	5	4	3	4	3
6	5	3	2	5	5	5	3	2	5	4	3	3	2	5	2	3	4	4	5	2	2	5	2
7	4	3	2	4	5	2	3	3	2	5	1	1	2	3	5	1	2	1	3	4	1	3	2
8	3	1	3	3	2	3	4	3	3	3	3	3	2	3	3	3	4	3	4	3	3	4	3
9	4	2	3	4	2	3	5	3	4	2	5	5	3	4	2	3	5	3	5	2	5	5	3
10	4	3	4	5	4	4	4	3	4	4	2	2	2	4	3	3	3	3	5	4	3	4	3
11	4	2	3	5	2	3	5	4	5	3	3	5	3	5	2	3	5	1	5	2	3	5	3
12	4	2	2	5	3	2	5	3	5	4	1	4	1	5	3	2	5	2	5	3	2	5	3
13	1	3	1	2	3	2	2	2	2	3	2	2	1	2	3	2	3	3	2	3	2	3	3
14	4	3	2	4	3	2	4	2	4	3	2	4	3	4	3	2	4	2	4	3	2	4	3
15	5	3	1	5	4	3	5	3	5	4	3	5	1	5	4	3	5	3	5	4	3	5	2
16	3	2	3	4	3	3	4	2	4	3	3	4	3	4	3	3	4	5	4	3	3	4	3
17	4	2	4	3	2	3	3	2	3	1	4	4	3	3	1	3	2	2	3	2	3	2	1
18	4	2	3	4	3	3	4	3	3	5	1	1	2	4	3	2	3	4	4	4	2	2	3
19	4	4	1	4	2	3	5	3	3	4	2	3	5	3	3	2	4	4	3	3	2	4	3
20	3	4	4	4	3	4	3	3	2	2	3	2	2	4	3	3	4	3	3	4	3	3	4
21	5	1	3	5	2	4	5	3	5	1	4	5	2	5	1	5	5	1	5	2	5	5	3
22	3	4	2	4	4	3	4	3	4	4	2	4	2	4	4	2	4	3	4	4	2	4	2
23	4	3	2	4	3	2	2	3	1	4	1	3	1	2	4	3	3	3	2	4	3	3	4
24	4	3	3	4	3	3	4	3	4	3	3	4	3	4	3	3	4	3	4	3	3	4	3
25	4	3	2	4	3	3	4	2	5	4	3	5	2	4	3	3	4	3	4	3	3	4	4
26	5	2	3	5	3	3	5	3	5	3	3	5	3	5	3	3	5	3	5	3	3	5	3
27	5	1	3	5	1	4	5	3	4	3	2	5	1	5	2	3	5	2	5	2	3	5	2
28	3	2	3	3	2	3	4	3	3	2	4	4	3	4	2	3	4	3	4	2	3	3	3
29	4	3	3	3	3	3	5	3	4	4	3	4	2	4	3	3	5	3	4	4	3	4	3
30	4	3	2	4	3	4	4	3	4	3	3	3	4	4	3	3	4	3	4	3	3	4	3
31	3	3	3	3	2	3	4	3	3	4	3	4	2	3	3	3	4	3	3	3	3	4	3
32	4	2	3	4	2	3	5	3	4	2	3	5	3	4	2	3	5	3	4	2	3	5	3

Table A.6: Data for numerical evaluation of tolerated distances and approach stop methods. Abbreviations for randomized direction order use initial letters. Stop methods with an asterisk indicate head safety measure intervention. Frame taps are to the front of the quadrotor unless noted otherwise. Distances are in meters.

User	Order	Direction							
		Front		Back		Left		Right	
		Method	Distance	Method	Distance	Method	Distance	Method	Distance
1	blrf	foot	0.726	foot	0.473	foot	0.398	foot	0.755
2	bfrl	foot	1.844	foot	0.465	foot	1.945	foot	1.317
3	blfr	foot	0.419	foot*	0.317	foot*	0.326	foot*	0.332
4	rflb	tap	0.509	foot	0.378	foot	0.718	foot	0.514
5	lrfb	tap	0.539	foot	0.629	foot	0.449	foot	0.493
6	lbfr	foot	0.833	foot	0.475	foot	0.867	foot	0.381
7	fblr	foot	0.766	foot	0.436	foot	0.558	foot	1.005
8	frbl	foot*	0.328	foot	0.348	foot	0.552	foot	0.379
9	flbr	foot	0.433	foot	0.358	tap left	0.425	tap	0.469
10	rblf	tap	0.523	foot	0.445	foot	0.930	foot	0.673
11	lrfb	foot	0.773	foot	1.572	foot*	0.283	foot	0.541
12	frlb	foot*	0.325	foot*	0.327	foot*	0.355	foot*	0.314
13	rflb	head*	0.341	foot	0.461	tap	0.352	tap	0.459
14	rlfb	tap	0.383	foot	0.446	foot	0.416	foot	0.538
15	rflb	foot	0.585	foot*	0.325	foot	0.516	tap	0.340
16	frbl	foot	0.514	foot	0.679	foot	1.009	foot	0.597
17	fbrl	foot	1.224	foot	0.583	foot	0.785	foot	0.650
18	lrfb	foot	0.986	foot	1.021	foot	1.325	foot	0.976
19	rflb	head*	0.356	foot	0.982	tap	0.532	tap	0.374
20	lbfr	tap left	0.452	foot	0.538	tap	0.643	tap	0.488
21	rflb	none	0.348	none	0.330	none	0.315	none	0.321
22	lbfr	tap	0.682	foot	0.721	foot	1.362	foot	0.776
23	fblr	foot	0.672	foot	0.403	foot	0.768	foot	0.893
24	flrb	none	0.341	none	0.328	none	0.338	none	0.354
25	blfr	foot*	0.342	foot	0.374	foot	0.748	foot	0.691
26	lrbf	head*	0.350	head*	0.332	tap	0.364	head*	0.349
27	lrfb	foot	0.349	foot	0.509	foot	0.596	foot	0.571
28	flbr	tap	0.367	foot*	0.344	none	0.341	none	0.375
29	rblf	foot	0.642	foot	0.331	tap	0.467	tap	0.492
30	lfrb	foot	1.303	foot	1.791	foot	0.923	foot	1.111
31	rlbf	tap	0.345	foot	0.403	tap	0.482	tap	0.425
32	rlbf	tap	0.532	tap	0.545	tap	0.558	tap	0.550

A.4 VOCALICS

In the following, screenshots from the vocalics experiment described in Section 4.4 are shown. The collected key research data is provided thereafter. The videos presented to the participants during the study are available on YouTube¹.

A.4.1 *Survey Pages*

The following screenshots display the survey pages shown to the study participants.

¹<https://www.youtube.com/playlist?list=PLPhCA5Y9lesmAPD5TRXISybvMxNve0jA>

Herzlich Willkommen

Über diese Studie

In dieser Studie werden Ihnen zwei kurze Videos gezeigt, in denen eine Drohne jeweils eine bestimmte Bewegung ausführt. Anschließend werden Ihnen zwei Audiodateien vorgespielt. Es ist Voraussetzung für diese Studie, dass Ihr Gerät über eine Audioausgabe verfügt. Stellen Sie bitte jetzt sicher, dass der Ton eingeschaltet ist.

Die Dauer Ihrer Teilnahme an der Studie wird etwa zwei Minuten betragen. Die Größe der zu übertragenden Medien beträgt unter 10 MB.

Allgemeine Hinweise und Einverständniserklärung

Um an dieser Studie teilnehmen zu können, müssen Sie mindestens 18 Jahre alt sein. Sie dürfen nicht mehr als einmal teilnehmen. Ihre Entscheidung, an dieser Studie teilzunehmen, ist freiwillig. Sie können jederzeit ohne Angabe von Gründen von der Studie zurücktreten. In dem Fall werden alle erhobenen Daten gelöscht. Ihre Antworten werden in anonymer Form gesammelt. Weder Ihre IP-Adresse, Ihr Name, Ihr Standort noch andere Daten, die Sie identifizieren könnten, werden erfasst. Neben Ihren Antworten werden folgende Daten erhoben:

- Name und Version des Browsers,
- Datum und Uhrzeit des Zugriffs,
- Dauer des Aufenthalts auf einzelnen Seiten der Umfrage,
- die Seitenzahl der Umfrageseite, auf der die Teilnahme beendet wurde.

- die Seitenzahl der Umfrageseite, auf der die Teilnahme beendet wurde.

Da die Umfrage an einer deutschen Hochschule durchgeführt wird, werden die erhobenen Daten auf den Hochschulserver in Wiesbaden, Deutschland, übertragen, anonymisiert für 10 Jahre gespeichert und im Anschluss gelöscht.

Die aggregierten Ergebnisse dieser Studie können auf wissenschaftlichen Konferenzen vorgestellt, in wissenschaftlichen Zeitschriften veröffentlicht oder auf andere Weise in anonymisierter Form an die wissenschaftliche Gemeinschaft weitergegeben werden.

Wenn Sie Fragen zu dieser Studie haben oder bei Veröffentlichung über die Forschungsergebnisse informiert werden möchten, senden Sie bitte eine E-Mail an marc.lieser@hs-rm.de.

Indem Sie fortfahren, bestätigen Sie, dass Sie diese Einverständniserklärung gelesen und verstanden haben und an der Studie teilnehmen möchten.

