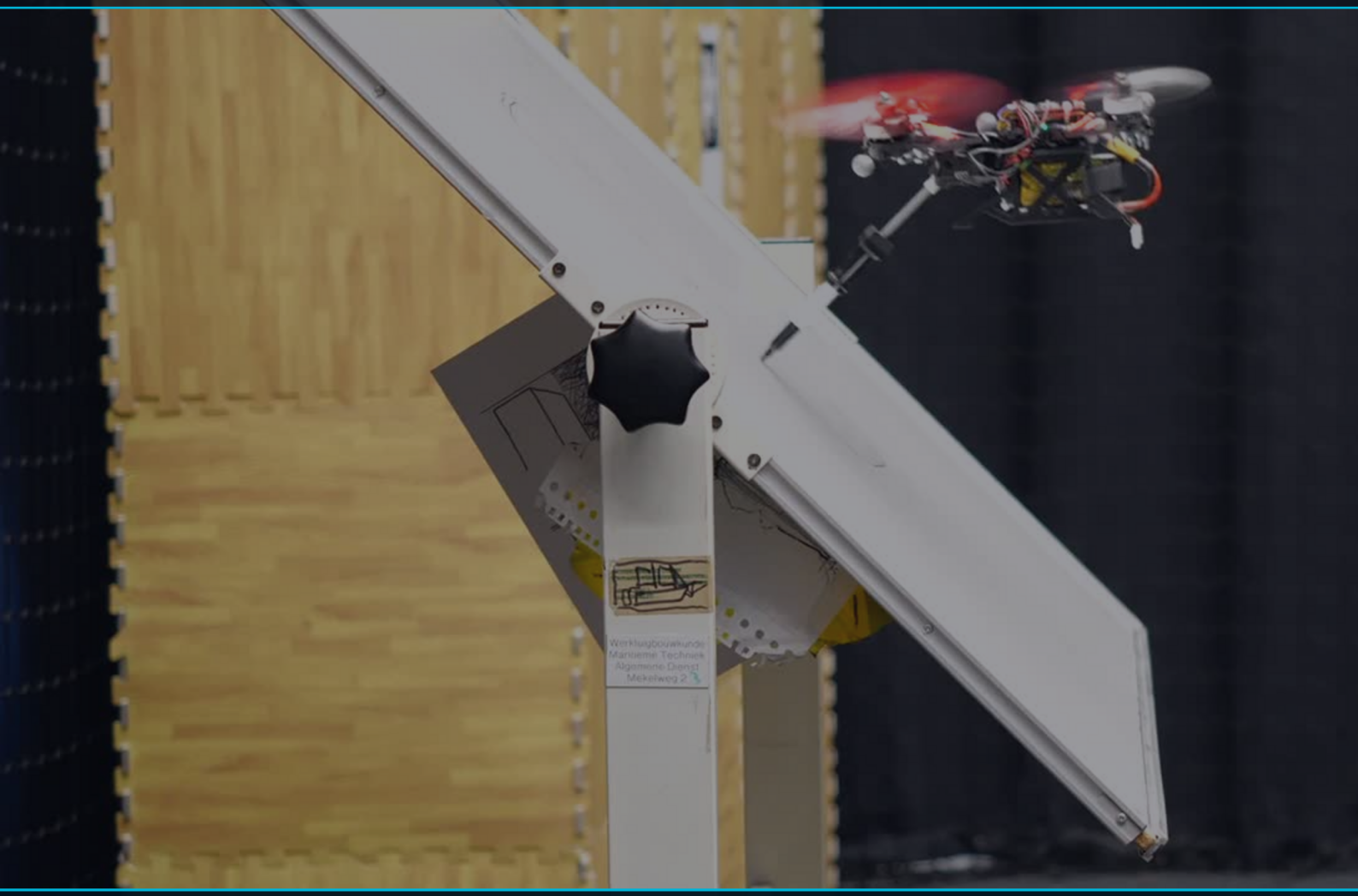
DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)
Delft University of Technology

MASTER OF SCIENCE THESIS

Incremental Nonlinear Dynamic Inversion for Aerial Manipulation

D. Liu
AUTHOR



TU Delft

DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)
Faculty of Mechanical Engineering (ME)
April 2, 2026

Incremental Nonlinear Dynamic Inversion for Aerial Manipulation

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

D. Liu

April 2, 2026

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Abstract

The field of aerial manipulation has many applications, for example autonomous maintenance in difficult to reach or hazardous locations. However, achieving precise control of a floating-base multi-body system presents considerable challenges, particularly given the requirement for robust performance under strong external disturbances in uncontrolled environments. To address these challenges, this work investigates the Incremental Nonlinear Dynamic Inversion (INDI) method as a promising control framework.

