ROBOTICS AND MECHATRONICS

HUMAN-AWARE MOTION CONTROL IN AERIAL HUMAN-ROBOT HANDOVERS

B.J. (Bastiaan) Bekooij

MSC ASSIGNMENT

Committee:
A. Franchi, Ph.D HDR
ir. A.N.M.G. Afifi
dr. ir. D. Reidsma

February, 2022

010RaM2022
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

In recent years research on Aerial Physical Interaction has been accelerated by both improvements in Aerial platform design and also optimization control knowledge. In robotics literature, the topic of Human-robot interaction is making rapid improvements that allow for complex collaborative actions such as handovers. This leads to the natural conception of doing aerial human-robot interaction. As part of the Aerial-CORE project, this thesis develops a method to do a perception-based pre-handover, the approach phase. In doing human-robot interaction, the safety of the human is paramount as well that the human should feel as comfortable as possible. With that purpose in mind, several objectives and constraints are formulated and implemented in a Non-linear Model Predictive Controller. Dedicated perception objectives and constraints make sure that the robot keeps the human in its view. To show intent to the human, a special legibility-inspired objective has been implemented. The controller is validated in simulation.

Acknowledgements

I would like to thank my daily supervisor Amr Afifi for his great supervision. I could always rely on you to guide me on the right course whenever I went off track. Your great understanding of control theory was of great help and also motivated me to delve deep into the literature. I wish you all the best with your PhD project.

To Antonio Franchi, I'm grateful for giving me the opportunity to work on this project under his supervision. The meetings we had always gave me plenty of fruit for thought with your insightful comments.

A few people I would like to thank for helping with some technical challenges are, Hemjyoti Das with his general knowledge of the GenoM system (your cheerfulness was always welcome too), Martin Jacquet for his expert knowledge on the NMPC software, Max Kivits for helping me get started on the NMPC framework and Davide Bicego for having a nice discussion on MPC tuning.

During Covid, it was sometimes challenging to have little social contact. I want to thank other RaM thesis students, especially Bernard Prakken and Cedric Damien Dsouza for the enjoyable coffee and lunch breaks and general support in doing a thesis.

To my family, especially my parents Jur and Suzan, I appreciate your everlasting support. Doing my Master's even after struggling with my earlier study years was only possible because I knew I could count on you. Thank you Aya, my grandmother, for the relaxing evenings and listening ear (if the battery wasn't empty).

Finally, I want to thank my fellow students and friends, with whom I could share the study struggle. The valuable walks and runs with Muriel, the good dinner talks with my housemates, Martijn, Monique, Jolisa, Mike, Lieke, Marleen, Noa and all those from the past, and the many evenings we spend playing volleyball and drinking beers with my teammates and many other members of Harambee.

Contents

1 Introduction

In all areas of life, it's become evident that technology is advancing at a steady pace. In office jobs, the computer is the fundamental piece of equipment while jobs without computers are often assisted by machinery. Technology has not only become more powerful or efficient but also smarter. It's the age of Artificial Intelligence (AI), the brains of robots. On a personal note, I learned about AI when Deepmind's AlphaGo earned fame by beating the top professional Go players for the first time ever. This moment gave many people the realization that AI could one day be smarter than humans. Although we're still not there, AI has accelerated robotics in many ways. Most notably in object recognition [1] and other image processing needs, but also in control [2].

Granting that the majority of robotic applications are in factories, many other areas have started to also apply smart robots. Such as social and monitoring robots in healthcare. These robotic co-workers are the future of daily human-robot interaction in households, supermarkets and other public places. Nowadays the COVID pandemic also introduced a need for safe and sterile robotic co-workers. In monotonous labor, these co-workers could assist humans in reducing workload and repetitiveness.

One area that is yet unoccupied by robotic co-workers is in aerial jobs. That is why a large scale European project has been set up to develop an Aerial Co-Working (ACW) system, Aerial-Core: *AERIAL COgnitive integrated multi-task Robotic system with Extended operation range and safety.* Within the RaM group of the University of Twente, the Aerial-Core efforts focus on creating fully actuated aerial robots that can work with, and in close proximity of humans. Here the design, mechanics, electronics, software and control system all have to come together.

Many advancements in recent years improved the physical capabilities of aerial robots. The next step is Aerial Physical Interaction (APhI) with humans, a combination of APhI and Human-Robot Interaction (HRI). One of the most fundamental interactions between humans is the handover, transferring an object by hand. This thesis studies the handover between an aerial robot and a human.

To communicate with the human and observe its position the robot is equipped with On-Board Vision. When doing the approach for the handover, perception is used to do human-aware path planning. This should ensure the safety of the human and improve human comfortability.

In complex tasks such as Human-Robot Interaction and flight, it's common to use optimization methods in the controller. This allows optimizing many different objectives when there is not a mathematical solution. The optimization is often bound by inequality constraints, such as actuator limitations.

1.1 Research Question

To advance the field of Aerial Human-Robot Interaction the main research question of this thesis is: **What are suitable Objectives and Constraints in Aerial Human-Robot Handovers?**

To answer this question several sub-questions are investigated:

1. How can human safety be ensured?
2. What parameters and objectives lead to comfortable behaviour for the human?
3. How can aerial robots show legible motion?
4. What kind of perception is necessary in human-robot handovers?
5. Which control method is needed to include all these objectives and constraints?

1.2 Contribution

The contributions of this thesis are a contextual analysis of Aerial Human-Robot Handovers, a methodology using an optimization control method and an engineering effort that implements the method.

In the literature review, relevant research is given to build the foundations of an aerial handover: Human-Robot handovers, human-aware path planning, robot perception, aerial platform design and aerial control methods.

A Non-linear Model Predictive Control scheme is proposed that takes into account the servoing, perception and human comfort. It does this by optimizing for several objectives and inequality constraints. The main control software and perception objectives are adapted from previous work [3] and expanded with a safety posited velocity constraint and legibility-inspired objective.[1]

The proposed method is implemented and validated in a software-in-the-loop approach using the GenoM3 framework. The resulting controller is able to approach the human in a comfortable and safe manner. The proposed legibility objective modifies the trajectory such that the robot's intentions are clearer.

1.3 Report Layout

The literature review (chapter 2) discusses what components are needed and how they come together. Next in chapter 3 method, the model for multi-rotor UAVs is given and the control method is elaborated on. With this in hand, the implementation is discussed in chapter 4. Using the implemented system, experiments are done to show the validity of the method. The results of which can be found in chapter 5 and are discussed in chapter 6. Finally, chapter 7 concludes the thesis and summarises answers to the research questions.

[1]The concept of legible motion isn't very well known, please refer to section 3.2.2 or [4] for an explanation.

2 Literature Review

This literature review is divided into three subjects. First, the general handover (section 2.1) will be discussed, both conceptually as in a Human-Robot Interaction (HRI) setting. The next subject is perception (section 2.2), which will go into robot perception, specifically of humans. 2D and 3D pose estimation of humans and gesture recognition are important for HRI. In Aerial Robotics the field of Aerial Physical Interaction is a hot topic and the main focus of the final section 2.3, Aerial Robots. This section will highlight the fully-actuated UAV design and different control methods. Special attention should be paid to the perception constraint NMPC.

2.1 Handover

A handover is the action of passing an object from one agent (giver) to another (receiver). Handovers happen all the time and are second nature to Human beings. Although humans can seemingly effortlessly do a handover, many things happen before and during the interaction.

2.1.1 General handover

In an object handover review for robotics [5], researchers found aspects "on the cognitive level (e.g., prediction, perception, motion planning, learning) and the physical level (e.g., motion, grasping, grip release) of the handover". Furthermore, the handover is described as being two phases, the pre-handover and physical handover. The pre-handover phase is initiated either with the receiver by object request or with the giver by task request [5]. After initiation of the object handover, humans use many types of communication to signal their intentions. Such as when theyre ready and where the handover will take place. The giver also anticipates how the receiver would like to grasp to object with regards to the intended use case of the object. For example how you hand over a knife with the handle towards the receiver. The pre-handover phase continues by reaching the object out to the receiver while tracking their readiness. In a continuous cycle, both agents update their motion plans based on visual and tactile feedback. The receiver on the other hand might predict the handover location and move their hand towards it. After the receiver makes contact with the object the physical handover starts. In the physical phase when both agents have contact with the object, the timing is crucial. When the object is released early the object will fall, but a late release will increase the energy that both participants exert. Here "the modulation of the grip force" [5] is required for a successful transfer. The end of the physical handover concludes the full handover, although some papers also include a retraction phase in the handover [6].

2.1.2 HRI

In the Human-Robot Interaction (HRI) field, a lot of research is dedicated to the handover [5, 7, 8, 9, 10, 6, 11, 12]. For the pre-handover phase research is interested in the optimal handover location for human ergonomics. In [7], a robot is able to estimate the personalized human kinematics to improve human ergonomics in a Human-robot handover when the human has a restricted range of motion. If the robot needs to move towards the human it's also possible to account for human mobility, for example, some people can reach out more easily and would prefer a shorter path [8].

When a Human-Robot handover is to take place it's important to show intent to do so. The robot's tasks might not be limited to only HRI, but also moving in the general area. One study on human-aware path planning argues that besides safe motion, a robot should also be legible and socially acceptable [9]. Legibility is a desired property of robots but not to be confused with predictability. Where legible motion expresses intent, predictable motion is expected and safe [4]. Predictability is goal-to-action, an observer can infer the action/motion from knowing the goal. On the other hand, legibility being action-to-goal shows the observer the goal with its action; intent.

From the perspective of the robot, it's also necessary to predict the handover location. The human might have already extended their arm or still be in motion. [10] developed a human motion predictor based on online Gaussian Process regression that accurately predicts the human motion with an RMSE less than 1cm. Another study used a Model Predictive Control (MPC)-based framework to optimize a collaborative objective [13].

The physical handover requires less communication but no less attention. We require the robot to be compliant and fluid while also reducing human effort. To achieve a fluid handover, [6] combines all handover phases (including retracting) and develops a bi-directional and human-inspired controller that treats the handover as a single continuous entity. This resulted in smoother handovers that require less human effort. The ergonomic preferences of human agents in a human-robot handover can be learned to enhance the human's comfortability [11]. This applies mostly to the way an object is grasped.

Human-Robot Interaction literature rarely tackles the combination of perception together with interaction control.

2.2 Robot Perception

When it comes to HRI, perception is one of the key obstacles to overcome, a robot does not simply "see". First of all, it should recognize that there is a human present in its close environment. The most common type of perception is using cameras, simple optical cameras, but also infrared or depth cameras could be used. Most studies in HRI use a system called Optitrack to track the location of a small reflective sphere that's attached to the human [14, 6, 15, 10]. However, in real life situations, it might not be reasonable to expect humans to be instrumented with sensors. Note also that these motion tracking systems are external to the robot, while in the case of aerial robots, it's not feasible to install systems around the interaction space. Instead, on-board perception is needed.

Computer vision is needed to distinguish and locate humans when only camera images are available. With recent developments in Machine Learning and Deep Learning, State-of-the-Art Neural Networks are being developed that can estimate the pose (also called skeleton) of a human, see figure 2.1. Arguably the best Open-source pose estimation software is OpenPose [16], a real-time algorithm that can find the human pose skeleton in a fraction of a second. Note however that this skeleton is in the image plane and does not convey depth information, nor give the 3D position of key points.

There are also methods that do 3D Pose estimation using a 3D sensor [17, 18, 19]. In [17] a Microsoft Kinect is used to estimate the upper body pose of people sitting behind a desk. However, their system is sensitive to occlusions. Similarly [18] uses synthetic data to train a neural network, but also only on the upper body. In [19] a full pose is estimated from depth images by comparing a point cloud to predefined poses.

As discussed earlier communication is an important part of any handover. Although verbal communication is maybe the most common way to start a handover from human to human, for robots this would require accurate speech recognition and understanding. Moreover, in the case of Aerial Robots speech recognition is hindered by the loud noise coming from the propellers. Therefore gesture recognition is a more suitable method. Gestures are commonly done with the arms and hands, like pointing and waving. The hands are also able to make a large number of gestures by just moving the fingers and orientation of the hand. However, hand signals might be hard to distinguish from a distance. In aerial applications, there is a lot of space and the human and robot might not be close when they need to communicate. Instead, full body gestures are easier to perceive from a distance. Researchers from the Aristotle University of Thessaloniki developed a Convolutional Neural Network (CNN) approach

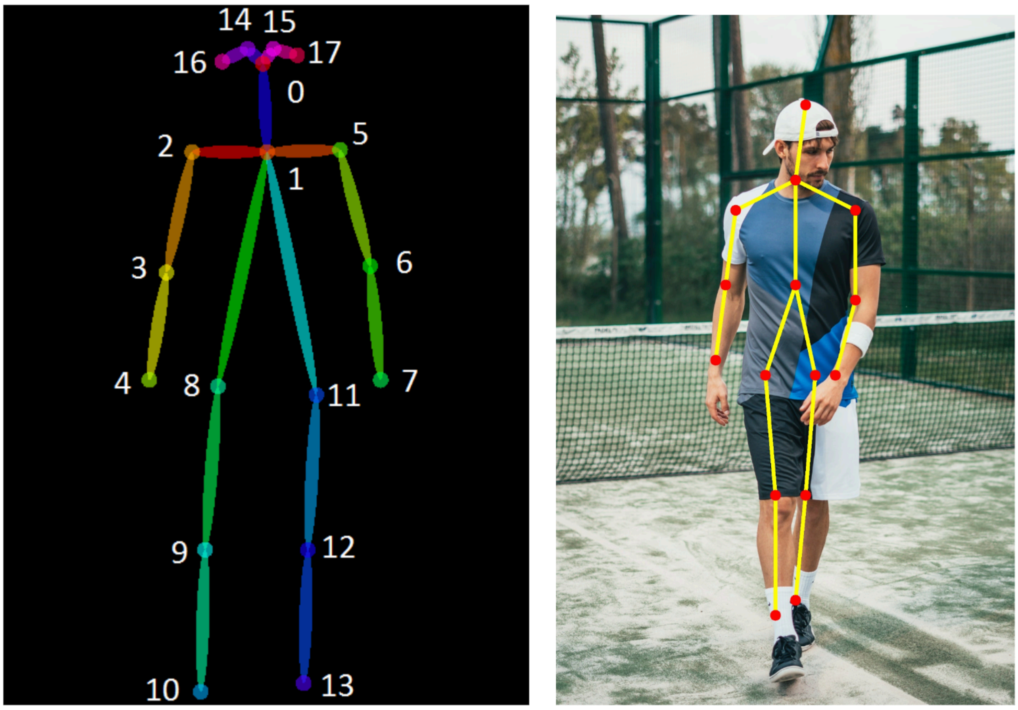


Figure 2.1: Human Pose Estimation example. Left: The 'skeleton' of keypoints that correlate to the human body. Right: An estimated skeleton overlayed on the human.

that recognizes gestures in Human-Robot Interaction [20, 21, 22]. Using a 2D skeleton, as described in earlier paragraphs, they can distinguish different gestures from a variety of angles. The gestures came from the AUTH UAV Gesture dataset [21] (6 gestures) and the UAV-gesture dataset [23] (13 gestures).