Weiter

Allgemein

Bitte beantworten Sie die folgenden Fragen.

Wie alt sind Sie?

Mit welchem Geschlecht identifizieren Sie sich?

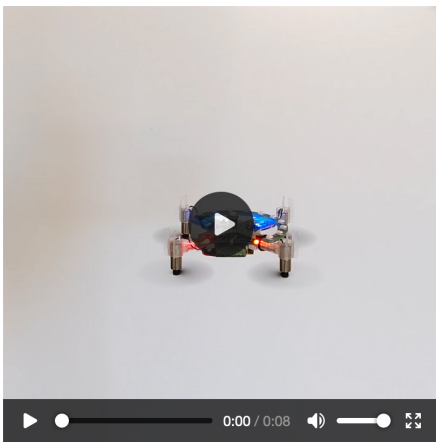
Ohne Angabe ▾

Was sind Sie von Beruf bzw. was studieren Sie?

Weiter

Video 1 von 2

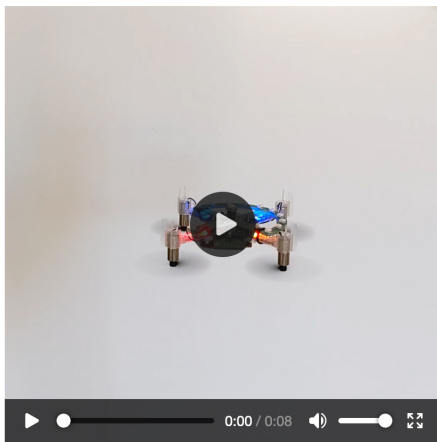
Bitte sehen Sie sich das erste Video an.



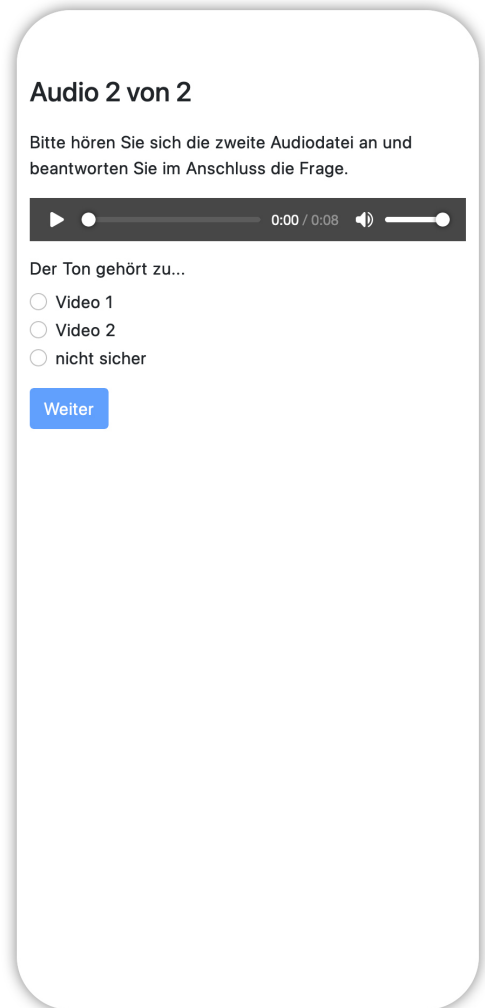
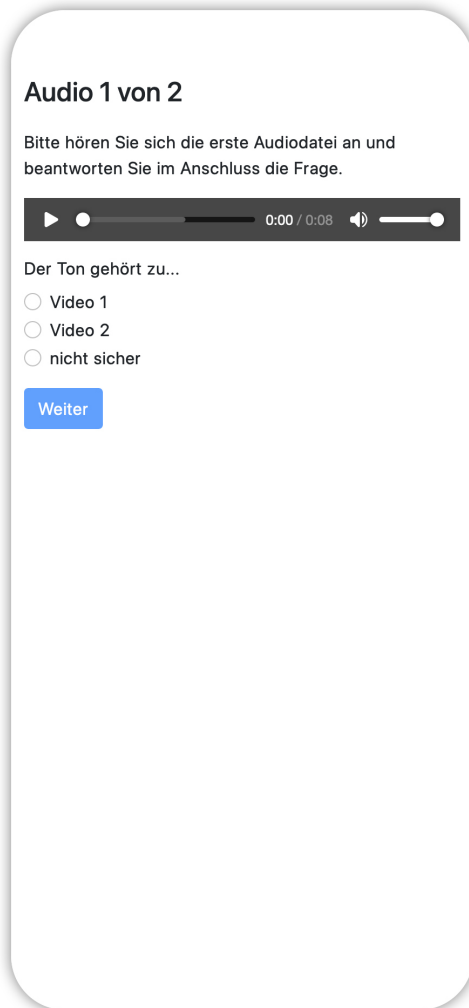
Weiter

Video 2 von 2

Bitte sehen Sie sich das zweite Video an.



Weiter



Vielen Dank

Vielen Dank für Ihre Teilnahme!

Bei Fragen zur Studie, senden Sie bitte eine E-Mail an marc.lieser@hs-rm.de.

Sie können diesen Tab nun schließen.

A.4.2 *Research Data*

The data collected in the vocalics online survey is presented in Table A.7.

Table A.7: Key research data from the vocalics user study. The study consisted of two groups A and B with four possible media orders (1–4) provided in Table 4.4. Relevant is the "Audio to Video 1/2" assignment, where the provided audio index is assigned to the video index, both depending on the media order. Columns V1, V2, A1, and A2 (A for audio, V for video) provide the durations participants stayed on the respective media page, measured from page load to clicking of the button to move to the next page. The total times are the sum of these four columns and the estimation by oTree's own page times database, however, some entries are missing and some are not plausible.

User	Group	Order	Age	Gender	Time							
					Audio to		per Media Page				Total (Media)	
					V1	V2	V1	V2	A1	A2	Sum	oTree
1	A	1	29	m	2	2	23	11	32	50	116	119
2	A	2	31	m	2	1	22	12	14	10	58	60
3	B	1	37	f	1	2	13	11	17	11	52	55
4	B	2	34	f	2	1	13	10	12	12	47	51
5	B	4	33	m	1	2	12	11	15	11	49	53
6	A	1	29	f	1	2	27	12	13	10	62	65
7	A	3	34	f	2	1	12	9	16	14	51	56
8	A	4	36	m	1	2	32	17	45	15	109	115
9	B	1	33	m	1	2	9	9	13	11	42	56
10	B	3	32	f	2	1	11	12	11	12	46	49
11	A	3	24	f	1	2	13	11	37	17	78	81
12	B	1	31	f	0	2	14	12	22	11	59	61
13	B	2	21	f	2	2	12	11	12	11	46	49
14	B	3	30	f	2	1	10	10	14	12	46	49
15	A	1	25	f	1	2	13	10	12	12	47	50
16	A	2	34	m	2	0	15	13	13	11	52	66
17	A	3	35	m	2	2	11	10	13	12	46	50
18	A	4	35	f	2	0	15	10	14	20	59	66
19	B	1	31	f	2	2	11	10	15	11	47	62
20	B	3	30	f	2	1	12	9	12	12	45	51
21	A	2	33	f	2	2	10	10	12	13	45	52
22	B	2	26	m	2	1	12	9	13	10	44	48

Table A.7 (continued)

23	B	3	32	f	2	1	12	10	13	7	42	72
24	B	4	37	f	2	2	13	12	17	13	55	60
25	A	4	40	m	2	0	13	12	25	13	63	66
26	B	1	31	f	2	1	14	10	42	20	86	92
27	B	4	36	m	1	2	15	11	14	13	53	59
28	A	2	41	m	2	1	13	11	15	11	50	53
29	A	3	31	f	1	0	12	11	14	14	51	54
30	A	4	33	f	1	2	11	10	14	15	50	52
31	A	3	31	f	1	1	11	13	17	10	51	61
32	A	4	32	f	1	2	11	10	12	12	45	49
33	B	3	44	m	2	1	63	14	22	11	110	2637
34	A	2	44	m	2	1	14	12	12	13	51	54
35	A	4	38	f	2	1	10	12	16	10	48	
36	B	2	35	f	2	1	16	11	13	12	52	55
37	B	4	42	f	2	0	10	5	22	14	51	15 858
38	A	2	34	f	2	1	12	10	13	11	46	
39	A	3	36	d	1	1	12	11	11	10	44	47
40	B	4	40	u	1	2	14	11	14	11	50	84
41	A	2	30	m	2	1	11	10	12	11	44	54
42	A	3	36	f	2	1	15	11	14	11	51	55
43	A	4	36	m	2	2	25	11	14	14	64	66
44	B	1	40	f	2	1	13	16	14	12	55	57
45	B	4	38	m	1	2	13	11	15	14	53	56
46	A	1	43	m	1	2	11	10	13	11	45	49
47	A	4	31	m	2	1	12	11	13	13	49	51
48	B	3	38	m	2	1	44	13	14	15	86	89
49	B	4	64	f	2	0	44	12	25	19	100	106
50	A	3	19	m	2	1	17	10	12	10	49	51
51	A	4	21	m	1	2	16	10	11	10	47	52
52	B	2	40	m	2	1	12	11	24	14	61	64
53	B	3	43	m	2	2	13	10	18	14	55	59
54	B	4	40	f	1	2	11	10	12	11	44	47
55	A	2	30	f	2	2	11	10	11	14	46	51
56	A	3	21	m	2	1	11	10	11	10	42	45
57	A	4	34	f	0	0	21	9	13	10	53	57
58	B	2	50	m	2	1	16	124	13	13	166	169
59	B	3	41	m	1	0	10	14	16	12	52	60
60	A	1	29	f	2	1	13	15	17	14	59	68