While INDI has been commonly applied to multi-rotor platforms and other single-body aerial vehicles, its use in aerial manipulation remains largely unexplored. Moreover, existing applications have focused on reference tracking in free flight, leaving the application of INDI in Aerial Physical Interaction (APhI) tasks unaddressed. To fill this research gap, we propose a novel Nonlinear Model Predictive Control (NMPC) formulation, building upon state-of-the-art works in multi-rotor control and aerial manipulation, and combine it with INDI as an inner-loop controller similar to [55]. Two control schemes are developed and evaluated: one in which INDI is applied solely to attitude control, and one in which INDI is used in both position and attitude control. The aerial manipulator platform used and studied in this work is a Differential Shoulder Aerial Manipulator (DSAM) which consists of an underactuated quadrotor base equipped with a two-degrees-of-freedom (DoF) robotic arm.

The proposed controllers are validated through both simulation and real-world experiments, including free-flight maneuvers and APhI sliding tasks. Real world results demonstrate that the baseline controller without an INDI inner loop is unable to successfully perform the sliding tasks, whereas both INDI-augmented schemes are able to both track end-effector position and attitude and a desired reference contact force simultaneously. This is shown for several trajectories on both a whiteboard using a marker and on a blackboard using a crayon. Also simultaneous end-effector pose and force tracking in different directions in the world frame are shown. Finally it has been experimentally validated that extending the controller with a INDI position control layer reduces the end-effector position tracking error compared to one without the INDI position control layer. In a real world experiment in which a figure-eight trajectory is tracked while simultaneously tracking a desired contact force with external wind disturbance, it has even shown a reduction of 45% in maximum end-effector position tracking error.

Table of Contents

Chapter 1

Introduction

The Unmanned Aerial Vehicle (UAV) is becoming increasingly important in our society. One of the reasons is their agility and ability to reach locations that are difficult to reach for humans. They are already being used in many sensing applications such as exploration [50] and inspection tasks [3]. Aerial manipulators are UAVs designed to not only see, but also interact with the environment. They can be used for many applications such as maintenance [23, 6], valve turning [31], assembly [18], or load transportation [5].

An aerial manipulator consists of an aerial platform and an attached tool for contact with the environment. As tasks require contact with the environment, these are also referred to as Aerial Physical Interaction (APhI) tasks. Load transportation was one of the first APhI tasks accomplished. Most early works used helicopters as platforms due to their superior payload capacity and operation time compared to multirotor aerial vehicles [4, 5]. These tasks do not require active manipulation. The tasks are performed using passive tools, such as ropes.

Early occurrences of active manipulation include quadrotor platforms capable of applying forces while maintaining stability [35] and grasping objects while hovering [56]. More recent accomplishments include sliding on surfaces while maintaining normal force [41, 33, 66], autonomous aerial perching and unperching [32], and opening drawers or doors [33, 15]. To achieve these complex aerial manipulation tasks, also more complex aerial platforms such as fully-actuated (Fully-Actuated (FA)) or omnidirectional systems are increasingly used [6, 42, 15].

To accomplish aerial manipulation tasks, accurate and robust control methods are necessary due to the many uncertainties and disturbances inherent in aerial operations. Furthermore aerial manipulation systems are highly nonlinear, which requires nonlinear control methods.

A widely used nonlinear control approach is Nonlinear Dynamic Inversion (NDI), which cancels system nonlinearity through feedback, reducing the control problem to a linear one that can be addressed with conventional linear techniques. However, NDI requires an accurate system model to effectively cancel nonlinearity. This requirement is difficult to satisfy in real systems due to modeling uncertainties, computational errors, and necessary simplifications from reality.