2.3 Aerial Robotics

Since the subject of the thesis is the interaction of an aerial robot with a human, it's necessary to discuss aerial platform design and control methods for aerial robots.

2.3.1 Aerial platform design

In the past decade, Unmanned Aerial Vehicles (UAVs) have seen rapid improvements in capabilities. Although quadrotors are still the most common drone for private users, UAVs with six or eight propellers are more often used in commercial settings like professional photography. A recent trend in research is hexarotors with non-collinear tilted propellers [24, 25, 26, 27, 28]. This trend is mainly powered by the desire to do Aerial Physical Interaction (APhI). With non-collinear thrust vectors, the hovering efficiency is decreased somewhat, however, it provides one large benefit: Full 6D actuation, covering the three translational dimensions and three rotational dimensions. This means that the UAV is able to accelerate laterally without changing orientation, something that is not possible for regular quadrotors and underactuated hexarotors.

Fully-actuated UAVs are highly valuable in the field of Aerial Physical Interaction (APhI) because it offers the full range of motions and forces. Nevertheless, APhI has also been studied for quadrotors [30, 31]. The flying end-effector paradigm [29] uses this 6D actuation concept "to exert a full-wrench (force and torque independently) with a rigidly attached end-effector".

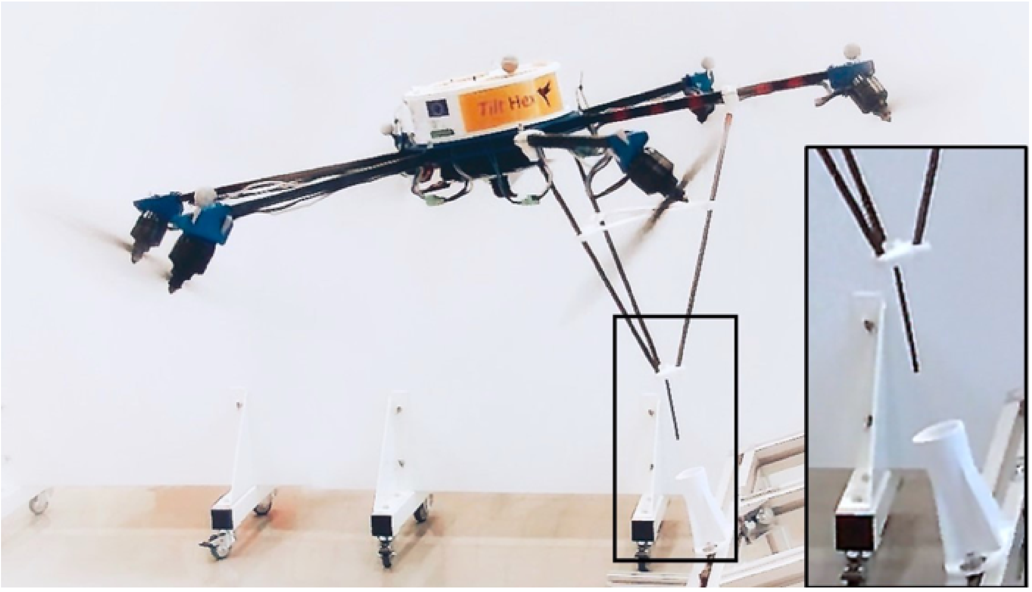


Figure 2.2: Typical hexarotor with tilted propeller design. [29]

2.3.2 Control methods

In APhI literature, a range of control methods has been used. Commonly there are the impedance/admittance control methods. A method that includes perception is Visual Servoing (VS), which is used by [28]. More advanced are the Optimization methods, such as Quadratic Programming (QP) [32] and Model Predictive Control (MPC) [3, 33, 34].

When dealing with image-based information, Visual Servoing is particularly interesting because it couples visual input to control output. Two types of visual servoing exist, Image Based Visual Servoing (IBVS) and Position Based Visual Servoing (PBVS) [35, 36]. In IBVS points in the image plane are used to directly control the motion, while in PBVS (a) point(s) in the image is first converted to a position in 3D space, which in turn is a reference for the controller. Visual servoing has also been used in Aerial Robots [28], which combines the flying end-effector paradigm with Hybrid Visual Servoing.

In recent years there has been an increasing interest in using optimization-based methods for APhI in order to take into account system limitations, such as minimum and maximum propeller speeds. A basic method is quadratic programming (QP), which can optimize a multivariate quadratic cost function. In aerial robotics, QP control has been used cascaded with an admittance filter to achieve compliant behaviour [32].

Another commonly used optimization method is Model Predictive Control (MPC). MPC optimizes a cost function over a finite horizon under certain inequality constraints. In [3] Non-linear MPC is applied to Multi-Rotor Aerial Vehicles with generic designs. Their work shows that prevailing assumptions, such as linearizing the dynamic model, decoupled translation and rotation control, low-level reactive stabilization trackers and unconstrained optimization resolution are not needed [3]. Moreover, the controller uses an enhanced actuator model that takes into account the limitations of propeller torque.

MPC can also be used in tandem with other control methods such as impedance/admittance control and hybrid pose force tracking [34]. Three different NMPC controllers are proposed in [34]: Cascaded control, impedance control and hybrid control. The author shows that all three controllers are viable for APhI and each controller has its strength depending on the task.

In order to combine Human Pose estimation, visual servoing and other perception tasks, it's necessary to keep the target (in our case the human) in the field of view of the camera. One could of course install multiple cameras, but besides the downside of extra mass, the perception would need to combine all cameras, making the system more complex. Instead, it's more logical to constrain the motion of the UAV to track the target, as is done in [33]. They implemented objectives and constraints for a Non-linear MPC controller to track a target while following a trajectory. The controller is able to balance tracking and dynamics, allowing a position error when otherwise the target(s) would be lost. To optimize cooperative target tracking, this work has been extended with a Kalman Filter in [37, 38].

The aerial handover is an interdisciplinary problem that combines elements from motion planning, perception and control. In motion panning the approach, ergonomy and legibility are encapsulated. Perception is needed to keep the human visible and determine the handover location. Finally, control is needed to servo towards or alongside a human. As seen in [33, 37, 38], visibility of a generic target is considered by formulating an NMPC Optimal Control Problem (OCP). Which is solved in realtime for a tilted hexarotor. This work exploits the previous framework to tackle the approach phase of the handover.

3 Method

In this chapter the methodology to tackle the pre-handover phase is presented. First, the generic tilted hexarotor will be modelled, then the Optimal Control Problem (OCP) is formulated including the objectives and constraints that are related to perception and human-aware control.

In the previous chapter related work has been reviewed and the motivation for the following methodology has been presented. In this thesis, the NMPC control method will be applied to do the pre-handover phase. That is, to approach the human in a safe, comfortable and legible way. The main body of the work is focused on designing the objectives and constraints that the NMPC has to optimize. Nonetheless, the thesis will show how this system is connected to perception, Human pose estimation, gesture recognition and the physical handover.

To introduce the method, a user scenario is sketched:
A mechanic is working at a high altitude on high-voltage transmission lines. He notices that a certain part needs to be replaced, but didn't account for it from the ground. He would have to travel down to the ground, pick up the part and climb back up, at least if he didn't have the possibility to have the aerial robot fetch it for him. The mechanic calls his colleague on the ground that starts op the aerial robot and attaches the needed part in the end-effector. The robot automatically flies to a position close to the mechanic and waits for a gesture. When the mechanic sees the aerial robot he gestures him to come closer. Next, the mechanic signals the robot to do the handover of the part with his left hand because there is more space on that side, and he's left-handed too. The aerial robot moves towards the hand in a calm manner and the mechanic is able to reach out and grasp the part. The robot slowly transfers the load to the mechanic and retracts as soon as the handover is finished. The mechanic gives a final gesture that the robot can leave and the mission is successful.

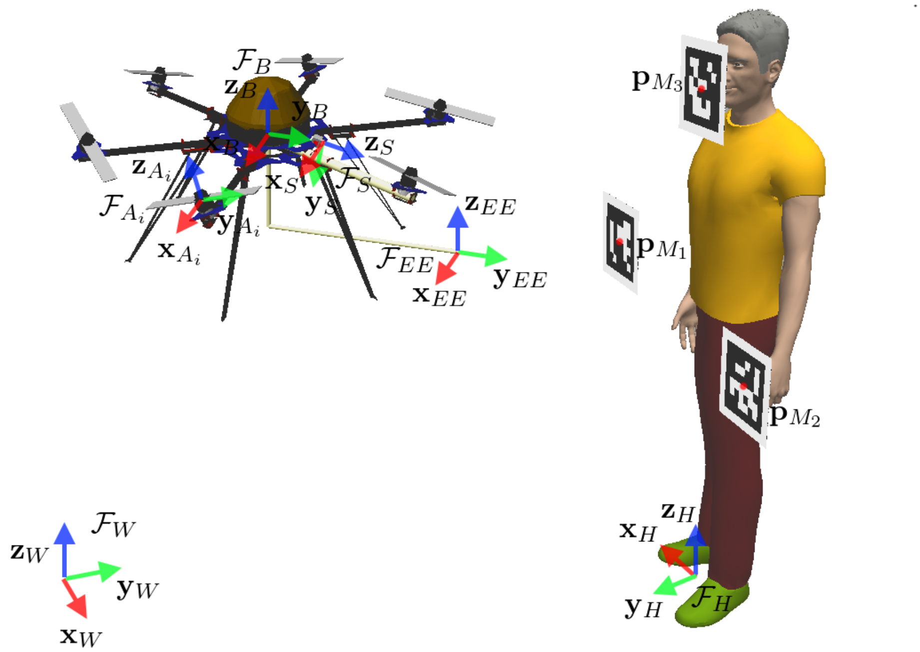


Figure 3.1: Overview of Aerial Robot and environment including reference frames (see table 3.1) and points.

3.1 Modelling

In this section, the model of a generic tilted multi-rotor UAV is formulated. In figure 3.1 the relevant frames are visualized. First, we have the inertial world frame $\mathcal{F}_W = O_W\{\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$. On the UAV there are four relevant types of frames: The body frame $\mathcal{F}_B = O_B\{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$, the end-effector frame $\mathcal{F}_{EE} = O_{EE}\{\mathbf{x}_{EE}, \mathbf{y}_{EE}, \mathbf{z}_{EE}\}$, the sensor/camera frame $\mathcal{F}_S = O_S\{\mathbf{x}_S, \mathbf{y}_S, \mathbf{z}_S\}$ and the actuator frames $\mathcal{F}_{A_i} = O_{A_i}\{\mathbf{x}_{A_i}, \mathbf{y}_{A_i}, \mathbf{z}_{A_i}\}$, for each i-th actuator. In the actuator frames, $\mathbf{z}_{A_i}$ is aligned with the thrust vector.

The human has its reference frame $\mathcal{F}_H = O_H\{\mathbf{x}_H, \mathbf{y}_H, \mathbf{z}_H\}$ at his feet, furthermore, there are three relevant feature locations $\mathbf{p}_{M_j}$. These features are located by a pose estimation algorithm and depth sensor, but in this thesis, Aruco-tags are used in simulation as a stand-in.

To model the dynamics a few symbols are used. The translational dynamics use $\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}} \in \mathbb{R}^3$ to represent the position, velocity and acceleration of the aerial robot, thus $\mathcal{F}_B$ with respect to $\mathcal{F}_W$. The rotation from $\mathcal{F}_B$ to $\mathcal{F}_W$ is given by $\mathbf{R}_B^W \in SO(3)$ or $\mathbf{R}$ for short. Another way to notate rotations is using quaternions $\mathbf{q}$, which will be used in the controller. Next, $\omega \in \mathbb{R}^3$ is the angular velocity and $\dot{\omega} \in \mathbb{R}^3$ the angular acceleration, both expressed in $\mathcal{F}_B$. In table 3.1 these and other symbols are summarized.