Table A.7 (continued)

61	A	2	42	m	2	1	14	11	12	11	48	52
62	A	3	38	f	1	2	10	10	18	13	51	58
63	A	4	32	f	2	1	12	11	11	12	46	49
64	B	1	38	f	2	0	13	13	22	21	69	74
65	B	2	38	m	2	1	13	12	16	13	54	56
66	B	4	30	f	1	2	70	10	12	10	102	104
67	A	1	33	m	1	2	13	11	14	11	49	56
68	A	2	21	f	2	1	10	9	11	10	40	44
69	A	3	51	f	1	2	19	14	41	19	93	98
70	B	1	35	m	1	2	23	12	18	10	63	68
71	B	2	39	m	2	2	13	11	15	11	50	55
72	B	3	34	f	2	1	12	11	15	11	49	53
73	A	1	69	m	2	1	13	10	21	12	56	87
74	A	2	24	m	2	1	13	10	14	10	47	51
75	B	1	36	m	1	2	21	12	17	11	61	63
76	B	4	43	f	1	2	15	11	28	13	67	74
77	A	1	32	f	2	1	15	11	12	11	49	52
78	A	2	34	m	2	0	14	11	12	11	48	55
79	A	4	27	f	2	1	13	12	20	12	57	60
80	B	1	33	m	1	2	13	11	16	10	50	54
81	B	2	27	u	2	1	11	10	16	10	47	50
82	B	4	26	m	1	2	10	15	19	12	56	63
83	A	2	41	f	2	1	13	10	17	15	55	58
84	A	3	40	f	1	2	12	11	14	10	47	49
85	A	4	45	m	2	1	10	9	19	10	48	55
86	A	2	41	f	2	1	18	12	16	14	60	
87	A	4	61	f	2	1	13	12	15	12	52	55
88	B	1	45	m	2	0	12	10	23	13	58	62
89	A	2	33	m	2	2	12	10	26	13	61	74
90	B	1	27	m	1	2	10	9	18	11	48	55
91	A	1	30	m	1	2	11	10	21	13	55	64
92	B	1	41	m	2	1	13	11	15	11	50	53
93	B	4	24	m	2	2	13	11	17	10	51	55
94	A	1	44	m	2	1	31	7	20	10	68	78
95	A	2	41	m	0	0	14	11	16	14	55	59
96	A	2	46	m	2	1	29	13	22	11	75	79
97	A	3	41	m	2	1	12	11	27	11	61	65
98	A	4	21	f	2	1	29	13	24	14	80	87

Table A.7 (continued)

99	B	1	34	f	1	1	17	11	27	40	95	99
100	A	1	41	m	1	2	13	11	19	22	65	99
101	A	3	45	m	2	0	14	20	27	12	73	76
102	A	4	30	m	1	1	24	10	14	13	61	72
103	A	1	27	m	2	2	13	9	12	11	45	52
104	A	2	35	m	2	1	12	10	11	10	43	49
105	A	3	36	m	2	1	12	10	16	21	59	203
106	A	4	45	m	2	2	15	11	14	10	50	55
107	B	2	40	f	2	1	25	9	17	14	65	73
108	A	1	40	m	1	2	11	10	14	10	45	49
109	A	2	38	m	2	1	22	10	21	12	65	69
110	B	3	36	f	2	1	12	9	13	10	44	49
111	B	4	34	m	1	2	13	12	14	12	51	53
112	A	4	40	m	0	0	12	11	15	10	48	51
113	A	1	40	f	0	0	12	10	17	12	51	54
114	B	3	28	f	2	1	13	10	13	12	48	51
115	B	4	29	f	2	2	14	12	13	11	50	55
116	A	1	32	f	2	1	15	10	12	12	49	54
117	A	4	41	m	1	2	12	13	38	16	79	87
118	B	1	32	m	1	2	12	10	13	10	45	49
119	A	3	34	m	2	1	11	10	15	11	47	53
120	A	4	32	m	2	2	16	11	12	10	49	55
121	B	1	23	f	1	2	10	9	17	10	46	52
122	A	1	49	m	2	2	16	11	18	14	59	64
123	A	2	61	f	2	2	15	25	17	16	73	75
124	A	3	34	f	2	1	34	11	11	11	67	96
125	A	4	34	m	1	2	11	10	13	11	45	64
126	B	1	40	m	1	2	12	10	13	10	45	47
127	A	2	42	m	2	1	14	12	16	12	54	57
128	B	1	39	f	2	0	12	10	16	7	45	47
129	B	3	40	m	2	1	29	15	14	11	69	81
130	A	2	33	f	1	2	12	10	14	11	47	50
131	B	3	33	f	2	1	88	9	12	12	121	136
132	A	3	32	f	2	1	13	10	13	11	47	49
133	B	3	36	m	2	1	23	11	15	18	67	74
134	A	4	38	m	1	2	14	12	16	11	53	55
135	B	2	35	m	2	1	28	12	11	12	63	66
136	B	3	28	m	1	1	14	10	15	10	49	52

Table A.7 (continued)

137	A	1	33	f	2	2	14	12	17	12	55	57
138	A	2	39	f	2	1	27	13	26	19	85	87
139	B	1	33	f	1	2	25	20	20	12	77	80
140	B	1	40	f	0	2	14	13	19	17	63	98
141	B	2	36	m	2	1	21	14	16	14	65	71
142	B	4	26	f	1	2	17	11	13	10	51	55
143	A	1	28	f	1	1	11	9	12	12	44	48
144	A	3	32	m	2	1	12	10	14	11	47	52
145	B	4	39	m	1	2	12	9	17	14	52	59
146	A	3	50	m	2	1	17	12	18	11	58	60
147	B	2	43	f	2	1	10	10	13	14	47	58
148	A	2	44	f	2	0	16	12	14	22	64	68
149	A	3	41	f	2	2	54	11	14	23	102	104
150	B	2	40	m	2	1	13	9	14	9	45	71
151	A	1	37	m	2	1	12	11	15	13	51	56
152	B	1	33	f	2	2	12	11	13	11	47	50
153	B	2	41	m	2	1	13	10	19	14	56	61
154	B	4	37	m	1	2	17	13	12	11	53	55
155	B	1	31	m	2	0	13	11	14	12	50	52
156	B	2	37	m	2	1	13	12	12	11	48	50
157	A	1	36	f	2	2	13	13	13	12	51	53
158	B	2	34	f	2	0	14	11	19	16	60	65
159	A	3	31	f	1	2	13	11	13	19	56	61
160	B	1	37	m	1	2	14	10	18	10	52	57
161	B	4	29	m	2	1	11	10	11	10	42	45
162	A	3	36	m	2	1	12	10	15	11	48	59
163	B	1	45	m	2	1	23	11	21	12	67	70
164	B	3	43	m	2	1	20	14	17	11	62	65
165	B	3	41	m	2	1	15	11	13	12	51	
166	B	2	33	m	1	2	12	10	20	12	54	58
167	B	3	34	m	2	1	13	10	11	12	46	48
168	B	3	38	m	2	1	26	12	14	11	63	66
169	B	2	41	m	2	1	24	9	27	11	71	77
170	B	4	43	m	1	2	14	10	12	11	47	49
171	B	2	46	m	2	1	15	11	14	16	56	59
172	B	3	32	m	2	1	12	11	12	10	45	53
173	A	1	40	f	2	1	14	12	23	10	59	66
174	B	4	36	m	1	2	13	11	15	11	50	56

Table A.7 (continued)

175	B	2	33	f	2	1	11	10	10	10	41	42
176	B	3	35	f	2	1	13	10	13	12	48	52
177	B	4	38	m	1	2	14	12	18	44	88	114
178	A	4	32	m	2	1	11	11	11	11	44	48
179	A	1	31	f	1	2	12	11	10	9	42	48
180	B	2	33	f	2	1	11	13	12	10	46	
181	B	3	37	m	2	0	12	13	12	8	45	53
182	B	4	36	f	2	1	14	11	13	10	48	50
183	B	4	66	f	0	0	14	14	18	14	60	
184	A	1	35	f	2	0	21	18	14	12	65	79
185	A	4	29	f	2	0	12	10	12	11	45	56
186	B	2	33	f	1	2	14	10	19	12	55	59
187	B	3	31	f	2	1	12	10	14	10	46	
188	A	1	34	f	1	2	12	12	14	10	48	52
189	A	3	28	f	2	1	14	11	24	10	59	63
190	B	2	28	f	2	1	20	10	12	10	52	63
191	B	3	34	f	0	2	10	10	10	10	40	
192	B	4	59	f	0	0	23	14	32	42	111	119

B

PUBLICATIONS

The following is a list of conference contributions that are part of this thesis.