To address these limitations, more robust nonlinear control methods have been developed, including Sliding Mode Control (SMC), backstepping and Lyapunov-based methods [29]. Incremental Nonlinear Dynamic Inversion (INDI) is a more recently developed robust nonlinear control method which is increasingly being used in flight control [52, 61]. INDI is sometimes referred to as a sensor-based controller because it directly uses measurements in its control law to compensate for parts of the model, reducing its dependence on a model. The INDI method has been successfully implemented in different aerospace applications, including fault-tolerant flight control in aircraft [37, 38] and control of multirotor platforms [54, 1, 55].

Although INDI has been implemented widely in quadrotor control it has not yet been used commonly in aerial manipulator control. To the author's knowledge, only in the work of Park et al. [47] INDI is used to control an aerial manipulator in free flight with swift arm motions and in [17] it is used as an inner loop attitude controller for a RL-based outer loop controller for a Differential Shoulder Aerial Manipulator (DSAM). Current state-of-the-art controllers for aerial manipulation include Nonlinear Model Predictive Control (NMPC), which has been applied to sliding and pick-and-place tasks [39, 59, 24]. Also NDI-based approaches remain common [12, 44]. NMPC offers advantages such as its predictive nature and ability to incorporate constraints. In [55] it is shown that NMPC is even able to remain stable while trying to follow dynamically infeasible trajectories. The downsides to NMPC is the large required onboard computational power. Recent advances in computational power however have made NMPC more practical.

Recent work has shown that adding an inner control loop to NMPC improves tracking performance in quadrotor agile flight control [55]. This hierarchical structure combining an outer loop NMPC controller with an inner loop is a promising method to make existing NMPC-formulations more robust and reactive. Another outer loop controller explored in [55] is Differential Flatness Based Control (DFBC). This method has also shown its effectiveness in quadrotor agile flight [20, 57], however this method can only be applied on differentially flat systems. In [64] it is shown how an aerial manipulator system can be differentially flat. For general aerial manipulator configurations this has not been proven and therefore DFBC can not easily be applied to an arbitrary aerial manipulator system.

1-1 Research Goals

Based on the introduction, we notice a lack of application of INDI in control of aerial manipulators, even though it performs well in quadrotor control. This includes many different works in which the superior disturbance rejection properties of INDI are shown compared to PID control [54, 53, 55]. Therefore the main goal of this thesis is to study the use of INDI for aerial manipulator control. For this we build upon results for quadrotor control where NMPC with INDI inner loop has shown great performance in tracking agile trajectories [55]. In this work the system consistently remains stable with this control structure, even while tracking infeasible trajectories. Furthermore NMPC has already shown its effectiveness in aerial manipulators in many different works as mentioned before and therefore will also be used in this work.

There are a number of modifications necessary in order to apply the control method based on [55] to an aerial manipulator. The original quadrotor control problem only requires position

tracking in free flight, but an aerial manipulator also needs to interact with the environment, for example by applying forces. We will design a control approach that tracks end-effector positions and simultaneously a desired force contact force at the end-effector if required. Also another important difference between quadrotor and aerial manipulator control is that the system is now a multibody system consisting of the base and the manipulator. This leads to different dynamics for model-based control and also introduces more design choices, such as decoupled or coupled control as commonly referred to in aerial manipulation literature [9]. This will also be explored in this thesis. Finally, we will investigate the robustness of the controller to external disturbances. Specifically we are interested in robustness to friction forces on the end-effector when sliding. In [55] INDI is applied only for the attitude dynamics of the quadrotor to reject disturbance moments. In this thesis we will also study the effect of using a cascaded INDI structure based on [53] to additionally reject disturbance forces.

We validate this approach on the aerial sliding task, where the manipulator must maintain contact with a surface while rejecting unmodeled disturbances, specifically friction forces at the end-effector. The proposed controller is validated in both simulation and real experiments using a DSAM [17], a quadrotor-based aerial manipulator with a two-degrees-of-freedom (DoF) arm, and compared against an adapted baseline NMPC controller adapted from the work of Sun et al. [55] consisting of a contact-unaware NMPC and INDI inner loop and the work of Tzoumanikas et al. [59] for the contact model. To our knowledge, this is the first study of INDI control for aerial manipulation in contact-based tasks. Lastly we also want to highlight that the aerial manipulator considered in this work consists of an underactuated base (quadrotor) and a simple non compliant 2-DoF arm, showing that no complex omnidirectional base or delta arm is necessary to achieve simultaneous end-effector position, attitude and force tracking.