Table 3.1: Symbols

Definition	Symbol
Inertial world frame	$\mathcal{F}_W$
Body frame	$\mathcal{F}_B$
End Effector frame	$\mathcal{F}_{EE}$
Sensor frame	$\mathcal{F}_S$
Actuator frame (i-th)	$\mathcal{F}_{A_i}$
Position, velocity and acceleration of O_B in $\mathcal{F}_W$	$\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}}$
Rotation matrix of $\mathcal{F}_B$ w.r.t $\mathcal{F}_W$	$\mathbf{R}$
Quaternion representing rotation of $\mathcal{F}_B$ w.r.t $\mathcal{F}_W$	$\mathbf{q}$
Angular velocity of $\mathcal{F}_B$ w.r.t $\mathcal{F}_W$ expressed in $\mathcal{F}_B$	$\boldsymbol{\omega}$
Angular acceleration of $\mathcal{F}_B$ w.r.t $\mathcal{F}_W$ expressed in $\mathcal{F}_B$	$\dot{\boldsymbol{\omega}}$
Propeller thrusts	$\boldsymbol{\gamma}$
Actuator torques	$\dot{\boldsymbol{\gamma}}$
Force and moment allocation matrix	$\mathbf{G}_f, \mathbf{G}_\tau$

Rotations are commonly displayed in Euler-angles because it's straightforward to understand and use. However, Euler angles do suffer from singularities because the basis vectors are not linearly independent. That is the reason that quaternions are used inside the controller and OCP. The results however will be displayed using Euler-angles for easier interpretation. The propeller thrust is noted with $\boldsymbol{\gamma} \in \mathbb{R}^6$ and the actuator torques with $\dot{\boldsymbol{\gamma}} \in \mathbb{R}^6$.

As discussed above, the translational dynamics are in the world frame, while rotations are in the body frame. This results in

$$\begin{bmatrix} m\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{J} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} -mg\mathbf{e}_3 \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{q} \otimes \mathbf{G}_f \boldsymbol{\gamma} \otimes \mathbf{q}^* \\ \mathbf{G}_\tau \boldsymbol{\gamma} \end{bmatrix} \tag{3.1}$$

where m and J are the mass and inertia of the UAV respectively and g is the gravitational acceleration. The last term includes the forces and torques on the body. $\mathbf{G}_f, \mathbf{G}_\tau \in \mathbb{R}^{3 \times n}$ are the force and moment allocation matrices [39] and $\mathbf{q}^*$ is the conjugate quaternion of $\mathbf{q}$. The operator $\otimes$ is used for the Hamiltonian product of two quaternions.

3.2 Control

In the literature review, it was noted that Model Predictive Control (MPC) gained popularity in the Aerial Physical Interaction field for its ability to optimize multiple objectives and constraints. This led to the substantiated choice to use MPC in the handover task. In this section, the Optimal Control Problem will be formulated.

3.2.1 Motivation of Objectives and Constraints

The objectives are soft constraints that the MPC attempts to keep at a certain value. However, objectives can conflict with each other or with hard constraints. In that case, the MPC will balance the objectives and stay within the constraints. Balancing objectives is based on tuned weights. To take inventory of the necessary objectives we look at the desires of the system. From the literature review (chapter 2), it's desired that the UAV:

1. Is safe with regards to humans
2. Has comfortable behaviour
3. Is legible (see subsection 2.1.2)
4. Holds it's reference position and orientation
5. Keeps human in sight

From the desires concrete objectives can be formulated:

- Zero position and orientation error (4)
- Zero velocity (1, 2)
- Keep human features centered in camera FoV (5)
- Have a legible trajectory towards goal (3)
- Stay at human sight line height when traveling (2)

Secondly there are some requirements (needs) that always need to be maintained (constraints):

- Maximum velocity (1)
- Human features are kept within FoV boundaries (5)

A simple description of an OCP is: Minimize the state error with respect to a reference state.

$$\min \|\mathbf{x} - \mathbf{x}_r\| \tag{3.2}$$

where $\mathbf{x}$ is the state and $\mathbf{x_r}$ the reference. This state can be expanded with other objectives that are not in the state itself, but are desired to be controlled. Section 3.2.5 goes through this process, but first the objectives are formulated as measurable values that have reference values.

3.2.2 Legibility

From [4]: "We thus formalize *legible* motion as motion that enables an observer to *confidently* infer the *correct* goal configuration G after observing only a snippet of the trajectory, $\xi_{S\to Q}$, from the start of the trajectory S to the configuration at a time t, $Q = \xi(t)$: $\mathcal{I}_L(\xi_{S\to Q}) = G$".

Here the authors use an inference function as a map between trajectories Ξ and goals [4]:

$$\mathcal{I}_L : \Xi \to \mathcal{G} \tag{3.3}$$

To finish the formalism, "The *quicker* this happens (i.e. the smaller t is), the more legible the trajectory is." [4].

Next it's possible to model $\mathcal{I}_L$ as the goal candidate with the highest probability [4]:

$$\mathcal{I}_L(\xi_{S\to Q}) = \arg\max_{G\in\mathcal{G}} P(G|\xi_{S\to Q}) \tag{3.4}$$

Using Bayes' Rule, this probability can be calculated when the probability of a trajectory is known. Since we're talking about legibility from the human's perspective, a model is needed of the probability of a trajectory ξ from the point of view of the human observer [4]. If a cost function can be assigned to trajectories, the observer expects the trajectory to have a minimum cost, or at least close to it: $P(\xi) \propto \exp\big(-C(\xi)\big)$ [4]. Using methods explained in [4], the probability $P(G|\xi_{S\to Q})$ is then further derived to be:

$$P(G|\xi_{S\to Q}) \propto \frac{\exp\Big(-C(\xi_{S\to Q}) - C(\xi^*_{Q\to G})\Big)}{\exp(-C(\xi^*_{S\to G}))} P(G) \tag{3.5}$$

Here the star * denotes the optimal trajectory, the one with minimal cost C.

Using the formalism given before, a legibility score is defined that follows the probability of the goal being the actual goal G^* while giving more weight to the start of the trajectory using $f(t)$ (e.g. $f(t) = T - t$, where T is the duration of the full trajectory):

$$\text{legibility}(\xi) = \frac{\int P(G^*|\xi_{S\to\xi(t)})f(t)dt}{\int f(t)dt} \tag{3.6}$$

However maximising this objective might lead to higher and higher costs, thus a regularizer is introduced [4]:

$$L(\xi) = \text{legibility}(\xi) - \lambda C(\xi) \tag{3.7}$$

This equation concludes the modelling of the legibility as adapted from [4]. Given this theory the question arises how this could be implemented in the human-aware controller. The legibility formalism is depended on the unknown cost function C that is human dependent. A straightforward approximation could be given by

$$C_{approx} = \sum_t \big\|\xi(t+1) - \xi(t)\big\|^2 \tag{3.8}$$

that is, the sum squared velocity. In a follow up paper [40] the cost function is further refined as

$$C_{approx}(\xi) = \frac{1}{2}\int \xi'(t)^2 dt \tag{3.9}$$

Although this cost function gives us access to a method to measure the legibility of a trajectory, it's not given how to optimize a trajectory from scratch. In a follow-up paper, [40] goes in-depth to optimize the trajectory using a gradient descent method. This is however not in line with this thesis since we want the MPC to generate the trajectory based on objectives and constraints in real-time. This leads to the proposed method, a hand-designed objective that increases legibility. The resulting trajectories should have increased legibility, but are not the most legible because they're generated in-flight while having to also optimize other objectives and work within the constraints.

In figure 3.2 the legibility-inspired objective is visualized. The objective can be described as "Keep the human on the right"[1] from the UAVs point of view. The slope is a measure of how much to turn and the y-intercept modifies the end state.

[1]Or left when the right hand is targeted

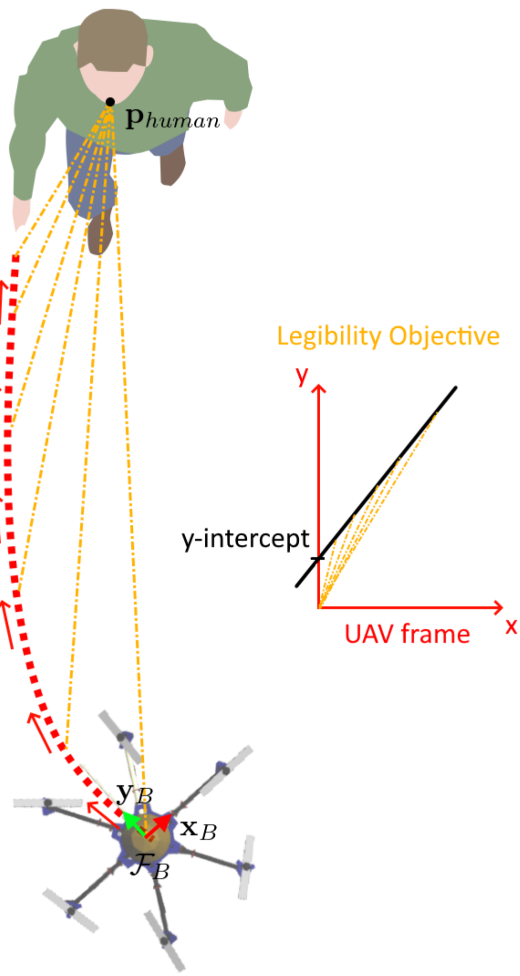


Figure 3.2: A desired trajectory is shown with the red dotted line. The yellow line represents the location of the human at varying times on the trajectory. The graph shows the same yellow line but in the UAV frame (scaled down).

The ability to tune these parameters gives the engineer the option to modify the behaviour of the controller regarding legible motion. It's possible to fly the trajectory and measure the legibility using equation 3.7, which will assist in tuning the parameters.

The equation of the visualized function is: ${}^{B}\mathbf{p}_{human,y} = a * {}^{B}\mathbf{p}_{human,x} + c$
Where a is the slope in figure 3.2 and c the y-intercept.

Making one side a constant: $a * {}^{B}\mathbf{p}_{human,x} - {}^{B}\mathbf{p}_{human,y} + c = 0$

Above equations are in body frame $\mathcal{F}_B$, but the MPC receives the human location as a feature M in the camera frame $\mathcal{F}_S$. Therefor we give the legibility objective η in the camera frame.

$$\eta = (a * x_M - z_M + c) * \|\mathbf{p} - \mathbf{p}_r\| \tag{3.10}$$

Here the feature ${}^{S}\mathbf{p}_M$ is in the camera frame, thus z is the depth and not to be confused with height. Parameters [a, c], the slope and y-intercept in figure 3.2, are used to tune the behaviour. The norm of the position error is used to attenuate the objective close to the reference position. It should be noted that using the position error introduces a second proportional gain in the system, this should be taken into account when tuning the weights of the system. The reference of this hand-designed objective should be zero in accordance with the earlier derivation.

3.2.3 Perception

The perception objectives are adapted from previous work with NMPC on the GenoM framework [33]. To formulate the perception objectives some symbols need to be defined. A feature M is tracked using a sensor S. The position of feature M: ${}^{S}\mathbf{p}_M = [x_M \; y_M \; z_M]^\intercal$ described in $\mathcal{F}_s$.

The tracking objective is to minimize the angle β between the tracked feature ${}^{S}\mathbf{p}_M$ and the principle camera axis. For better evaluation efficiency, the cosine of β (denoted $\mathrm{c}\beta$) is maximized instead [33].

3.2.4 Constraints

Besides the objectives there are also several constraints that the MPC has to adhere to. First there is the physical actuator constraints $\underline{\gamma}$ and $\overline{\gamma}$ that are determined by the manufacturers specifications. Furthermore, the enhanced actuator model from [3] is used to limit the input $\mathbf{u}$ by $\underline{\dot{\gamma}}$ and $\overline{\dot{\gamma}}$.

Next, there are the perception constraints (eq. 3.11) which make sure that the features stay in the field of view of the sensor. Otherwise, the perception objectives can't be optimized. One could argue that there are scenarios where it is okay to lose track if another objective is prioritized (such as safety), but this is left as future work.

$$\left|x_M/z_M\right| \leq \tan\,\alpha_h \tag{3.11}$$
$$\left|y_M/z_M\right| \leq \tan\,\alpha_v \tag{3.12}$$

Here α_h and α_v denote the horizontal and vertical angle of the Field of View (FoV) of the camera.

The final constraint is the safety constraint, commonly needed in Human-Robot Interaction. The controller should limit the velocity to a certain value. This thesis will show the constraint in effect, but the actual value of this constraint will depend on the environment, local regulations and supervisor desires. It's formulated as:

$$\|\dot{\mathbf{p}}\| \leq \overline{v} \tag{3.13}$$

where $\overline{v}$ is the upper boundary on the velocity. No lower boundary is needed since a negative velocity is impossible.

3.2.5 Formulating the OCP

The optimal control problem can be divided into control of the state $\mathbf{x}$, the input $\mathbf{u}$, and custom objectives. Furthermore, there are the physical actuator constraints, perception constraints and safety constraints.