- [Lie+17] Marc Lieser, Henning Tjaden, Robert Brylka, Lasse Löffler, and Ulrich Schwanecke. “A Low-Cost Mobile Infrastructure for Compact Aerial Robots Under Supervision”. In: *2017 European Conference on Mobile Robots (ECMR)*. Paris, France, Sept. 2017, pp. 1–6. DOI: 10.1109/ECMR.2017.8098678.
- [LSB21a] Marc Lieser, Ulrich Schwanecke, and Jörg Berdux. “Evaluating Distances in Tactile Human-Drone Interaction”. In: *30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*. 2021, pp. 1275–1282. DOI: 10.1109/RO-MAN50785.2021.9515313.
- [LSB21b] Marc Lieser, Ulrich Schwanecke, and Jörg Berdux. “EAVE: Emotional Aerial Vehicle Evaluator”. In: *INFORMATIK 2020*. Ed. by Ralf H. Reussner, Anne Koziolk, and Robert Heinrich. Bonn: Gesellschaft für Informatik, 2021, pp. 899–906. DOI: 10.18420/inf2020_81.
- [LSB21c] Marc Lieser, Ulrich Schwanecke, and Jörg Berdux. “Tactile Human-Quadrotor Interaction: MetroDrone”. In: *15th International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '21. Salzburg, Austria: Association for Computing Machinery, 2021. ISBN: 9781450382137. DOI: 10.1145/3430524.3440649.
- [LS24] Marc Lieser and Ulrich Schwanecke. “Vocalics in Human-Drone Interaction”. In: *33rd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. Pasadena, California, USA, 2024, pp. 2226–2232. DOI: 10.1109/RO-MAN60168.2024.10731428.

BIBLIOGRAPHY

- [ABD17] Urja Acharya, Alisha Bevins, and Brittany A. Duncan. “Investigation of human-robot comfort with a small Unmanned Aerial Vehicle compared to a ground robot”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2017, pp. 2758–2765. DOI: 10.1109/IROS.2017.8206104.
- [Abt+17] Parastoo Abtahi, David Y. Zhao, Jane L. E., and James A. Landay. “Drone Near Me: Exploring Touch-Based Human-Drone Interaction”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (Sept. 2017). DOI: 10.1145/3130899.
- [AF18] Mauro Avila Soto and Markus Funk. “Look, a Guidance Drone! Assessing the Social Acceptability of Companion Drones for Blind Travelers in Public Spaces”. In: *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ASSETS ’18. Galway, Ireland: Association for Computing Machinery, 2018, pp. 417–419. DOI: 10.1145/3234695.3241019.
- [AFH15] Mauro Avila Soto, Markus Funk, and Niels Henze. “DroneNavigator: Using Drones for Navigating Visually Impaired Persons”. In: *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. ASSETS ’15. Lisbon, Portugal: Association for Computing Machinery, 2015, pp. 327–328. DOI: 10.1145/2700648.2811362. URL: <https://doi.org/10.1145/2700648.2811362>.
- [ÅH06] K.K.J. Åström and T. Hägglund. *Advanced PID Control*. ISA-The Instrumentation, Systems, and Automation Society, 2006.
- [Al +16] Majed Al Zayer, Sam Tregillus, Jiwan Bhandari, Dave Feil-Seifer, and Eelke Folmer. “Exploring the Use of a Drone to Guide Blind Runners”. In: *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. ASSETS ’16. Reno, Nevada, USA: Association for Computing Machinery, 2016, pp. 263–264. ISBN: 9781450341240. DOI: 10.1145/2982142.2982204. URL: <https://doi.org/10.1145/2982142.2982204>.

- [Alh+20] Aysha Alharam, Ebrahim Almansoori, Wael Elmadeny, and Hasan Alnoiami. “Real time AI-based pipeline inspection using drone for oil and gas industries in Bahrain”. In: *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*. IEEE, 2020, pp. 1–5.
- [And15] Daniel Andrés López. “Rettungsroboter mit fliegendem Periskop”. Semester Project. Germany: RheinMain University of Applied Sciences, Wiesbaden, July 2015.
- [And16] Daniel Andrés López. “Kombination von Outside-In und Inside-Out Tracking zur Regelung autonomer Luftfahrzeuge”. Master’s Thesis. Germany: RheinMain University of Applied Sciences, Wiesbaden, Nov. 2016.
- [Auc+23] Emanuele Aucone, Steffen Kirchgeorg, Alice Valentini, Loïc Pellissier, Kristy Deiner, and Stefano Mintchev. “Drone-assisted collection of environmental DNA from tree branches for biodiversity monitoring”. In: *Science Robotics* 8.74 (2023), eadd5762. DOI: 10.1126/scirobotics.add5762. eprint: <https://www.science.org/doi/pdf/10.1126/scirobotics.add5762>. URL: <https://www.science.org/doi/abs/10.1126/scirobotics.add5762>.
- [Aud+21] Jonas Auda, Martin Weigel, Jessica R. Cauchard, and Stefan Schneegass. “Understanding Drone Landing on the Human Body”. In: *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*. MobileHCI ’21. Toulouse & Virtual, France: Association for Computing Machinery, 2021. DOI: 10.1145/3447526.3472031.
- [Avi+17] Mauro Avila Soto, Markus Funk, Matthias Hoppe, Robin Boldt, Katrin Wolf, and Niels Henze. “DroneNavigator: Using Leashed and Free-Floating Quadcopters to Navigate Visually Impaired Travelers”. In: *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. ASSETS ’17. Baltimore, Maryland, USA: Association for Computing Machinery, 2017, pp. 300–304. ISBN: 9781450349260. DOI: 10.1145/3132525.3132556. URL: <https://doi.org/10.1145/3132525.3132556>.
- [AZ12] F. Allgöwer and A. Zheng. *Nonlinear Model Predictive Control*. Progress in Systems and Control Theory. Birkhäuser Basel, 2012. ISBN: 9783034884075.

- [Bal+] Bob Balaram et al. “Mars Helicopter Technology Demonstrator”. In: *2018 AIAA Atmospheric Flight Mechanics Conference*. DOI: 10.2514/6.2018-0023.
- [Bar+20] Christoph Bartneck, Tony Belpaeme, Friederike Eyssel, Takayuki Kanda, Merel Keijsers, and Selma Šabanović. *Human-Robot Interaction – An Introduction*. Cambridge University Press, 2020.
- [BAR47] G. A. BARNARD. “SIGNIFICANCE TESTS FOR 2×2 TABLES”. In: *Biometrika* 34.1-2 (Jan. 1947), pp. 123–138. ISSN: 0006-3444. DOI: 10.1093/biomet/34.1-2.123. eprint: <https://academic.oup.com/biomet/article-pdf/34/1-2/123/552350/34-1-2-123.pdf>. URL: <https://doi.org/10.1093/biomet/34.1-2.123>.
- [BD21] Alisha Bevins and Brittany A. Duncan. “Aerial Flight Paths for Communication: How Participants Perceive and Intend to Respond to Drone Movements”. In: *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’21. Boulder, CO, USA: Association for Computing Machinery, 2021, pp. 16–23. ISBN: 9781450382892. DOI: 10.1145/3434073.3444645. URL: <https://doi.org/10.1145/3434073.3444645>.
- [Bha+19] S. M. Bhagya et al. “An Exploratory Study on Proxemics Preferences of Humans in Accordance with Attributes of Service Robots”. In: *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 2019, pp. 1–7. DOI: 10.1109/RO-MAN46459.2019.8956297.
- [BL94] Margaret M. Bradley and Peter J. Lang. “Measuring emotion: The self-assessment manikin and the semantic differential”. In: *Journal of Behavior Therapy and Experimental Psychiatry* 25.1 (1994), pp. 49–59. DOI: 10.1016/0005-7916(94)90063-9.
- [Cac+16] Jonathan Cacace, Alberto Finzi, Vincenzo Lippiello, Michele Furci, Nicola Mimmo, and Lorenzo Marconi. “A control architecture for multiple drones operated via multimodal interaction in search & rescue mission”. In: *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2016, pp. 233–239. DOI: 10.1109/SSRR.2016.7784304.

- [Cal+16] Jules C. Calella, Francisco R. Ortega, Naphtali Rishe, Jonathan F. Bernal, and Armando Barreto. “HandMagic: Towards user interaction with inertial measuring units”. In: *2016 IEEE SENSORS*. 2016, pp. 1–3. DOI: 10.1109/ICSENS.2016.7808524.
- [Cau+15] Jessica R. Cauchard, Jane L. E, Kevin Y. Zhai, and James A. Landay. “Drone & Me: An Exploration into Natural Human-drone Interaction”. In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp ’15. Osaka, Japan: ACM, 2015, pp. 361–365. DOI: 10.1145/2750858.2805823.
- [Cau+16] Jessica R. Cauchard, Kevin Y. Zhai, Marco Spadafora, and James A. Landay. “Emotion encoding in Human-Drone Interaction”. In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Mar. 2016, pp. 263–270. DOI: 10.1109/HRI.2016.7451761.
- [CCC17] Victoria Chang, Pramod Chundury, and Marshini Chetty. “Spiders in the Sky: User Perceptions of Drones, Privacy, and Security”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI ’17. Denver, Colorado, USA: ACM, 2017, pp. 6765–6776. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025632.
- [Col+17] Ashley Colley, Lasse Virtanen, Pascal Knierim, and Jonna Häkkinä. “Investigating Drone Motion as Pedestrian Guidance”. In: *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*. MUM ’17. Stuttgart, Germany: Association for Computing Machinery, 2017, pp. 143–150. ISBN: 9781450353786. DOI: 10.1145/3152832.3152837.
- [Cor+21] Evangeline Corcoran, Megan Winsen, Ashlee Sudholz, and Grant Hamilton. “Automated detection of wildlife using drones: Synthesis, opportunities and constraints”. In: *Methods in Ecology and Evolution* 12.6 (2021), pp. 1103–1114.
- [Cor13] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. 1st ed. Springer Publishing Company, Incorporated, 2013.
- [CSW16] Daniel L. Chen, Martin Schonger, and Chris Wickens. “oTree—An open-source platform for laboratory, online, and field experiments”. In: *Journal of Behavioral and Experimental Finance* 9 (2016), pp. 88–97. ISSN: 2214-6350. DOI: <https://doi.org/10.1016/j.jbef.2015.12.001>. URL: <https://www.sciencedirect.com/science/article/pii/S2214635016000101>.