1-2 Related Work

As we are validating our newly designed controller on the aerial sliding task, we want to compare it to current state-of-the-art achievements in aerial sliding. However note that it is difficult to directly compare different controllers from the obtained results in literature, as the hardware of the aerial manipulators may vary in many ways which influences the obtained results.

A common approach for control of aerial manipulators with underactuated base is to use NMPC [33, 59, 39]. Advantages of using NMPC control is the ability to find solutions given input and state constraints. Furthermore it has also already been extensively studied for quadrotor control [2, 55]. As this work will be mostly focused on showing the first application of the INDI control framework in APhI tasks, these advantages were sufficient for choosing the NMPC as outer loop control. On the other hand, impedance control is also often applied for APhI control for aerial manipulators [66, 48, 25]. It should be noted that these works all utilize an omnidirectional base. In [35] impedance control is applied on an underactuated base, however it is only tested in simulation and in a very limited task of remaining stable while applying a force on a vertical wall. In [13] it is also applied on an underactuated base to perform sliding tasks, however it should be noted that this control scheme consists of three layers. The impedance filter applied on the end-effector is used to calculate inputs to the inverse kinematics of the 6-DoF arm used in this aerial manipulator. The motion controller of the base is separate from this inverse kinematics controller. In our work our two-DoF arm does

not have redundant joints to independently obtain compliant behavior. Compliance should instead should arise from the base motion. From this brief review and some initial testing, Cartesian impedance control seems to be less suited for aerial manipulator control consisting of an underactuated base and a simple two-DoF arm and this strengthens the motivation to use NMPC as outer loop controller for the DSAM.

The task of aerial writing, which is also commonly referred to as push and slide tasks, are commonly found in aerial manipulation literature. In [7] an omnidirectional platform with a rigid end-effector is used for active force and end-effector position tracking using axis-selective impedance control in combination with direct force control. In this work they show the ability to approximately track force references of up to 5 N and position references with centimeter level accuracy for the aerial sliding task. They also achieved force tracking on curved surfaces. In Tzoumanikas et al. [59] an NMPC controller is developed for a quadrotor base with a delta arm and compliant end-effector. In this work they performed the aerial writing tasks with millimeter level end-effector reference tracking precision on a whiteboard in the purely horizontal direction. The contact force tracked for the writing experiments was around 0.2 N. This is a relatively low contact force, which also means relatively low friction forces. Guo et al. [25] used a hybrid motion-force controller for aerial calligraphy. In this work they used an omnidirectional base and also a compliant end-effector. By applying different contact forces, different line-widths can be achieved with this custom built end-effector. The sliding task was performed on a vertical wall and with a prior measured friction coefficient. This lead to end-effector position tracking RMSE of around 2.5 cm for their experiments. In Zhang et al. [66] an impedance controller with learned adaptive stiffness is used for sliding on heterogeneous surfaces. Their contribution is mainly using a reinforcement learning policy for adaptive stiffness in the impedance controller. Their system uses an omnidirectional base with a rigid end-effector. This means that they do not have to account for underactuation, which characterizes quadrotor-based aerial manipulators such as the one in this work.

A key design choice in aerial manipulator control is how to model the interaction between the manipulator and the aerial base. In some works mentioned before a rigid end-effector is used for aerial manipulation and the system can be modeled as a single rigid body. If the end-effector is not rigidly attached to the base, multibody dynamics need to be considered. In many works, the arm and base are treated as decoupled systems, where the coupling between them is ignored [35]. This simplification allows modular use of control methods developed for the base or the manipulator. However, ignoring coupling requires very restricting assumptions on the morphology of the aerial manipulator and tasks that can be performed.

When the manipulator's weight is non-negligible or moves rapidly, the dynamic interaction forces between the platform and base cannot be ignored. Most literature assumes quasi-static coupling, where these forces are treated as slowly varying or constant [9]. However, for tasks requiring rapid or forceful arm motions, such as sliding with unmodeled friction forces, a more complete model of the coupled system is necessary. Bonyan Khamseh et al. [9] provides a review of coupled versus decoupled control approaches for aerial manipulators. The challenge is that a complete coupled model is difficult to obtain accurately and may not transfer well to physical systems due to many unmodeled real-world effects. Therefore, different levels of modeling the coupling represent trade-offs between model fidelity and practical robustness. Also many applications use a decoupled control approach where an estimate of the dynamic interaction forces are used to improve end-effector reference tracking and disturbance rejection [8, 12]. For the INDI control method, the decoupled approach has also been explored in the

work of Park et al. [47], where they developed an adaptive INDI controller for the base to handle swift arm motions. In their work a decoupled approach is proposed and INDI is used to reject the unmodeled interaction between the arm and base.