The state vector and input vector are defined as:

$$\mathbf{x} := [\mathbf{p}^\intercal \; \mathbf{q}^\intercal \; \dot{\mathbf{p}}^\intercal \; \boldsymbol{\omega}^\intercal \; \gamma^\intercal] \tag{3.14}$$

$$\mathbf{u} := \dot{\gamma} \tag{3.15}$$

Now that all parts are defined, the OCP can be formulated. First the output map $\mathbf{y}$ and its reference $\mathbf{y}_r$ are the following:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{p}_M) = [\mathbf{p}^\intercal \; \mathbf{q}^\intercal \; \dot{\mathbf{p}}^\intercal \; \boldsymbol{\omega}^\intercal \; \ddot{\mathbf{p}}^\intercal \; \dot{\boldsymbol{\omega}}^\intercal \; \mathrm{c}\beta \; \dot{\mathrm{c}\beta} \; \eta]^\intercal \tag{3.16a}$$
$$\mathbf{y}_r = [\mathbf{p}_r^\intercal \; \mathbf{q}_r^\intercal \; \dot{\mathbf{p}}_\mathbf{r}{}^\intercal \; \boldsymbol{\omega}_r^\intercal \; \ddot{\mathbf{p}}_r^\intercal \; \dot{\boldsymbol{\omega}}_r^\intercal \; 1 \; 0 \; 0]^\intercal \tag{3.16b}$$

Next, the output map and its reference are discretized and used in the optimization function:

$$\min_{\substack{\mathbf{x}_0 \ldots \mathbf{x}_N \\ \mathbf{u}_0 \ldots \mathbf{u}_{N-1} \\ \mathbf{p}_{M_0} \cdots \mathbf{p}_{M_N}}} \sum_{k=0}^{N} \left\|\mathbf{y}_k - \mathbf{y}_{r,k}\right\|_{\mathbf{Q}}^2 \tag{3.17a}$$

$$\text{s.t. } \mathbf{x}_0 = \mathbf{x}(t) \tag{3.17b}$$

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \qquad {\scriptstyle k\in\{0,N-1\}} \tag{3.17c}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_{M,k}), \qquad {\scriptstyle k\in\{0,N\}} \tag{3.17d}$$

$$\underline{\gamma} \leq \gamma_k \leq \overline{\gamma}, \qquad {\scriptstyle k\in\{0,N\}} \tag{3.17e}$$

$$\underline{\dot{\gamma}} \leq \mathbf{u}_k \leq \overline{\dot{\gamma}}, \qquad {\scriptstyle k\in\{0,N-1\}} \tag{3.17f}$$

$$\left| x_M / z_M \right|_k \leq \tan \alpha_h, \qquad {\scriptstyle k\in\{0,N\}} \tag{3.17g}$$

$$\left| y_M / z_M \right|_k \leq \tan \alpha_v, \qquad {\scriptstyle k\in\{0,N\}} \tag{3.17h}$$

$$\|\dot{\mathbf{p}}\| \leq \overline{v}, \qquad {\scriptstyle k\in\{0,N\}} \tag{3.17i}$$

This function is optimized over the so-called receding horizon using a solver. Many types of solvers exist, in this thesis, the implemented framework uses a QP-Solver. Where QP stands for Quadratic Programming, a mathematical method that's used for solving an OCP.

4 Implementation

In this chapter, the controller will be implemented as software and connected to a simulation environment. Thus this simulation will be in the category of "software in the loop", where the real-time software that also could run on-board an aerial robot, is connected to a simulation. Thus treating the simulated aerial robot as if it was physical.

Simulations are necessary for human-robot interaction research to show the validity of the control method and ensure its safety. Using the simulation, the goal is to show desired behaviour for Human-Robot Interaction and show compelling legible movement in a pre-handover.

In this thesis, Gazebo is used as the physics simulator. Gazebo [41, 42] is a free robot simulation platform with a big community in research.

In simulation, it's very complex to use the same image processing techniques as in real life since rendering simulated models give a very different image. it's also overcomplicated to have accurate human models since image processing can easily be decoupled from simulation tests. Instead, it's better to use markers in simulation to substitute for human pose estimation. Aruco tags are commonly used to mark points in simulated and real systems. If the tag size is known, it's possible to accurately estimate its position and orientation relative to the camera.

For the experiments, a so-called world is created that contains the models needed for the simulation. In figure 3.1 this world can be seen. Two models can be seen, a Human (with tags) and the Fiberthex.

A software framework called GenoM3 [43] is used to implement the MPC. GenoM3, an acronym for *Generation of Modules*, allows creating efficient software from a descriptive language. This is useful for on-board controllers where speed and low energy consumption are required. In table 4.1 the GenoM modules that are used in this thesis are listed.

Table 4.1: GenoM modules

GenoM Module	Description
mrsim	Multi-Robot SIMulator, a transparent interface that emulates real aerial vehicles
pom	UKF-based state estimator
optitrack	Connects motion capture data from Optitrack system to other GenoM modules
rotorcraft	Low-level hardware controller
maneuver	Trajectory generation
nhfc	Near-Hovering Flight Controller, used for take-off
uavmpc	Core module that controls the UAV using MPC
camgazebo	Links gazebo cameras to other modules
camviz	Visualization of camera feed(s)
arucotag	Generation of Aruco Tags and estimation of tag location using Kalman-Filters
tagcontrol	Module that controls Gazebo models (incl. tags) from Matlab

4.1 Setup of machine

This section will highlight some technical details of the implementation.

First of all the operating system being used is Ubuntu 18.04 'Bionic Beaver' [1]. This operating system has been installed on a Virtual Machine (VM) for convenience purposes (VMware Workstation).

[1] https://releases.ubuntu.com/bionic/

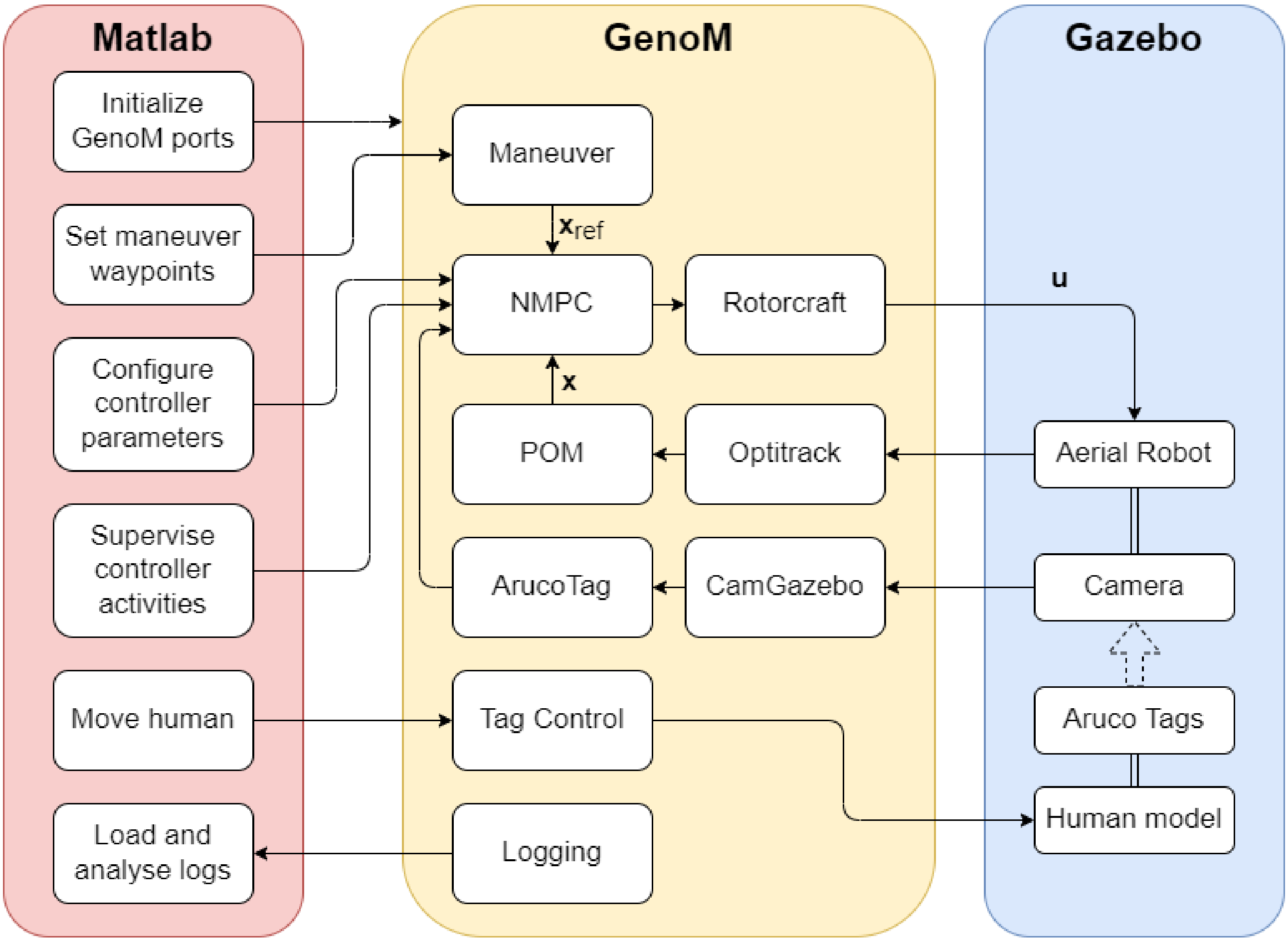


Figure 4.1: System architecture showing the connection between Matlab, GenoM components and Gazebo physics simulation.

Settings of the VM can be reviewed in table 4.2. Note that it is very crucial to turn on 'Accelerate 3D graphics', but depending on the PC some troubleshooting might be needed to get Gazebo to work [2].

Setting	Value
Disk size	60 GB
Amount of Cores	6
Memory	15 GB
Accelerate 3D graphics	ON

Table 4.2: Setting used in VMware Workstation.

The system that's used for simulation is very different from an On-Board computer. For running the framework used in this thesis it's recommended to use a powerful, but efficient, embedded system that contains graphics computing. Graphics computation is needed to do the necessary computer vision regarding human pose estimation. An analysis of different embedded systems has been done (see appendix A) and concludes that a *NVIDIA Jetson Xavier NX* has the best balance between efficiency and computing power that conforms to the requirements.

[2] https://robocademy.com/2020/05/02/solved-opengl-issues-with-gazebo-and-vmware/

4.2 Coding

The UAVMPC framework from [33] is the basis of this thesis, but several modifications are needed to implement the new objectives and constraints.

In the UAVMPC module there are three files that needed foremost expansion:

1. uavmpc.gen
2. control_codels.cpp
3. model_fiberthex.py

In *uavmpc.gen* activities are added that control the approach phase of handover. Activities are a GenoM concept that involve several functions and are intended to run in parallel with other activities. Next to those functions, some variables are defined to set the approach locations relative to the Human via Matlab.

The functions of this activity are implemented in *control_codels.cpp*. The approach is implemented by reading Feature position (Aruco tag or human pose estimation) and subsequently setting the reference position $\mathbf{p}_r$ relative to a feature position $\mathbf{p}_{M_i}$. Two features are considered to use, one centered on the human body and one of the hand (left or right). Although intuitively the hand is used to do the handover, and would thus seem best, the hand might be involved in other human actions and move much faster. Also when the hand is next to the body, the UAV shouldn't servo towards the hand, but a distance from it and when the hand reaches out to do the handover this distance should somehow decrease. That all is to say, the hand is not a robust feature to use as a reference. The method instead uses the central core of the human as a reference and the supervisor can modify the vector relative to the human as an approach location.

Finally, the objectives and constraints are added in the model file *model_fiberthex.py*. This python file describes the UAV and sets the objectives/constraints. Adding the velocity constraint was straightforward by adding a variable to the general constraints. The legibility objective, on the other hand, required some modifications to the cost function evaluation.

The above-mentioned files will be uploaded and can be requested from the author.

5 Results

In this chapter, the simulation results will be shown and discussed. Two categories of experiments have been done. The first category is free flight in human presence, here the base stability of the platform will be shown. The velocity constraint and perception constraints are turned on, referred to as tracking mode. Secondly, there are the results of the Approach Phase. Both the approach to the interaction space of the human and the approach to the Handover Location are discussed. For approaching the Human, the legibility objective will be demonstrated. In figure 5.1 the process is illustrated with the steps that are followed.[1] Lastly, in section 5.3 the simulated human will be moved around to show the sensitivity of the MPC controller.

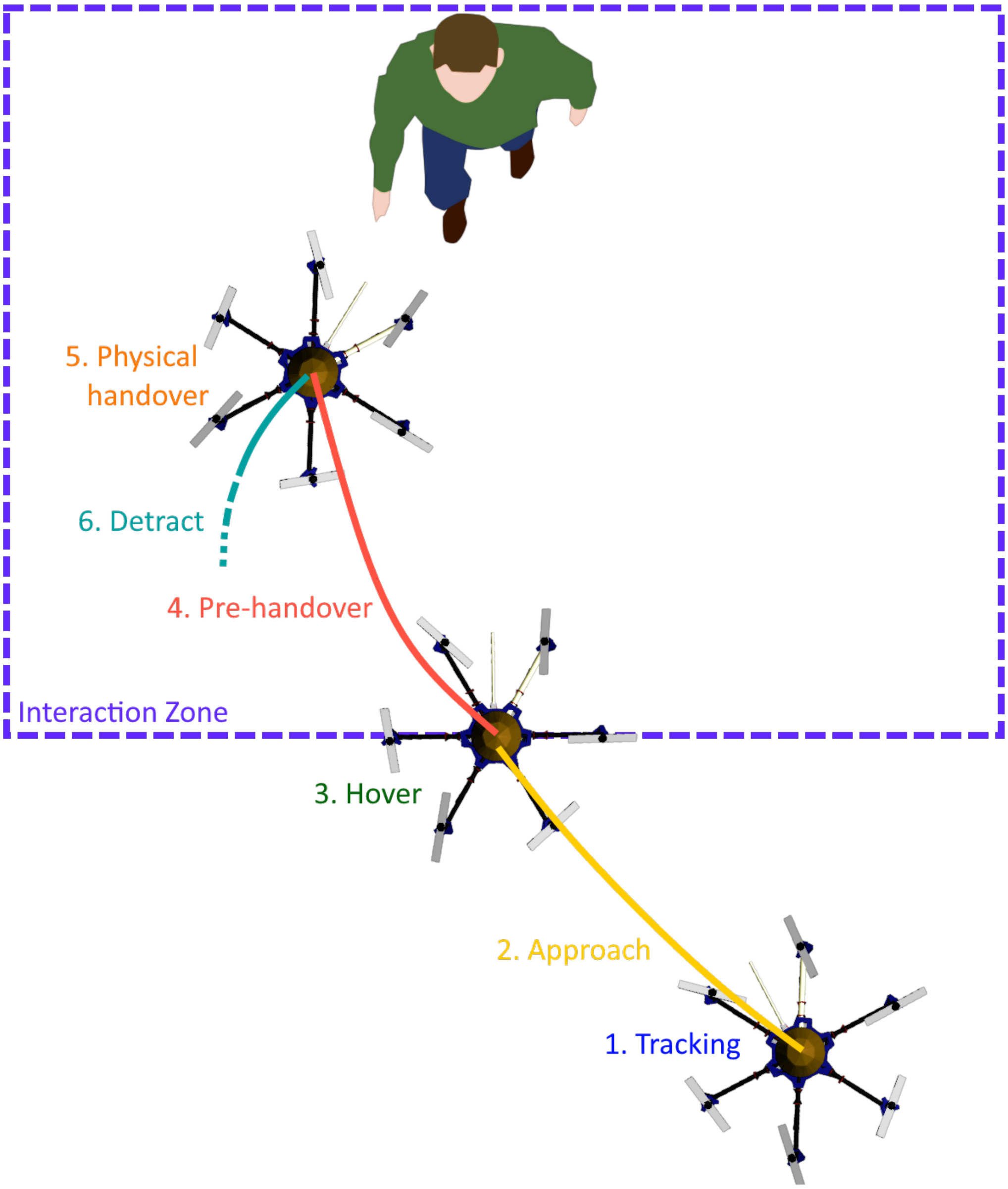


Figure 5.1: Illustration of the approach phase experiments.