- [CWE99] Stanley Coren, Lawrence M. Ward, and James T. Enns. *Sensation and Perception*. Harcourt Brace College Publishers, 1999. ISBN: 9780155080508. DOI: 10.1002/9781118133880.hop201007.
- [Dar72] Charles Darwin. *The Expression of the Emotions in Man and Animals*. eng. Ed. by Francis Darwin. London: John Murray, 1872. ISBN: 978-1-108-00483-1.
- [Dau+22] Sharifah Mastura Syed Mohd Daud et al. “Applications of drone in disaster management: A scoping review”. In: *Science & Justice* 62.1 (2022), pp. 30–42.
- [DB16] Richard C. Dorf and Robert H. Bishop. *Modern Control Systems*. 13. Pearson Education, 2016.
- [Den+18] Honghao Deng, Jiabao Li, Allen Sayegh, Sebastian Birolini, and Stefano Andreani. “Twinkle: A Flying Lighting Companion for Urban Safety”. In: *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI ’18. Stockholm, Sweden, 2018, pp. 567–573. DOI: 10.1145/3173225.3173309.
- [DM13] B. A. Duncan and R. R. Murphy. “Comfortable approach distance with small Unmanned Aerial Vehicles”. In: *2013 RO-MAN*. 2013, pp. 786–792.
- [ESC14] Jakob Engel, Jürgen Sturm, and Daniel Cremers. “Scale-aware navigation of a low-cost quadrocopter with a monocular camera”. In: *Robotics and Autonomous Systems* 62.11 (2014), pp. 1646–1656.
- [Fae+14] M. Faessler, E. Mueggler, K. Schwabe, and D. Scaramuzza. “A monocular pose estimation system based on infrared LEDs”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. May 2014, pp. 907–913. DOI: 10.1109/ICRA.2014.6906962.
- [Fal+18] Davide Falanga, Philipp Foehn, Peng Lu, and Davide Scaramuzza. “PAMPC: Perception-Aware Model Predictive Control for Quadrotors”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IROS ’18. Oct. 2018, pp. 1–8. DOI: 10.1109/IROS.2018.8593739.
- [FFS17] Matthias Faessler, Davide Falanga, and Davide Scaramuzza. “Thrust Mixing, Saturation, and Body-Rate Control for Accurate Aggressive Quadrotor Flight”. In: *IEEE Robot. Autom. Lett.* 2.2 (Apr. 2017), pp. 476–482. ISSN: 2377-3766. DOI: 10.1109/LRA.2016.2640362.

- [FFS18] Matthias Faessler, Antonio Franchi, and Davide Scaramuzza. “Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories”. In: *IEEE Robotics and Automation Letters* 3.2 (Apr. 2018), pp. 620–626. DOI: 10.1109/LRA.2017.2776353.
- [FH15] G. Farin and D. Hansford. *Practical Linear Algebra: A Geometry Toolbox, Third Edition*. CRC Press, 2015. ISBN: 9781498759779.
- [FM12] David Feil-Seifer and Maja J Matarić. “Distance-Based Computational Models for Facilitating Robot Interaction with Children”. In: *J. Hum.-Robot Interact.* 1.1 (July 2012), pp. 55–77. DOI: 10.5898/JHRI.1.1.Feil-Seifer. URL: <https://doi.org/10.5898/JHRI.1.1.Feil-Seifer>.
- [För15] Julian Förster. “System Identification of the Crazyflie 2.0 Nano Quadcopter”. Bachelor’s Thesis. Switzerland: ETH Zurich, Aug. 2015. DOI: 10.3929/ethz-b-000214143.
- [Fra19] Eitan Frachtenberg. “Practical Drone Delivery”. In: *Computer* 52.12 (2019), pp. 53–57. DOI: 10.1109/MC.2019.2942290.
- [GC23] Eyal Ginosar and Jessica R. Cauchard. “At First Light: Expressive Lights in Support of Drone-Initiated Communication”. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI ’23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 9781450394215. DOI: 10.1145/3544548.3581062. URL: <https://doi.org/10.1145/3544548.3581062>.
- [GKP03] J. Goetz, S. Kiesler, and A. Powers. “Matching robot appearance and behavior to tasks to improve human-robot cooperation”. In: *The 12th IEEE International Workshop on Robot and Human Interactive Communication*. 2003, pp. 55–60. DOI: 10.1109/ROMAN.2003.1251796.
- [Gom+16] Antonio Gomes, Calvin Rubens, Sean Braley, and Roel Vertegaal. “BitDrones: Towards Using 3D Nanocopter Displays As Interactive Self-Levitating Programmable Matter”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI ’16. ACM, 2016, pp. 770–780. DOI: 10.1145/2858036.2858519.
- [Goo96] Walter A. Good. *Autobiography of Dr. Walter (Walt) A. Good*. Mar. 1996. URL: <https://www.modelaircraft.org/sites/default/files/GoodDrWalterAWalt.pdf> (visited on 12/25/2023).

- [Gra17] R. G. Grant. *Flight: The Complete History of Aviation*. Dorling Kindersley Limited, 2017.
- [Gro+19] Boris Gromov, Jérôme Guzzi, Luca Gambardella, and Alessandro Giusti. “Demo: Pointing Gestures for Proximity Interaction”. In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2019, pp. 665–665. DOI: 10.1109/HRI.2019.8673329.
- [GT11] M. J. Gielniak and A. L. Thomaz. “Generating anticipation in robot motion”. In: *2011 RO-MAN*. 2011, pp. 449–454.
- [Gui15] Erico Guizzo. *Jibo Is as Good as Social Robots Get. But Is That Good Enough?* 2015. URL: <https://spectrum.ieee.org/jibo-is-as-good-as-social-robots-get-but-is-that-good-enough> (visited on 01/02/2024).
- [Gut15] Annalena Gutheil. “Multicopter-Steuerung mittels Kinect”. Semester Project. Germany: RheinMain University of Applied Sciences, Wiesbaden, July 2015.
- [HA17] Wolfgang Hönig and Nora Ayanian. “Flying Multiple UAVs Using ROS”. In: *Robot Operating System (ROS): The Complete Reference (Volume 2)*. Springer International Publishing, 2017, pp. 83–118. DOI: 10.1007/978-3-319-54927-9_3.
- [Hal66] Edward T. Hall. *The hidden dimension*. 1st ed. Doubleday Garden City, N.Y, 1966.
- [HBV16] Guy Hoffman, Shira Bauman, and Keinan Vanunu. “Robotic Experience Companionship in Music Listening and Video Watching”. In: *Personal Ubiquitous Comput.* 20.1 (Feb. 2016), pp. 51–63. ISSN: 1617-4909. DOI: 10.1007/s00779-015-0897-1.
- [HJ14] Guy Hoffman and Wendy Ju. “Designing Robots with Movement in Mind”. In: *J. Hum.-Robot Interact.* 3.1 (Feb. 2014), pp. 91–122. DOI: 10.5898/JHRI.3.1.Hoffman.
- [Hof07] Guy Hoffman. “Ensemble: Fluency and Embodiment for Robots Acting with Humans”. <http://hdl.handle.net/1721.1/41705>. Retrieved: 2024-01-02. PhD Thesis. Cambridge, Massachusetts, United States: Massachusetts Institute of Technology, 2007.

- [Hue+06] Helge Huettenrauch, Kerstin Severinson Eklundh, Anders Green, and Elin A. Topp. “Investigating Spatial Relationships in Human-Robot Interaction”. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006, pp. 5052–5059. DOI: 10.1109/IROS.2006.282535.
- [Hug04] Peter C. Hughes. *Spacecraft Attitude Dynamics*. Dover Books on Aeronautical Engineering. Dover Publications, 2004. ISBN: 9780486439259.
- [HV13] Guy Hoffman and Keinan Vanunu. “Effects of robotic companionship on music enjoyment and agent perception”. In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2013, pp. 317–324. DOI: 10.1109/HRI.2013.6483605.
- [Hwa+17] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter. “Control of a Quadrotor With Reinforcement Learning”. In: *IEEE Robotics and Automation Letters* 2.4 (Oct. 2017), pp. 2096–2103. DOI: 10.1109/LRA.2017.2720851.
- [HZ03] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press, 2003.
- [Inc21] Draganfly Inc. *About Us*. <https://draganfly.com/about-us/>. (Online; accessed 2021-04-14). 2021.
- [Ive+21] Nicolai Iversen, Oscar Bowen Schofield, Linda Cousin, Naeem Ayoub, Gerd Vom Bögel, and Emad Ebeid. “Design, integration and implementation of an intelligent and self-recharging drone system for autonomous power line inspection”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 4168–4175.
- [JB20] Yuhan Jiang and Yong Bai. “Estimation of construction site elevations using drone-based orthoimagery and deep learning”. In: *Journal of Construction Engineering and Management* 146.8 (2020), p. 04020086.
- [JHK18] Walther Jensen, Simon Hansen, and Hendrik Knoche. “Knowing You, Seeing Me: Investigating User Preferences in Drone-Human Acknowledgement”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI ’18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–12. DOI: 10.1145/3173574.3173939.