1-3 Outline

This work is structured as follows: In Chapter 2 the aerial manipulator considered in this work is introduced. Preliminaries on multi-body modeling will be presented and applied to the aerial manipulator system. In Chapter 3 the relevant nonlinear control methods are introduced for the controller design. In Chapter 4 the proposed controllers in this work are introduced. The controllers consist of a novel contact-aware NMPC formulation and a INDI-based inner loop. In Chapter 5 results are presented from simulation and real world experiments, including the aerial sliding task in different world directions and different surfaces. Finally in Chapter 6, some conclusions are drawn on applying INDI in aerial manipulation and future research directions are discussed.

Chapter 2

Preliminaries: Modeling of the DSAM

In this chapter a kinematic and dynamic model of the Differential Shoulder Aerial Manipulator (DSAM) system will be derived. The DSAM is developed at the Mobile Robotics Lab (MRL) at the TU Delft. The aerial manipulator consists of a quadrotor platform and a 2-degrees-of-freedom (DoF) manipulator. This manipulator is a single link connected to the aerial base using a differential shoulder joint that can be modeled as two consecutive rotations, first around the pitch-axis and then around the rotated roll-axis. On this link a gripper can be attached or a pen as is done in this work. The system has 8 DoF, and has 6 inputs. First we will choose a set of generalized coordinates and velocities to describe our system and derive the corresponding kinematics. After that we will derive the dynamics using the Projected-Newton Euler Method [51], which is a systematic way of deriving constraint-consistent multibody dynamics. We will model the aerial manipulator as a floating base system, with the quadrotor and the arm as separate rigid body. In the dynamic model we will use the force and moments generated by the propellers as input. In multi-rotor control it is then also necessary to map the forces and moments to propeller speeds. This is called control allocation and will also be derived for the DSAM. Lastly as an aerial manipulator is designed to interact with the environment, also contact modeling will be introduced. This modeling is then used for simulation, but also to design part of the controller.

2-1 Kinematic Model

In this section, we derive the kinematic model of the DSAM. We define the following reference frames: the inertial frame Σ_I, the quadrotor base (body) frame Σ_B, the manipulator root frame Σ_R (rigidly attached to the quadrotor), and the end-effector frame Σ_E. A schematic overview of these reference frames is depicted in Figure 2-1.

2-1-1 Notation and Coordinates

We adopt the following notation: a general vector is denoted as ${}_A\boldsymbol{p}_{bc}$, where Σ_A is the reference frame in which the vector is expressed, b is the origin, and c is the target point. A rotation

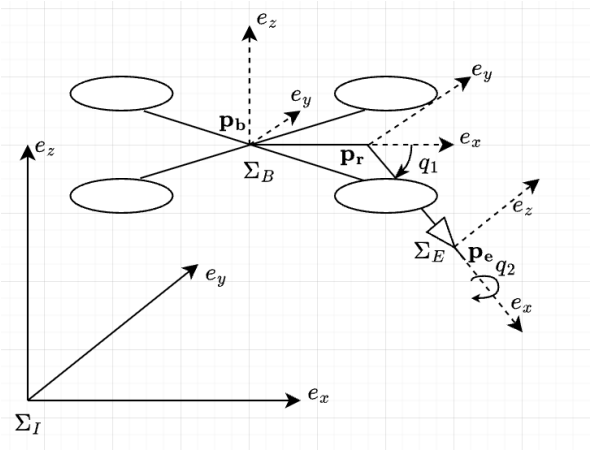


Figure 2-1: Schematic representation of the DSAM and relevant reference frames.

matrix $R_{AB} \in \mathrm{SO}(3)$ transforms a vector from frame Σ_B to frame Σ_A, such that ${}_A\boldsymbol{p} = R_{AB}\,{}_B\boldsymbol{p}$. Vectors without a prescript are assumed to be expressed in the inertial frame Σ_I.