[1]The detraction is left out of the results for clarity.

For most results six plots will be shown:

1. A trajectory plot, showing the top-down path of the aerial robot's position $\mathbf{p}_B$ as well as the position of the end-effector $\mathbf{p}_{EE}$. Yellow stars dictate the position of the Aruco tags $\mathbf{p}_{M_1}$ (Human's right hand), $\mathbf{p}_{M_3}$ (Human's face/center) and $\mathbf{p}_{M_2}$ (Human's left hand). In cases where the human moves, the estimated $\mathbf{p}_{M_3}$ is plotted as a yellow line (as soon as the movement starts).

2. Five graphs that share the same time axis. If highlights are present in the plot, blue represents *tracking*, yellow represents the first approach towards the interaction space (2.5 meters distance) and red represents the second approach, or pre-handover, towards the Handover Location. Legibility is never on for the first approach.

   (a) Position of body frame $\mathbf{p}$ in [m]

   (b) Velocity of body frame $\dot{\mathbf{p}}$ in [m/s]

   (c) Attitude of body frame $\mathbf{q}$ in [deg]

   (d) Propeller thrust γ in [N]

   (e) Values of cost for different objectives[2]

5.1 Free Flight, Tracking and Velocity constraint

In figure 5.3 the default flight performance is shown. First part of the trajectory is without perception objective. The robot flies to position [1,-1,3], then to [2,1,2], next it turns on *tracking* mode and moves to position [-2,1,2] while orienting towards the human. Tracking mode means that the perception objectives, the perception constraints and the velocity constraints are turned on.

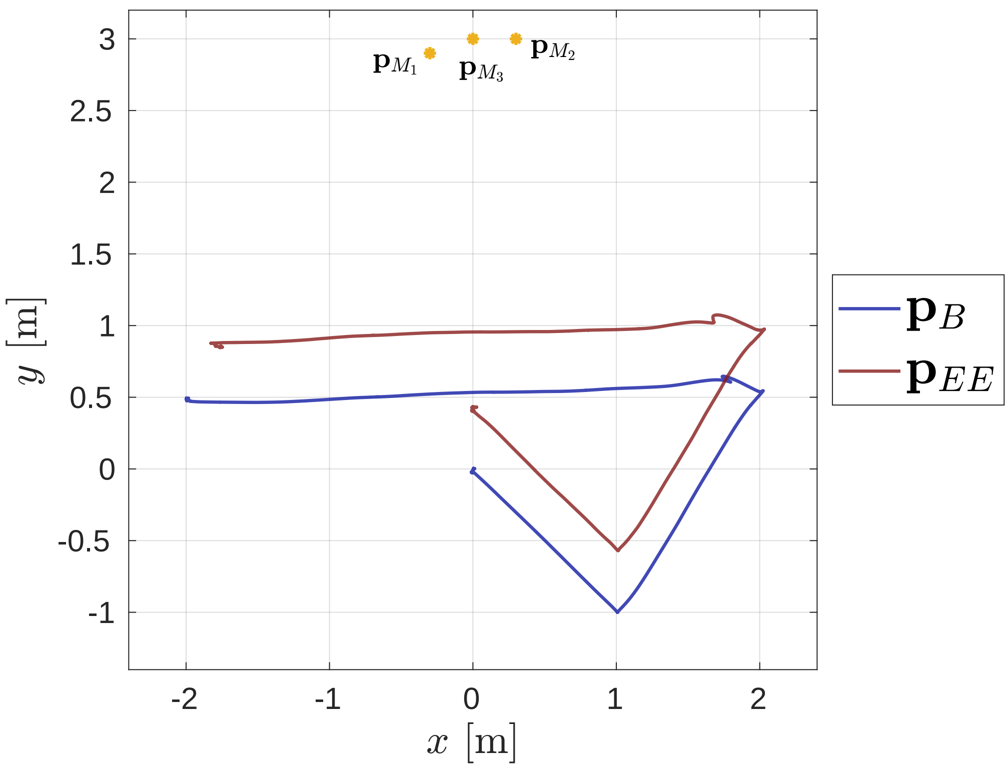


Figure 5.2: Position (x, y) of body and end-effector frame, in yellow the right hand and face of the human.

[2]The value is estimated based on the logged data and may differ from the internal value that isn't logged.

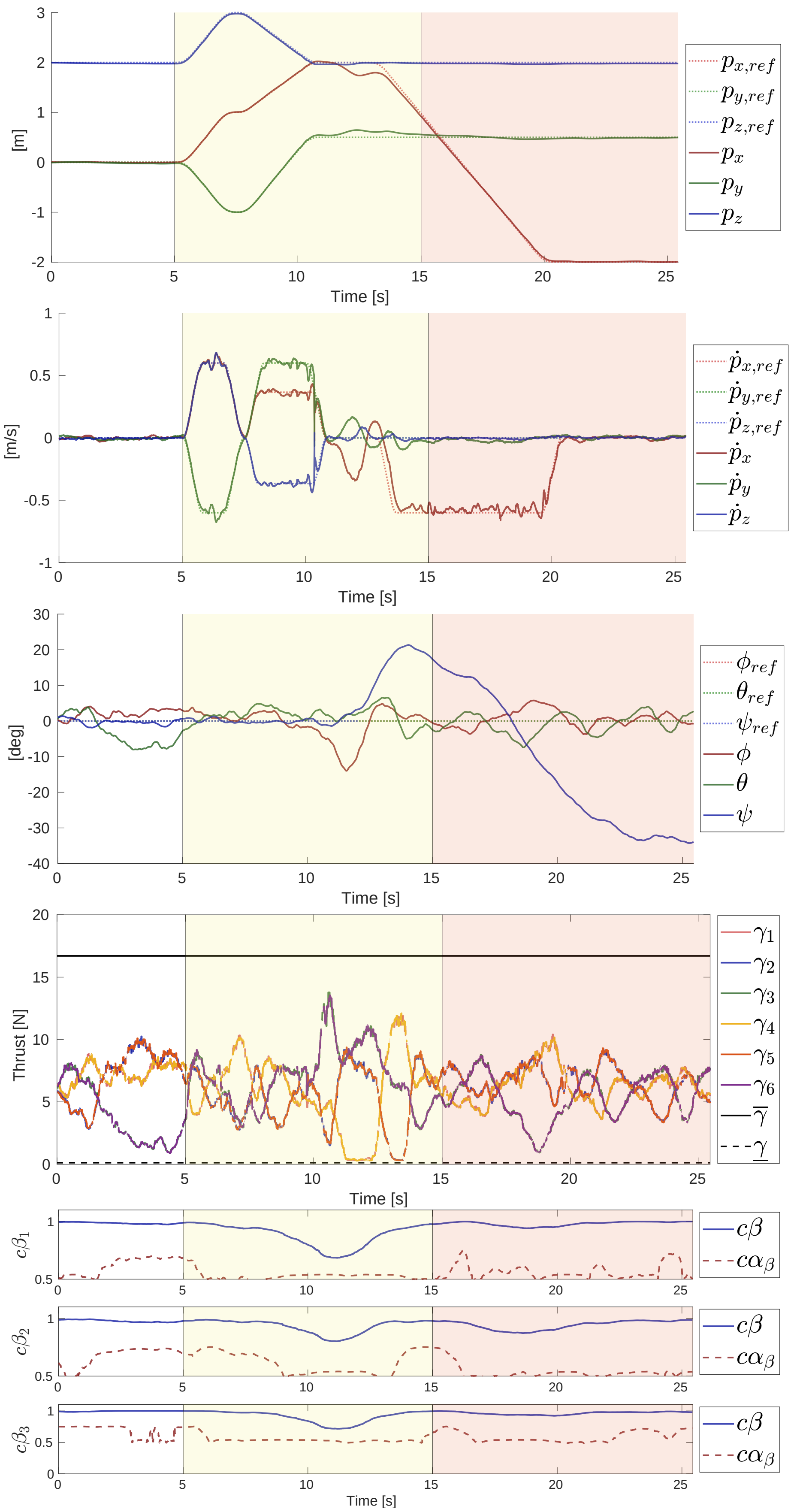


Figure 5.3: Perception objective in action.

5.2 Approach Phase

The approach phase exists of two parts, in the first part the robot approaches the interaction space and in the second part, the pre-handover approach is done. Refer back to figure 5.1 for an illustration of the approach.

First the approach is shown without legibility in figures 5.4, 5.5.

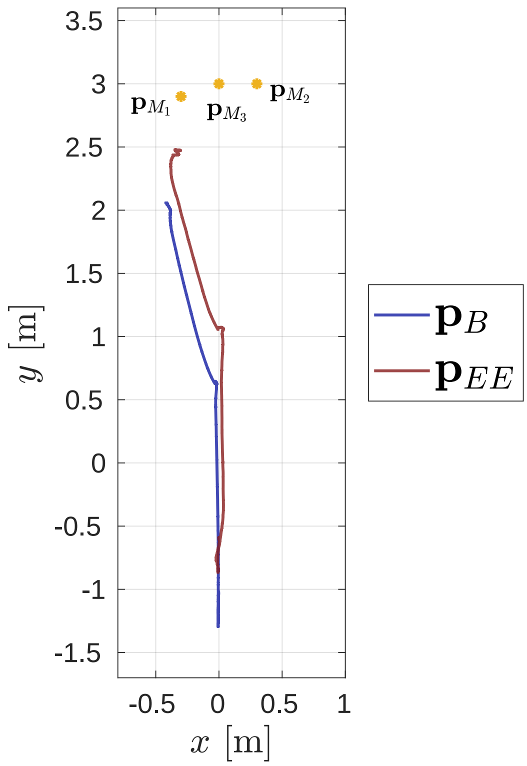


Figure 5.4: Position (x, y) of body and end-effector frame, in yellow the right hand and face of the human.

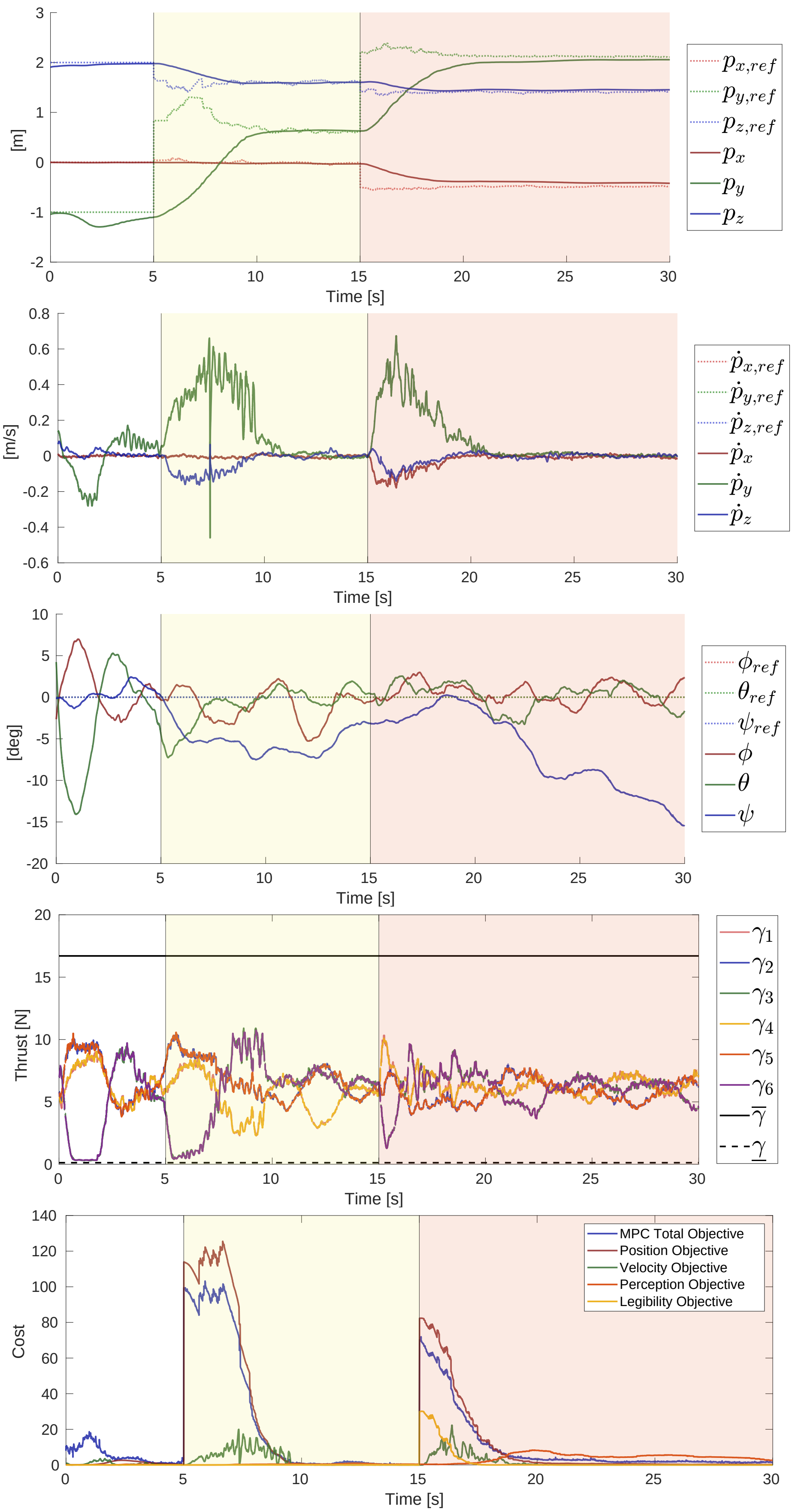


Figure 5.5: Approach to human without legibility active.