- [Joh91] Oliver P John. “The big five inventory—versions 4a and 54”. In: (*No Title*) (1991).
- [Jon+16] Brennan Jones et al. “Elevating Communication, Collaboration, and Shared Experiences in Mobile Video through Drones”. In: *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. DIS ’16. Brisbane, QLD, Australia: Association for Computing Machinery, 2016, pp. 1123–1135. ISBN: 9781450340311. DOI: 10.1145/2901790.2901847. URL: <https://doi.org/10.1145/2901790.2901847>.
- [Jon22] Michael Jonga. *Drones: the fight against poaching in Africa*. Aug. 12, 2022. URL: <https://www.aerotime.aero/articles/31889-drones-conservation-war-poaching-africa-uav> (visited on 04/08/2023).
- [Jor16] John M. Jordan. *Robots*. 1st ed. The MIT Press, 2016.
- [Kai+17] Jean-Marie Kai, Guillaume Allibert, Minh-Duc Hua, and Tarek Hamel. “Nonlinear feedback control of Quadrotors exploiting First-Order Drag Effects”. In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 8189–8195. DOI: <https://doi.org/10.1016/j.ifacol.2017.08.1267>.
- [KKK09] Jaewook Kim, Sonya S. Kwak, and Myungsuk Kim. “Entertainment robot personality design based on basic factors of motions: A case study with ROLLY”. In: *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*. 2009 RO-MAN. 2009, pp. 803–808. DOI: 10.1109/ROMAN.2009.5326222.
- [KMN09] Hideki Kozima, Marek Piotr Michalowski, and Cocoro Nakagawa. “Kepon: A playful robot for research, therapy, and entertainment”. In: *International Journal of Social Robotics* 1.1 (Jan. 2009), pp. 3–18. DOI: 10.1007/s12369-008-0009-8.
- [Kna72] M.L. Knapp. *Nonverbal Communication in Human Interaction*. Holt, Rinehart and Winston, 1972. ISBN: 9780030858628.
- [Kni+18] Pascal Knierim, Thomas Kosch, Alexander Achberger, and Markus Funk. “Flyables: Exploring 3D Interaction Spaces for Levitating Tangibles”. In: *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI ’18. Stockholm, Sweden: Association for Computing Machinery, 2018, pp. 329–336. ISBN:

9781450355681. DOI: 10.1145/3173225.3173273. URL: <https://doi.org/10.1145/3173225.3173273>.
- [Koel13] Stefan Koelsch. “From Social Contact to Social Cohesion–The 7 Cs”. In: *Music and Medicine* 5 (Oct. 2013), pp. 204–209. DOI: 10.1177/1943862113508588.
- [Kri+15] R. V. Krishna, B. S. Sathish, P. Ganesan, P. J. Babu, and R. Abilash. “Design of voice and gesture controlled Quadcopter”. In: *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. 2015, pp. 1–6. DOI: 10.1109/ICIIECS.2015.7193152.
- [Kun+23] Siya Kunde, Nathan Simms, Gerson Uriarte, and Brittany Duncan. “Let’s Run An Online Proxemics Study! But, How Do Results Compare To In-Person?” In: *Social Robotics: 14th International Conference, ICSR 2022, Florence, Italy, December 13–16, 2022, Proceedings, Part I*. Florence, Italy: Springer-Verlag, 2023, pp. 24–37. ISBN: 978-3-031-24666-1. DOI: 10.1007/978-3-031-24667-8_3. URL: https://doi.org/10.1007/978-3-031-24667-8_3.
- [LA09] Manolis I. A. Lourakis and Antonis A. Argyros. “SBA: A Software Package for Generic Sparse Bundle Adjustment”. In: *ACM Trans. Math. Softw.* 36.1 (Mar. 2009). DOI: 10.1145/1486525.1486527.
- [Lar18] Jan Larwig. “Development of a Neural Network With Reinforcement Learning for Stable Flight Control of a Quadcopter”. Bachelor’s Thesis. Germany: RheinMain University of Applied Sciences, Wiesbaden, June 2018.
- [Len18] Dominic Lenton. “The measure of Volocopter flying taxi”. In: *Engineering & Technology* 13.7/8 (2018), pp. 10–11. DOI: 10.1049/et.2018.0712.
- [Li+19] Rui Li, Marc van Almkerk, Sanne van Waveren, Elizabeth Carter, and Iolanda Leite. “Comparing Human-Robot Proxemics Between Virtual Reality and the Real World”. In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2019, pp. 431–439. DOI: 10.1109/HRI.2019.8673116.

- [Lie+17] Marc Lieser, Henning Tjaden, Robert Brylka, Lasse Löffler, and Ulrich Schwanecke. “A Low-Cost Mobile Infrastructure for Compact Aerial Robots Under Supervision”. In: *2017 European Conference on Mobile Robots (ECMR)*. Paris, France, Sept. 2017, pp. 1–6. DOI: 10.1109/ECMR.2017.8098678.
- [Lie14] Marc Lieser. “Outside-in Tracking-basierte Regelung von Multi-coptern”. Master’s Thesis. Germany: RheinMain University of Applied Sciences, Wiesbaden, May 2014.
- [LP17] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control*. 1st. USA: Cambridge University Press, 2017.
- [LS24] Marc Lieser and Ulrich Schwanecke. “Vocalics in Human-Drone Interaction”. In: *33rd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. Pasadena, California, USA, 2024, pp. 2226–2232. DOI: 10.1109/RO-MAN60168.2024.10731428.
- [LSB21a] Marc Lieser, Ulrich Schwanecke, and Jörg Berdux. “EAVE: Emotional Aerial Vehicle Evaluator”. In: *INFORMATIK 2020*. Ed. by Ralf H. Reussner, Anne Koziolk, and Robert Heinrich. Bonn: Gesellschaft für Informatik, 2021, pp. 899–906. DOI: 10.18420/inf2020_81.
- [LSB21b] Marc Lieser, Ulrich Schwanecke, and Jörg Berdux. “Evaluating Distances in Tactile Human-Drone Interaction”. In: *30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*. 2021, pp. 1275–1282. DOI: 10.1109/RO-MAN50785.2021.9515313.
- [LSB21c] Marc Lieser, Ulrich Schwanecke, and Jörg Berdux. “Tactile Human-Quadrotor Interaction: MetroDrone”. In: *15th International Conference on Tangible, Embedded, and Embodied Interaction*. TEI ’21. Salzburg, Austria: Association for Computing Machinery, 2021. ISBN: 9781450382137. DOI: 10.1145/3430524.3440649.
- [Lup+14] Sergei Lupashin, Markus Hehn, Mark W Mueller, Angela P Schoellig, Michael Sherback, and Raffaello D’Andrea. “A platform for aerial robotics research and demonstration: The Flying Machine Arena”. In: *Mechatronics* 24 (1 2014), pp. 41–54. DOI: 10.1016/j.mechatronics.2013.11.006.

- [Ma+03] Yi Ma, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2003. ISBN: 0-387-00893-4.
- [Mar+07] Landis Markley, Yang Cheng, John Crassidis, and Yaakov Oshman. “Averaging Quaternions”. In: *Journal of Guidance, Control, and Dynamics* 30 (July 2007), pp. 1193–1196. DOI: 10.2514/1.28949.
- [Mar08] Landis Markley. “Unit Quaternion from Rotation Matrix”. In: *Journal of Guidance, Control, and Dynamics* 31 (Mar. 2008), pp. 440–442. DOI: 10.2514/1.31730.
- [MD18] UM Rao Mogili and BBVL Deepak. “Review on application of drone systems in precision agriculture”. In: *Procedia computer science* 133 (2018), pp. 502–509.
- [MHF16] David Matsumoto, Hyisung C. Hwang, and Mark G. Frank. *APA Handbook of Nonverbal Communication*. APA Handbooks in Psychology. American Psychological Association, 2016. ISBN: 9781433819698. DOI: 10.1037/14669-000.
- [Mic+10] Nathan Michael, D. Mellinger, Q. Lindsey, and V. Kumar. “The GRASP Multiple Micro-UAV Testbed”. In: *Robotics Automation Magazine, IEEE* 17.3 (Sept. 2010), pp. 56–65. DOI: 10.1109/MRA.2010.937855.
- [MK11] Daniel Mellinger and Vijay Kumar. “Minimum snap trajectory generation and control for quadrotors”. In: *2011 IEEE International Conference on Robotics and Automation*. ICRA ’11. May 2011, pp. 2520–2525. DOI: 10.1109/ICRA.2011.5980409.
- [MM11] Jonathan Mumm and Bilge Mutlu. “Human-robot proxemics: Physical and psychological distancing in human-robot interaction”. In: *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2011, pp. 331–338. DOI: 10.1145/1957656.1957786.
- [MMT15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. “ORB-SLAM: A Versatile and Accurate Monocular SLAM System”. In: *IEEE Transactions on Robotics* 31.5 (Oct. 2015), pp. 1147–1163.
- [Nag+14] Jawad Nagi, Alessandro Giusti, Gianni A. Di Caro, and Luca M. Gambardella. “Human Control of UAVs Using Face Pose Estimates and Hand Gestures”. In: *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’14. Bielefeld, Germany, 2014, pp. 252–253. DOI: 10.1145/2559636.2559833.