Legibility is turned on in figures 5.6, 5.7. The starting point is varied to show how the robot first approaches a location in front of the human before starting the pre-handover. The first part of the approach is without legibility enabled and the second part, which is the pre-handover, has legibility enabled. Between the two approaches, the robot hovers shortly at the location where, in a real scenario, some communication will take place between human and robot.

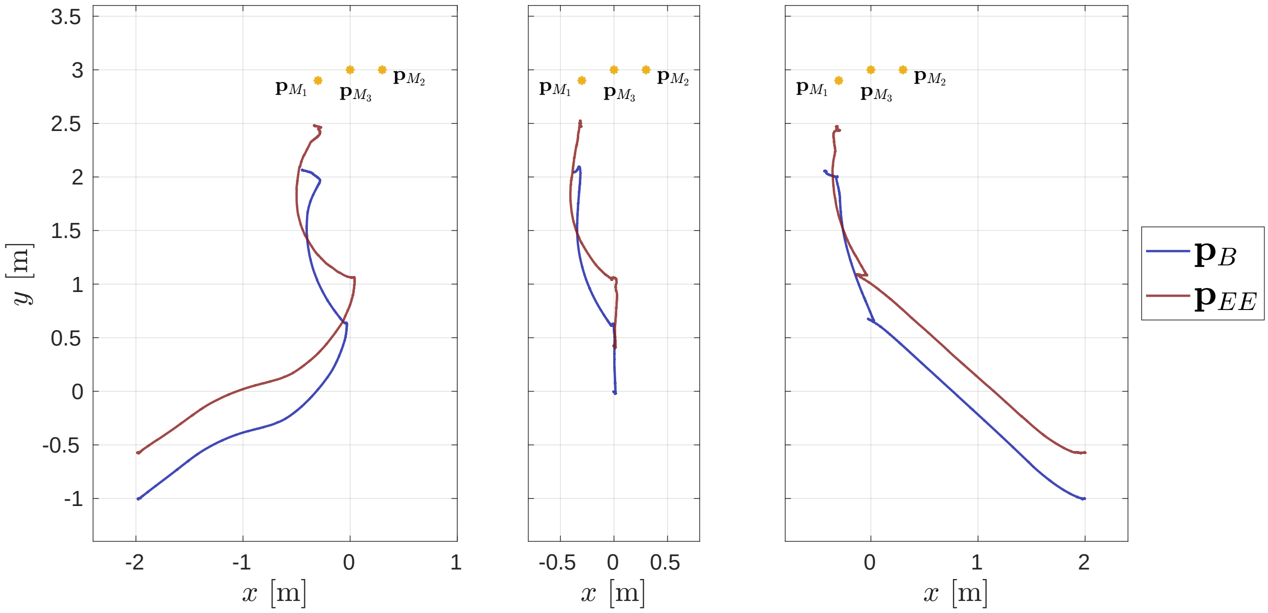


Figure 5.6: Position (x, y) of body and end-effector frame. Double approach with legibility in the interaction space. Starting points (left to right): [-2,-1], [0,0] and [2,-1].

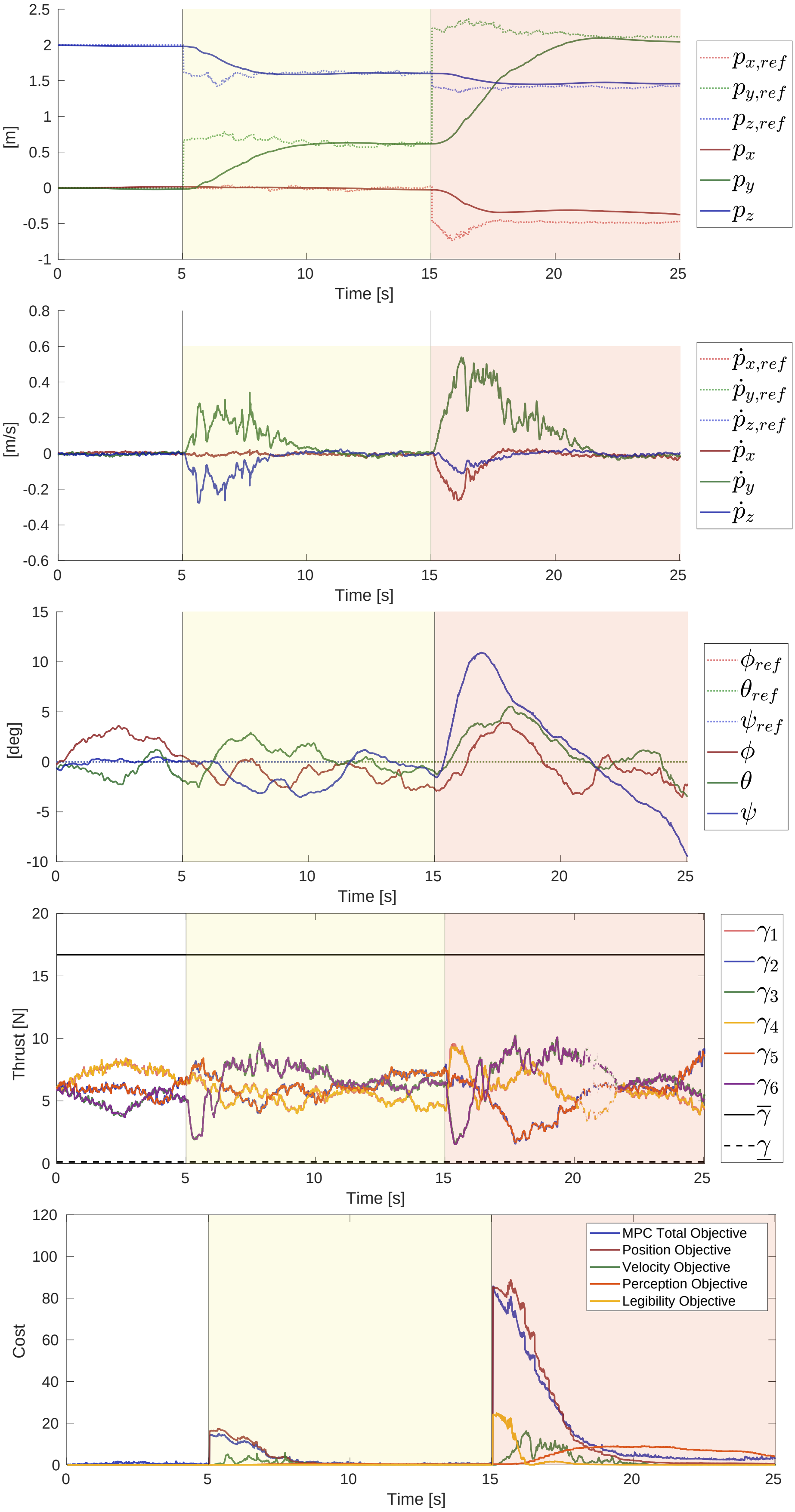


Figure 5.7: Approach to human with legibility active.

5.3 Human moves after approach

In figures 5.8, 5.9 after the approach, the human backs away diagonally, then steps sideways and then makes a step forward towards the Aerial Robot. Legibility is turned off here.

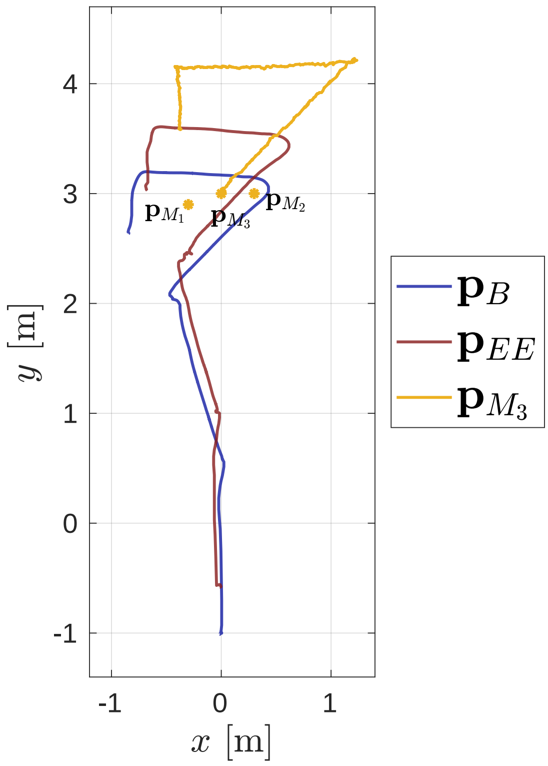


Figure 5.8: Position (x, y) of body and end-effector frame, in yellow the estimated human position.

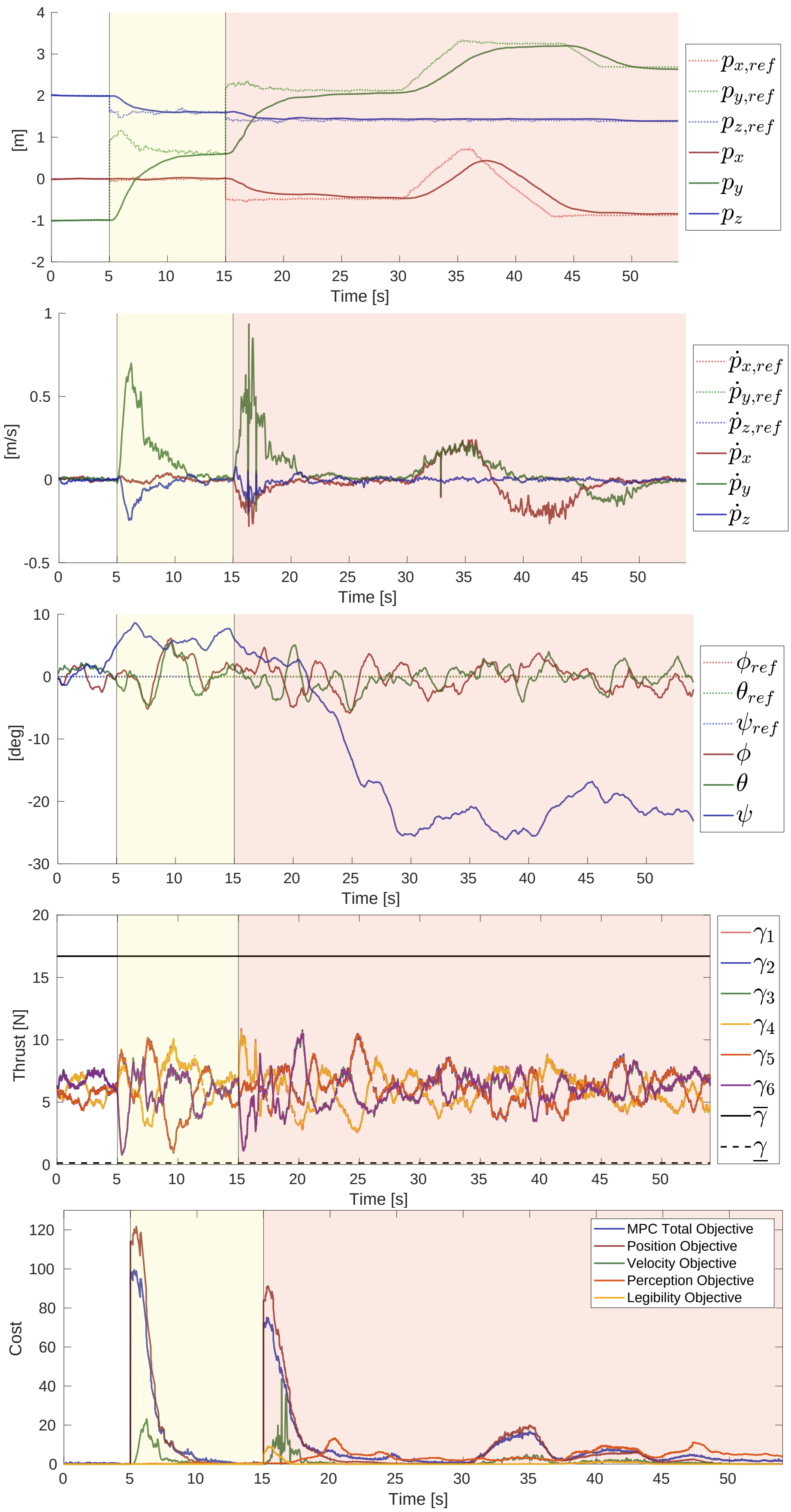


Figure 5.9: The robot approaches the human (without legibility) and then the human moves three times.

In figures 5.10, 5.11 the human moves again, but this time the legibility is turned on.

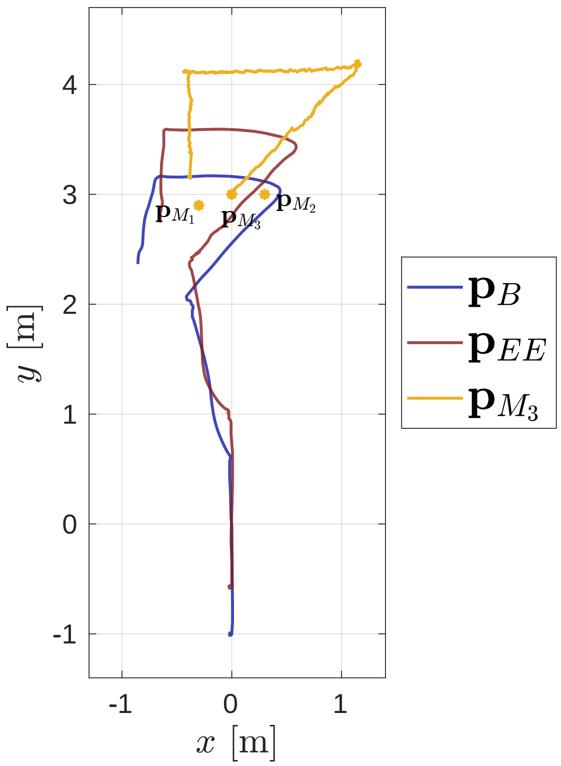


Figure 5.10: Position (x, y) of body and end-effector frame, in yellow the estimated human position.

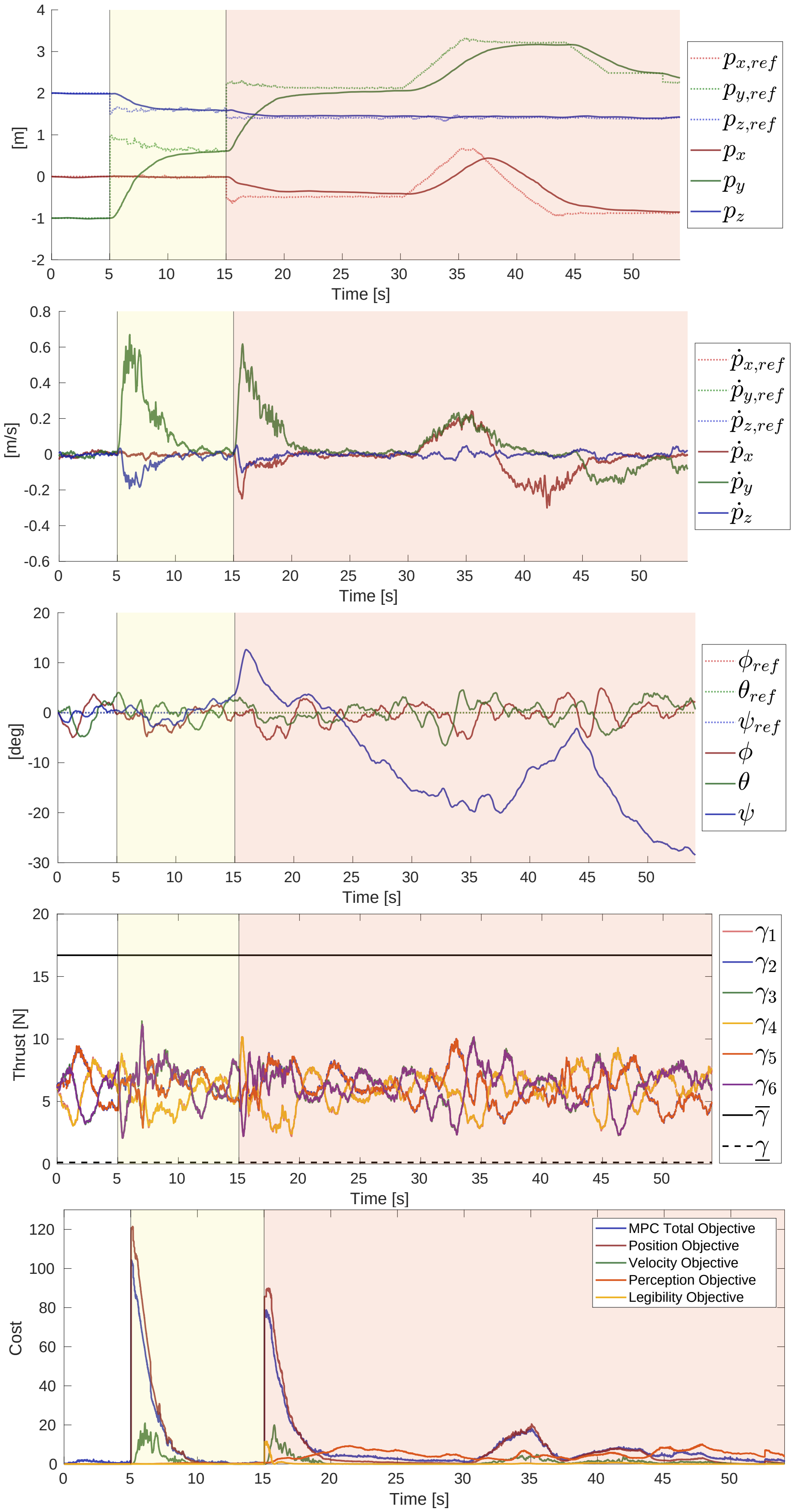


Figure 5.11: The robot once again approaches the human, with legibility active, and then the human moves three times.

5.4 Legibility Measure

To test whether the legibility-inspired objective actually increases the legibility, equation 3.7 is used to calculate the legibility of the pre-handover approach. The legibility is calculated for both goals and both with and without the objective turned on. In table 5.1, the results are given.

Table 5.1: Legibility measure values towards the actual goal M_1 relative to the other possible goal M_2. Different weights are used for the objective. The legibility is calculated for both relevant frames.

Objective Weight	Value	
	body frame	end-effector frame
0	0.5505	0.5332
0.2	0.5418	0.5376
0.3	0.5415	0.5323

We see that the legibility is higher than 0.5 which means that for all trajectories it's more legible towards the goal than to other goals. The sum of legibility measures, over all goals, is 1. However, the change in legibility is not significant enough to make concrete conclusions. In the discussion (chapter 6) some issues with the applied legibility measure are highlighted.

6 Discussion

The first experiment shows that the free flight performance is good. The reference state from maneuver is followed closely. The velocity plot shows some noise on the velocity, this seems to be at least partly from the measurement method.

In the perception experiment, the yaw seems a bit damped and doesn't react as fast as desired. This doesn't cause any malfunction but should be addressed. In the uavmpc software package, a function is used to free the yaw. This function sets the reference yaw to the current yaw, essentially making sure that the yaw error is always zero. However since the NMPC has a predictive horizon, the error could be non-zero for later time steps. Thus, the solver will try to keep the yaw at the same angle as it currently is. In systems where Euler angles are used, the rotation angles are separately weighted in the cost function. This is not the case when using quaternions because the geodesic distance between the reference and actual quaternion is used which is a single value. Therefore only one weight can be used for all rotations.

Experiment 5.2 shows that the Aerial robot can approach the human from different starting positions and follow that up with the pre-handover approach. The implemented legibility objective clearly shows the intended goal very fast by both having rotation and exaggerated movement towards the side that the handover will take place at. Although the rotation component is quite sudden the trajectory is smooth. Arguably this would be comfortable for a human that's interacting with the robot. Further research could replicate these experiments with real humans and measure their comfortability.

The final experiment tests the ability to stay at the handover location when the human moves. When the human (model) moves away, the aerial robot follows in a calm manner. It's however a bit too calm when the human moves towards the robot. Because the MPC has no directional relation to the human velocity, there is no change in the objective when the human starts to move closer with respect to when it moves away. There are three concepts that could improve this. First of all, the MPC velocity reference $\mathbf{v}_r$ could be set to be the human's velocity $\mathbf{v}_H$, or rather the perceived human feature $\mathbf{v}_{M_3}$. Secondly, a special objective could be introduced that will repulse the robot when it's close to the human, a type of object avoidance. Finally, the earlier velocity constraint could be modified to be relative to the human velocity, similar to the first concept. This one is probably not realistic since the sudden constraint change will likely cause the solver to fail.

The measured legibility values are not significant enough to draw objective conclusions. The problem probably lies in the evaluation of the measure. In [4] and [40] a clean trajectory is fed into the equations. However, for the results in this thesis, the data is logged from a simulated environment and the trajectory is generated in real-time by a controller that is optimizing many variables. This introduces some factors that increase the cost function, such as noise and non-optimal behaviour. When looking at equation 3.5, we see that the cost of the actual trajectory is compared to the optimal trajectory. Naturally, the generated trajectory has a high cost when compared to an optimal trajectory. It could thus be argued that the probabilities are deflated for any in-flight generated trajectory. The numerator in 3.5 contains two costs, $C(\xi_{S\rightarrow Q})$ and $C(\xi^*_{Q\rightarrow G})$. If the first cost is much higher it will dominate the exponential value even though the full trajectory needs to be considered. Concluding, we could say that the cost function might need to be reconsidered for real signals.

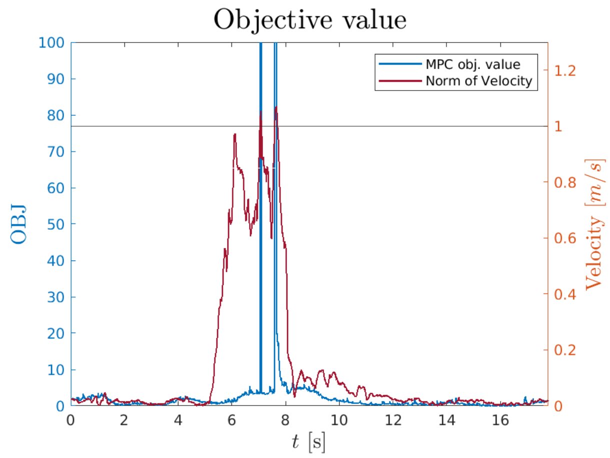
Another factor that might play a role is that the controller slows down the robot when it gets closer to the objective. This makes it hard to determine the time at which the goal is reached. In this thesis, a fixed time is used, but it might be better to use a fixed distance-based condition to determine this point. Another problem pops up however, namely that the literature considers the trajectory to have a fixed time. It's unclear how a difference in time influences the legibility measure. To come back to the used method, a large part of the trajectory is close to the actual goal thus increasing the legibility with respect to the other goal. This part at the end of the trajectory is very always close to the goal regardless of the legibility objective. This makes it so that the measurement focuses too little on the start of the legible trajectory. It should be said that the time function $f(t) = (T - t)$ is included for this exact purpose, but due to the nature of the system that slows down towards the end, this might not have enough effect. Another time function is proposed: $f(t) = (T - t)^2$. This will put even more emphasis on the start. It would be interesting to study the effect that a different order time function has on the legibility measure, but this is left for future research.

Even though subjectively, the rotation quickly shows the intent, the objective legibility measure however doesn't take into account the rotation and only the position. To compare this with a human to human handover, this would be like only looking at the hand while the giver is gazing very intensely at his intended goal. In human communication, various signals are utilized and the robot should be able to do so as well. This is arguably a shortcoming of the one-dimensional legibility measure.

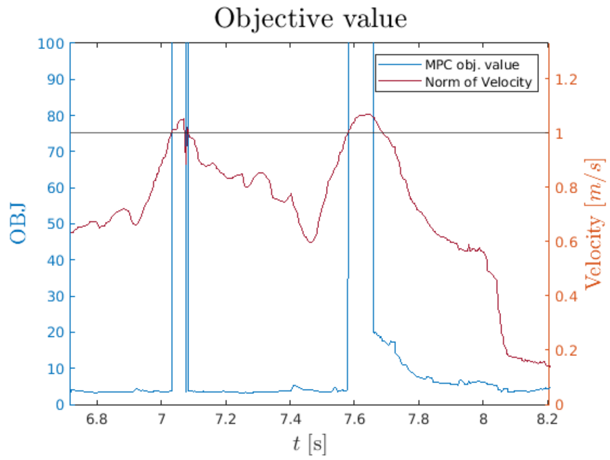
The legibility measure is also not thoroughly investigated due to time constraints. Tuning the parameters takes a lot of time and to show a correlation between the parameters/weights and the legibility requires multiple data points.

6.1 Velocity constraint

In the shown results the velocity constraint was never violated, however, if the weights are not tuned properly, or when the input trajectory (e.g. from GenoM3-Maneuver) is too extreme, the velocity might cross the upper boundary and cause the MPC to be unsolvable. This is obviously undesired since it will rapidly cause a crash or emergency protocol if the system does not slow down. In figure 6.1a the velocity crosses the boundary twice after which the UAV crashes. When we zoom in with figure 6.1b, it's clear that the objective value explodes precisely at the time the constraint is violated. In the first instance, the system slows down quick enough to continue, but in the next instance the period is too long and the controller activates emergency mode. (Which is why the value shoots down even with the constraint violated). There are a few ways one could remediate the problem, but at the end of the day, this is a software issue.



(a) Norm of velocity is plot against the objective value in the MPC



(b) Zoomed in it's clear that the obj. value explodes when the velocity crosses the constraint of 1 m/s

Figure 6.1: Velocity constraint

7 Conclusion

With advancements in Aerial Robotics and Human-Robot Interaction, the aerial handover is a compelling next step in Aerial Human-Robot Interaction. In this thesis, the handover is contextualized inside the Aerial domain. In the literature review, the handover is dissected into the approach phase and physical handover. The perception necessities for Human-Robot communication and interaction are highlighted. Together with human-aware motion planning motives, this led to the choice of using an optimization control method, Non-linear Model Predictive Control.