- [Nar+20] Michael D. Narine, Michael D. Narine, Anu G. Bourgeois, and Awad A. Younis Mussa. “Active Noise Cancellation of Drone Propeller Noise through Waveform Approximation and Pitch-Shifting”. Master’s Thesis. Atlanta, Georgia: Georgia State University, May 2020. DOI: <https://doi.org/10.57709/17621233>.
- [Noo+21] Parham Nooralishahi et al. “Drone-based non-destructive inspection of industrial sites: A review and case studies”. In: *Drones* 5.4 (2021), p. 106.
- [NS11] Wai Shan Ng and Ehud Sharlin. “Collocated interaction with flying robots”. In: *IEEE International Workshop on Robot and Human Communication*. 2011 RO-MAN. July 2011, pp. 143–149. DOI: [10.1109/ROMAN.2011.6005280](https://doi.org/10.1109/ROMAN.2011.6005280).
- [NSB11] Yogianandh Naidoo, Riaan Stopforth, and Glen Bright. “Development of an UAV for search & rescue applications”. In: *IEEE Africon '11*. 2011, pp. 1–6. DOI: [10.1109/AFRCOON.2011.6072032](https://doi.org/10.1109/AFRCOON.2011.6072032).
- [NSC13] Tayyab Naseer, Jürgen Sturm, and Daniel Cremers. “FollowMe: Person following and gesture recognition with a quadrocopter”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nov. 2013, pp. 624–630. DOI: [10.1109/IR0S.2013.6696416](https://doi.org/10.1109/IR0S.2013.6696416).
- [NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2. New York, NY, USA: Springer, 2006.
- [ONe44] John J. O’Neil. *Prodigal Genius: The Life of Nikola Tesla*. David McKay Publications, 1944.
- [Ons08] Katrina Onstad. *Pixar Gambles on a Robot in Love*. 2008. URL: <https://www.nytimes.com/2008/06/22/movies/22onst.html> (visited on 01/02/2024).
- [PBK20] Hannah R. M. Pelikan, Mathias Broth, and Leelo Keevallik. ““Are You Sad, Cozmo?”: How Humans Make Sense of a Home Robot’s Emotion Displays”. In: *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’20. Cambridge, United Kingdom: Association for Computing Machinery, 2020, pp. 461–470. ISBN: 9781450367462. DOI: [10.1145/3319502.3374814](https://doi.org/10.1145/3319502.3374814). URL: <https://doi.org/10.1145/3319502.3374814>.

- [Pel+21] Hannah Pelikan, Frederic Anthony Robinson, Leelo Keevallik, Mari Velonaki, Mathias Broth, and Oliver Bown. “Sound in Human-Robot Interaction”. In: *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’21 Companion. Boulder, CO, USA: Association for Computing Machinery, 2021, pp. 706–708. ISBN: 9781450382908. DOI: 10.1145/3434074.3444871. URL: <https://doi.org/10.1145/3434074.3444871>.
- [Per11] Marcus B. Perry. “The Exponentially Weighted Moving Average”. In: *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Ltd, 2011. ISBN: 9780470400531. DOI: <https://doi.org/10.1002/9780470400531.eorms0314>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470400531.eorms0314>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470400531.eorms0314>.
- [Pie16] David Pierce. *Meet the Smartest, Cutest AI-Powered Robot You’ve Ever Seen*. June 2016. URL: <https://www.wired.com/2016/06/anki-cozmo-ai-robot-toy/> (visited on 12/17/2023).
- [Poy93] Fernando Poyatos. *Paralanguage: A linguistic and interdisciplinary approach to interactive speech and sounds*. John Benjamins, 1993. ISBN: 9789027277015. DOI: <https://doi.org/10.1075/cilt.92>.
- [PV23] Pratik Prajapati and Vineet Vashista. “Aerial Physical Human Robot Interaction for Payload Transportation”. In: *IEEE Robotics and Automation Letters* PP (Aug. 2023), pp. 1–8. DOI: 10.1109/LRA.2023.3290410.
- [QS12] Carolyn Quam and Daniel Swingley. “Development in Children’s Interpretation of Pitch Cues to Emotions”. In: *Child Development* 83.1 (2012), pp. 236–250. DOI: <https://doi.org/10.1111/j.1467-8624.2011.01700.x>. eprint: <https://srcd.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8624.2011.01700.x>. URL: <https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8624.2011.01700.x>.
- [RBS17] Sujit Rajappa, Heinrich Bühlhoff, and Paolo Stegagno. “Design and implementation of a novel architecture for physical human-UAV interaction”. In: *The International Journal of Robotics Research* 36.5-7 (2017), pp. 800–819. DOI: 10.1177/0278364917708038.

- [RBV22] Frederic Robinson, Oliver Bown, and Mari Velonaki. “Designing Sound for Social Robots: Candidate Design Principles”. In: *International Journal of Social Robotics* 14 14 (June 2022), pp. 1507–1525. DOI: 10.1007/s12369-022-00891-0. URL: <https://doi.org/10.1007/s12369-022-00891-0>.
- [Res+15] Agoston Restas et al. “Drone applications for supporting disaster management”. In: *World Journal of Engineering and Technology* 3.03 (2015), p. 316.
- [Rig19] Philipp Rigoll. “Autonome Objekterfassung und -inspektion in unbekannter Umgebung mittels monokularer Kamera auf einer Mikro-Drohne”. Master’s Thesis. Germany: RheinMain University of Applied Sciences, Wiesbaden, Aug. 2019.
- [RMD17] J.B. Rawlings, D.Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.
- [RMM18] Francisco Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. “Speeded Up Detection of Squared Fiducial Markers”. In: *Image and Vision Computing* 76 (June 2018). DOI: 10.1016/j.imavis.2018.05.004.
- [RP12] T. Ribeiro and A. Paiva. “The illusion of robotic life: Principles and practices of animation for robots”. In: *HRI ’12*. 2012, pp. 383–390.
- [RVB21] Frederic Anthony Robinson, Mari Velonaki, and Oliver Bown. “Smooth Operator: Tuning Robot Perception Through Artificial Movement Sound”. In: *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’21. Boulder, CO, USA: Association for Computing Machinery, 2021, pp. 53–62. ISBN: 9781450382892. DOI: 10.1145/3434073.3444658. URL: <https://doi.org/10.1145/3434073.3444658>.
- [Sam+12] Carlos Sampedro, Alejandro Rodriguez-Ramos, Hriday Bavle, Adrian Carrio, Paloma de la Puente, and Pascual Campoy. “Development in Children’s Interpretation of Pitch Cues to Emotions”. In: *Child Development* 83.1 (2012), pp. 236–250. DOI: <https://doi.org/10.1111/j.1467-8624.2011.01700.x>. eprint: <https://srcd.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8624.2011.01700.x>. URL: <https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8624.2011.01700.x>.

- [Saq+18] Muhammad Saqib et al. “Real-time drone surveillance and population estimation of marine animals from aerial imagery”. In: *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE. 2018, pp. 1–6.
- [SC17] Patrik Schmuck and Margarita Chli. “Multi-UAV collaborative monocular SLAM”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 3863–3870. DOI: 10.1109/ICRA.2017.7989445.
- [Sch+10] Angela Schöllig, Federico Augugliaro, Sergei Lupashin, and Raffaello D’Andrea. “Synchronizing the motion of a quadcopter to music”. In: *2010 IEEE International Conference on Robotics and Automation*. 2010, pp. 3355–3360. DOI: 10.1109/ROBOT.2010.5509755.
- [Sch07] Dieter Schlüter. *Hubschrauber ferngesteuert*. 14. Neckar-Verlag, 2007.
- [SDW18] Junwon Seo, Luis Duque, and Jim Wacker. “Drone-enabled bridge inspection methodology and application”. In: *Automation in construction* 94 (2018), pp. 112–126.
- [SG08] J. Schupp and J.-Y. Gerlitz. *Big Five Inventory-SOEP (BFI-S)*. Zusammenstellung sozialwissenschaftlicher Items und Skalen (ZIS). 2008. DOI: <https://doi.org/10.6102/zis54>.
- [SG64] Abraham. Savitzky and M. J. E. Golay. “Smoothing and Differentiation of Data by Simplified Least Squares Procedures.” In: *Analytical Chemistry* 36.8 (1964), pp. 1627–1639. DOI: 10.1021/ac60214a047.
- [Sha+13] Megha Sharma, Dale Hildebrandt, Gem Newman, James E. Young, and Rasit Eskicioglu. “Communicating affect via flight path Exploring use of the Laban Effort System for designing affective locomotion paths”. In: *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Mar. 2013, pp. 293–300. DOI: 10.1109/HRI.2013.6483602.
- [Shi+19] ASM Shihavuddin et al. “Wind turbine surface damage detection by deep learning aided drone inspection analysis”. In: *Energies* 12.4 (2019), p. 676.
- [Sho85] Ken Shoemake. “Animating Rotation with Quaternion Curves”. In: *SIGGRAPH Comput. Graph.* 19.3 (July 1985), pp. 245–254. DOI: 10.1145/325165.325242.