Based on the literature several desires are formulated into objectives and constraints. Besides servoing to the handover location, the aerial robot should be comfortable and safe. First, a velocity constraint is introduced. Secondly, the aerial robot should clearly show its intent, having legible motion. Models of legibility from previous literature inspired a legibility objective that is shown to subjectively increase the legibility. To objectively prove this, the legibility measure was also calculated but didn't show significant results. However, this is likely due to limitations in the measurement quality and the fact that the controller has to optimize for more than just legibility.

Finally, the method is implemented and tested in simulation using the software-in-the-loop approach. The simulation results show that the control method can smoothly approach humans while keeping the constraints and velocities. The generated trajectory is subjectively more legible with the hand-designed legibility-inspired objective. The aerial robot can somewhat follow a human moving around but is a bit slow when the human moves towards the robot.

Returning to the research questions from chapter 1. By implementing a velocity constraint as well as using standard velocity damping via the state, the aerial robot keeps a safe velocity. Furthermore, using the perception objective makes sure that the human is always perceived by the robot.

To create comfortable behaviour for humans, the optimization weights of the model predictive control scheme are tuned. Because the aerial robot always faces the human, this generates a sense of security to the human. The two-step approach makes sure that the robot is in front of the human before the interaction which is predictable to the human.

With a legibility-inspired objective, the aerial robot quickly shows its intent when starting the pre-handover approach. The thesis also gives tools to measure the legibility and enable future tuning with hand-designed parameters. Using these tools, a better cost function could be developed to further the human-aware capabilities.

In human-robot handovers, it's very important that communication is well established. The perception of the robot should accurately know the 3D position and orientation of the human. From the literature review, it's substantiated that Human pose estimation coupled with 3D vision is needed. Furthermore, to communicate actions, commands and intent, a gesture recognition module should be part of the final design.

The control method chosen in this thesis is the Non-linear Model Predictive Control method. Optimization control is suitable for this kind of complex system that has many different objectives. Furthermore, NMPC is being used more frequently for physical interaction in aerial vehicles and proving to be effective in fully actuated UAVs.

7.1 Future work

The thesis' core work lied in development and simulation. The natural next step is to use the implemented work on a physical aerial robot. The current implementation was based on a software-in-the-loop approach, thus the framework is able to run on-board the real aerial robot. There are a few simplifications that have been made for simulation that must be worked out.

First of all, instead of human pose estimation, we've opted to use Aruco tags in simulation. Robust human pose estimation is needed that can determine the 3D positions of relevant features (hands, shoulders, head) and the human's global orientation. Several algorithms and papers have been suggested in the literature review. Although not part of this thesis, some time has been spent analysing these methods and for future work, the following method is proposed. Use OpenPose to determine the human features in 2D images. OpenPose is pretty reliable, very fast, open-source and has an established community. Next using a 3D camera, these features could be interpolated into 3D space. Some care should be taken to filter noise and check for impossible measurements.

Secondly, to create a real use system, a robust way to communicate has to be established. The human should be able to give several commands: emergency stop, approach at distance, start handover (approach handover location) and finish interaction. As discussed in the literature review, gesture recognition is the best option here. Gesture recognition is in active development as part of the Aerial-Core project at the Aristotle University of Thessaloniki.

Besides these large scale continuations, there are a few things that could improve the controller. As discussed before, the quaternion control is great to prevent singularities but lacks a weighted control on distinct orientations. So that yaw can have lower or zero weight associated with it.

The velocity constraint is important for human safety. With the current NMPC software, the solver can crash in situations close to the constraint. Adding a slack variable could allow solutions of the solver even after crossing the constraint.

Currently, the perception method assumes a static perception feature, with the assumption that it moves slow enough compared to the MPC time horizon. It might be optimal however if the feature position would also be modelled using its velocity and filtering (e.g. Kalman Filter).

If the aerial robot is to be employed in a realistic situation, it's advised to also add other cameras to find human presence. These cameras would have to be connected. It would be interesting to extend the perception objectives and constraints for overlapping camera views.

One limitation of the simulations is that the human body frame $\mathcal{F}_H$ was assumed to have one orientation. As part of the GenoM framework that was used as the starting point, the ArucoTag module was slightly outdated and didn't yet feed the orientation of the tags. Migrating the work to the more recent uavmpc module could allow the human orientation to transform the approach locations.

Implementing legibility has been a large part of the thesis. Although a legibility measure is adopted from [4], the measure is based on certain assumptions, especially the cost function, but also the time-dependent weighting. In future research, the legibility of an aerial vehicle could be studied with human subjects. The legibility-inspired objective is designed to give researchers control parameters to modify the trajectory and measure the objective and subjective legibility as well as comfort. An objective measure is 'how fast does the human know the goal' and subjectively the subjects could be asked 'how clear where the robots intentions to you?'. In all experiments, human comfort is important because a robot could exaggerate fast movements to show intent, but this is not pleasant in Human-Robot Interaction.

Bibliography

[1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] T. B. Sheridan, "Human-Robot Interaction," *Human Factors*, vol. 58, no. 4, pp. 525–532, 2016.

[3] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear Model Predictive Control with Enhanced Actuator Model for Multi-Rotor Aerial Vehicles with Generic Designs," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 100, no. 3-4, pp. 1213–1247, 2020.

[4] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," *ACM/IEEE International Conference on Human-Robot Interaction*, vol. 1, pp. 301–308, 2013.

[5] V. Ortenzi, A. Cosgun, T. Pardi, W. Chan, E. Croft, and D. Kulic, "Object Handovers: a Review for Robotics," no. July, pp. 1–20, 2020. [Online]. Available: http://arxiv.org/abs/2007.12952

[6] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard, "A human-inspired controller for fluid human-robot handovers," *IEEE-RAS International Conference on Humanoid Robots*, pp. 324–331, 2016.

[7] A. M. Bestick, S. A. Burden, G. Willits, N. Naikal, S. S. Sastry, and R. Bajcsy, "Personalized kinematics for human-robot collaborative manipulation," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, pp. 1037–1044, 2015.

[8] J. Mainprice, M. Gharbi, T. Simeon, and R. Alami, "Sharing effort in planning human-robot handover tasks," in *IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, 2012.

[9] E. A. Sisbot, K. F. Marin-Urias, R. Alami, and T. Siméon, "A human aware mobile robot motion planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[10] M. Wu, B. Taetz, E. D. Saraiva, G. Bleser, and S. Liu, "On-line Motion Prediction and Adaptive Control in Human-Robot Handover Tasks," in *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts, ARSO*, vol. 2019-Octob. IEEE Computer Society, oct 2019, pp. 1–6.

[11] A. Bestick, R. Pandya, R. Bajcsy, and A. D. Dragan, "Learning human ergonomic preferences for handovers," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3257–3264, 2018.

[12] M. Melchiorre, L. S. Scimmi, M. Stefano, and S. P. Pastorelli, "Vision-based control architecture for human-robot hand-over applications," 2020.

[13] A. Fishman, C. Paxton, W. Yang, D. Fox, B. Boots, and N. Ratliff, "Collaborative interaction models for optimized human-robot teamwork," in *IEEE International Conference on Intelligent Robots and Systems*. Institute of Electrical and Electronics Engineers Inc., oct 2020, pp. 11 221–11 228.

[14] NaturalPoint, "Optitrack - Motion Capture Systems." [Online]. Available: optitrack.com

[15] P. Kratzer, M. Toussaint, and J. Mainprice, "Prediction of Human Full-Body Movements with Motion Optimization and Recurrent Neural Networks," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1792–1798, 2020.

[16] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields." *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 172–186, jan 2021.

[17] T. Schnürer, S. Fuchs, M. Eisenbach, and H. M. Groß, "Real-time 3D Pose Estimation from Single Depth Images," *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 5, no. January, pp. 716–724, 2019.

[18] T. Blanc-Beyne, A. Carlier, S. Mouysset, and V. Charvillat, "Unsupervised Human Pose Estimation on Depth Images," in *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track*, Y. Dong, D. Mladenić, and C. Saunders, Eds. Cham: Springer International Publishing, 2021, pp. 358–373.

[19] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3D pose estimation from a single depth image," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 731–738, 2011.

[20] F. Patrona, I. Mademlis, and I. Pitas, "Self-Supervised Convolutional Neural Networks for Fast Gesture Recognition in Human-Robot Interaction," *2021 10th International Conference on Information and Automation for Sustainability, ICIAfS 2021*, pp. 88–93, 2021.

[21] ——, "An Overview of Hand Gesture Languages for Autonomous UAV Handling," *AIRPHARO 2021 - 1st AIRPHARO Workshop on Aerial Robotic Systems Physically Interacting with the Environment*, 2021.

[22] C. Papaioannidis, D. Makrygiannis, I. Mademlis, and I. Pitas, "Learning Fast and Robust Gesture Recognition," *Proceedings of European Signal Processing Conference*, pp. 761–765, 2021.

[23] A. G. Perera, Y. W. Law, and J. Chahl, "UAV-GESTURE: A Dataset for UAV Control and Gesture Recognition," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11130 LNCS, pp. 117–128. [Online]. Available: http://link.springer.com/10.1007/978-3-030-11012-3_9

[24] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, and A. Franchi, "6D physical interaction with a fully actuated aerial robot," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5190–5195, 2017.

[25] M. Ryll, D. Bicego, and A. Franchi, "A Truly Redundant Aerial Manipulator exploiting a Multidirectional Thrust Base," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 138–143, 2018.

[26] ——, "Modeling and Control of FAST-Hex : a Fully Actuated by Synchronized Tilting Hexarotor," no. 644271, pp. 1689–1694, 2020.

[27] L. . Peric, M. . Brunner, K. . Bodie, M. . Tognon, R. Siegwart, L. Peric, M. Brunner, K. Bodie, and M. Tognon, "Direct Force and Pose NMPC with Multiple Interaction Modes for Aerial Push-and-Slide Operations ETH Library Direct Force and Pose NMPC with Multiple Interaction Modes for Aerial Push-and-Slide Operations," *IEEE International Conference on Robotics and Automation*, 2021. [Online]. Available: http://doi.org/10.1109/ICRA48506.2021.9561990http://ieeexplore.ieee.org/

[28] G. Corsini, M. Jacquet, A. E. Jimenez-Cano, A. Afifi, D. Sidobre, and A. Franchi, "A General Control Architecture for Visual Servoing and Physical Interaction Tasks for Fully-actuated Aerial Vehicles," *AIRPHARO 2021 - 1st AIRPHARO Workshop on Aerial Robotic Systems Physically Interacting with the Environment*, 2021.

[29] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi, "6D interaction control with aerial robots: The flying end-effector paradigm," *International Journal of Robotics Research*, vol. 38, no. 9, pp. 1045–1062, 2019.

[30] M. Fumagalli, R. Naldi, A. Macchelli, F. Forte, A. Q. Keemink, S. Stramigioli, R. Carloni, and L. Marconi, "Developing an aerial manipulator prototype: Physical interaction with the environment," *IEEE Robotics and Automation Magazine*, vol. 21, no. 3, pp. 41–50, 2014.

[31] B. Yüksel, C. Secchi, H. H. Bülthoff, and A. Franchi, "Aerial physical interaction via IDA-PBC," *International Journal of Robotics Research*, vol. 38, no. 4, pp. 403–421, 2019.

[32] M. Holland, "Control methods and architectures for aerial manipulators in physical interaction with humans," Master's thesis, University of Twente, 2021.

[33] M. Jacquet and A. Franchi, "Motor and Perception Constrained NMPC for Torque-Controlled Generic Aerial Vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 518–525, 2021.

[34] A. Alharbat, H. Esmaeeli, and A. Franchi, "Exploiting Non-Linear Model Predictive Control in Contact-Based Aerial Physical Interaction," Tech. Rep., 2021.

[35] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[36] F. Chaumette and S. Hutchinson, "Visual Servo Control Part II: Advanced Approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[37] M. Kivits, "Cooperative target tracking with heterogeneous robots using nonlinear model predictive control," pp. 0–1, 2021.

[38] M. Jacquet, M. Kivits, H. Das, and A. Franchi, "Motor-level N-MPC for Cooperative Active Perception with Multiple Heterogeneous UAVs," *IEEE Robotics and Automation Letters*, 2022.

[39] G. Michieletto, M. Ryll, and A. Franchi, "Fundamental Actuation Properties of Multirotors: Force-Moment Decoupling and Fail-Safe Robustness," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 702–715, 2018.

[40] A. Dragan and S. Srinivasa, "Generating Legible Motion," 2016.

[41] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2149–2154, 2004.

[42] C. E. Agüero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. L. Rivero, J. Manzo, E. Krotkov, and G. Pratt, "Inside the Virtual Robotics Challenge: Simulating Real-Time Robotic Disaster Response," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 494–506, 2015.

[43] A. Mallet, C. Pasteur, M. Herrb, S. Lemaignan, and F. Ingrand, "GenoM3: Building middleware-independent robotic components," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4627–4632, 2010.

A Embedded System comparison

Table A.1: Comparison data for several Embedded Systems

Name	Weight (g)	Power consumption (Watt)	CPU Cores	CPU type	AI Performance	CUDA Cores	Tensor Cores	Price (ex. VAT)
Intel NUC	470	120?	6	Intel	-	0	0	€660.00
Jetson Nano	249	5 | 10	4	Arm Cortex-A57 MPCore	472 GFLOPs	128	0	€82.50
Jetson Xavier NX	400	10 | 15	6	NVIDIA Carmel Armv8.2	21 TOPs	384	48	€326.00
Jetson AGX Xavier	649	10 | 15 | 30	8	NVIDIA Carmel Armv8.2	32 TOPs	512	64	€573.00