- [Shu93] Malcolm D. Shuster. “Survey of attitude representations”. In: *Journal of the Astronautical Sciences* 41.4 (Oct. 1993), pp. 439–517.
- [SK16] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. 2. Berlin, Heidelberg: Springer-Verlag, 2016.
- [SMF14] Daniel Szafir, Bilge Mutlu, and Terrence Fong. “Communication of Intent in Assistive Free Flyers”. In: *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’14. Bielefeld, Germany: Association for Computing Machinery, 2014, pp. 358–365. ISBN: 9781450326582. DOI: 10.1145/2559636.2559672. URL: <https://doi.org/10.1145/2559636.2559672>.
- [SMF15] Daniel Szafir, Bilge Mutlu, and Terry Fong. “Communicating Directionality in Flying Robots”. In: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’15. Portland, Oregon, USA: ACM, 2015, pp. 19–26. ISBN: 978-1-4503-2883-8. DOI: 10.1145/2696454.2696475.
- [SMK17] James Svacha, Kartik Mohta, and Vijay Kumar. “Improving quadrotor trajectory tracking by compensating for aerodynamic effects”. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2017, pp. 860–866. DOI: 10.1109/ICUAS.2017.7991501.
- [SS20] Thomas Sebastian and Christopher Strem. *Toroidal Propeller*. U.S. Patent US10836466B2. 2020.
- [Suá+16] Ramon A. Suárez Fernández, Jose Luis Sanchez-Lopez, Carlos Sampedro, Hriday Bavle, Martin Molina, and Pascual Campoy. “Natural user interfaces for human-drone multi-modal interaction”. In: *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2016, pp. 1013–1022. DOI: 10.1109/ICUAS.2016.7502665.
- [TDJ11] L. Takayama, D. Dooley, and W. Ju. “Expressing thought: Improving robot readability with animation principles”. In: *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2011, pp. 69–76.
- [Tes98] Nikola Tesla. *Method of and apparatus for controlling mechanism of moving vessels or vehicles*. U.S. Patent US613809A. 1898.
- [TJ81] F. Thomas and O. Johnston. *The Illusion of Life: Disney Animation*. Disney Editions, 1981.

- [Tja+15] H. Tjaden, F.A. Stein, E. Schömer, and U. Schwanecke. “High-Speed and Robust Monocular Tracking”. In: *VISAPP '15*. May 2015, pp. 462–471.
- [Tja19] Henning Tjaden. “Robust Monocular Pose Estimation of Rigid 3D Objects in Real-Time”. <https://nbn-resolving.org/urn:nbn:de:hebis:77-diss-1000025478>. Retrieved: 2021-01-14. PhD Thesis. Germany: Johannes Gutenberg University Mainz, 2019.
- [TP09] Leila Takayama and Caroline Pantofaru. “Influences on proxemic behaviors in human-robot interaction”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 5495–5502. DOI: 10.1109/IR0S.2009.5354145.
- [TR05] Nikolas Trawny and Stergios Roumeliotis. *Indirect Kalman Filter for 3D Attitude Estimation*. Technical Report Number 2005-002, University of Minnesota, Dept. of Computer Science & Engineering. Mar. 2005.
- [Vis06] A. Visioli. *Practical PID Control*. Advances in Industrial Control. Springer, 2006.
- [VM98] Michiel J. Van Nieuwstadt and Richard M. Murray. “Real-time trajectory generation for differentially flat systems”. In: *International Journal of Robust and Nonlinear Control* 8.11 (1998), pp. 995–1020.
- [Wal+05a] M.L. Walters et al. “The influence of subjects’ personality traits on personal spatial zones in a human-robot interaction experiment”. In: *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005*. 2005, pp. 347–352. DOI: 10.1109/ROMAN.2005.1513803.
- [Wal+05b] Michael L. Walters et al. “Close encounters: spatial distances between people and a robot of mechanistic appearance”. In: *5th IEEE-RAS International Conference on Humanoid Robots, 2005*. Dec. 2005, pp. 450–455. DOI: 10.1109/ICHR.2005.1573608.
- [Wal+07] Michael L. Walters, Kerstin Dautenhahn, Sarah N. Woods, and Kheng Lee Koay. “Robotic etiquette: Results from user studies involving a fetch and carry task”. In: *2007 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2007, pp. 317–324. DOI: 10.1145/1228716.1228759.

- [Wal+18] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. “Communicating Robot Motion Intent with Augmented Reality”. In: *HRI '18*. Chicago, IL, USA: Association for Computing Machinery, 2018, pp. 316–324. DOI: 10.1145/3171221.3171253.
- [Wal21] Mike Wall. *Mars helicopter Ingenuity carries piece of Wright brothers' famous plane*. Mar. 24, 2021. URL: <https://www.space.com/mars-helicopter-ingenuity-wright-brothers-plane> (visited on 12/23/2023).
- [Wan+15] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy. “Evaluation of Pose Tracking Accuracy in the First and Second Generations of Microsoft Kinect”. In: *2015 International Conference on Healthcare Informatics*. 2015, pp. 380–389.
- [Wan+23] Ziming Wang, Ziyi Hu, Björn Rohles, Sara Ljungblad, Vincent Koenig, and Morten Fjeld. “The Effects of Natural Sounds and Proxemic Distances on the Perception of a Noisy Domestic Flying Robot”. In: *J. Hum.-Robot Interact.* 12.4 (Dec. 2023). DOI: 10.1145/3579859. URL: <https://doi.org/10.1145/3579859>.
- [Wei05] Eric Weisstein. *Music*. 2005. URL: <http://www.ericweisstein.com/encyclopedias/music/EqualTemperament.html> (visited on 03/28/2023).
- [Wei16] Benjamin Weißer. “Collaborative ORB-SLAM”. Master’s Thesis. Germany: RheinMain University of Applied Sciences, Wiesbaden, Oct. 2016.
- [Woj+19] A. Wojciechowska, J. Frey, S. Sass, R. Shafir, and J. R. Cauchard. “Collocated Human-Drone Interaction: Methodology and Approach Strategy”. In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. Mar. 2019, pp. 172–181. DOI: 10.1109/HRI.2019.8673127.
- [WS16] Stéfan van der Walt and Nathaniel Smith. *mpl colormaps*. 2016. URL: <https://bids.github.io/colormap/> (visited on 04/04/2023).
- [WT10] Sonia Waharte and Niki Trigoni. “Supporting Search and Rescue Operations with UAVs”. In: *2010 International Conference on Emerging Security Technologies*. 2010, pp. 142–147. DOI: 10.1109/EST.2010.31.

- [XIM19] XIMEA. *xiQ Technical Manual*. English. Version 1.35. XIMEA GmbH. Sept. 2019. Chap. 3.9.4.4.4, p. 50. URL: https://www.ximea.com/downloads/usb3/manuals/xiq_technical_manual.pdf (visited on 01/05/2024).
- [Yeh+17] Alexander Yeh et al. “Exploring Proxemics for Human-Drone Interaction”. In: *Proceedings of the 5th International Conference on Human Agent Interaction*. HAI ’17. Bielefeld, Germany, 2017, pp. 81–88. DOI: 10.1145/3125739.3125773.
- [Yil+16] Selma Yilmazyildiz, Robin Read, Tony Belpeame, and Werner Verhelst. “Review of Semantic-Free Utterances in Social Human–Robot Interaction”. In: *International Journal of Human–Computer Interaction* 32.1 (2016), pp. 63–85. DOI: 10.1080/10447318.2015.1093856.
- [Zag+16] Cristina Zaga, Roelof A.J. De Vries, Sem J. Spenklink, Khiet P. Truong, and Vanessa Evers. “Help-giving robot behaviors in child-robot games: Exploring Semantic Free Utterances”. In: *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2016, pp. 541–542. DOI: 10.1109/HRI.2016.7451846.
- [ZF23] Brian J. Zhang and Naomi T. Fitter. “Nonverbal Sound in Human-Robot Interaction: A Systematic Review”. In: *J. Hum.-Robot Interact.* 12.4 (Dec. 2023). DOI: 10.1145/3583743. URL: <https://doi.org/10.1145/3583743>.
- [Zha+16] Jian Zhang, Jianbo Hu, Juyuan Lian, Zongji Fan, Xuejun Ouyang, and Wanhui Ye. “Seeing the forest from drones: Testing the potential of lightweight drones as a tool for long-term forest monitoring”. In: *Biological Conservation* 198 (2016), pp. 60–69.
- [Zha00] Z. Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334. DOI: 10.1109/34.888718.
- [Zhe+13] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Åström, and Masatoshi Okutomi. “Revisiting the PnP Problem: A Fast, General and Optimal Solution”. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 2344–2351. DOI: 10.1109/ICCV.2013.291.