The quadrotor platform acts as a floating base with 6 DoF. While Euler angles are often used to parametrize attitude, they suffer from gimbal lock singularities [45]. To avoid this, we use unit quaternions to parametrize the attitude of the base. Consequently, the generalized coordinates are defined by the base position $\boldsymbol{p}_b = \boldsymbol{p}_{IB} \in \mathbb{R}^3$, the base orientation quaternion $\boldsymbol{q}_b = [q_w, q_x, q_y, q_z]^T \in \mathbb{H}$ (where $\|\boldsymbol{q}_b\| = 1$), and the joint angles $\boldsymbol{\eta} = [\eta_1, \eta_2]^T \in \mathbb{R}^2$ of the manipulator. The generalized coordinate vector is:

$$\boldsymbol{q} = [\boldsymbol{p}_b^T,\ \boldsymbol{q}_b^T,\ \boldsymbol{\eta}^T]^T \in \mathbb{R}^9 \tag{2-1}$$

For the generalized velocities $\boldsymbol{v}$, we select the inertial linear velocity of the base $\boldsymbol{v}_b$, the angular velocity of the base expressed in the body frame ${}_B\boldsymbol{\omega}_b$, and the joint velocities $\dot{\boldsymbol{\eta}}$. This yields the generalized velocity vector:

$$\boldsymbol{v} = [\boldsymbol{v}_b^T,\ {}_B\boldsymbol{\omega}_b^T,\ \dot{\boldsymbol{\eta}}^T]^T \in \mathbb{R}^8 \tag{2-2}$$

2-1-2 Differential Kinematics

The generalized velocities $\boldsymbol{v}$ are related to the time derivative of the generalized coordinates $\dot{\boldsymbol{q}}$ via a transformation matrix $E(\boldsymbol{q})$ which is usually the identity matrix. However as we are taking the quaternion parametrization of the orientation of the base in our generalized coordinates and the angular rates as generalized velocity, this mapping is not trivial anymore. The linear velocity $\boldsymbol{v}_b$ corresponds directly to $\dot{\boldsymbol{p}}_b$. The relationship between the body angular velocity and the quaternion derivative is given by $\dot{\boldsymbol{q}}_b = \frac{1}{2}\Lambda(\boldsymbol{q}_b)\,{}_B\boldsymbol{\omega}_b$, where $\Lambda(\boldsymbol{q}_b)$ is the standard quaternion propagation matrix [51]. The mapping between joint velocities and angles are trivial so the relation between time derivative of the generalized coordinates and generalized velocities are given as follows:

$$\dot{\boldsymbol{q}} = E(\boldsymbol{q})\boldsymbol{v} = \mathrm{blkdiag}\left(I_{3\times3}, \frac{1}{2}\Lambda(\boldsymbol{q}_b), I_{2\times2}\right)\boldsymbol{v} \tag{2-3}$$

2-1-3 Multibody Kinematics

To derive the equations of motion using the Projected Newton-Euler Method (PNEM), the absolute poses and velocities of specific points of interest are required. We model the DSAM as two rigid bodies connected by a 2-DoF joint at the manipulator base. The points of interest are the Center of Gravity (CoG) of the two bodies, namely the quadrotor base and the manipulator, and the end-effector (where contact forces are applied).

We utilize Homogeneous Transformation matrices $T \in \mathrm{SE}(3)$ to describe the kinematics. A transformation from frame B to A is defined as:

$$T_{AB} = \begin{pmatrix} R_{AB} & {}_A\boldsymbol{p}_{AB} \\ \mathbf{0}^T & 1 \end{pmatrix} \tag{2-4}$$

where ${}_A\boldsymbol{p}_{AB}$ is the position vector pointing from A to B expressed in frame Σ_A. The kinematic chain of the DSAM end-effector consists of the floating base transformation $T_{IB}(\boldsymbol{p}_b, \boldsymbol{q}_b)$, the fixed transformation T_{BR} to the manipulator root, the joint-dependent transformation for the arm T_{RA} and finally the fixed transform to the end-effector T_{AE}. The global pose of the end-effector is obtained by the chain:

$$T_{IE}(\boldsymbol{q}) = T_{IB}(\boldsymbol{q}_b) T_{BR} T_{MA}(\eta_1, \eta_2) T_{AE} \tag{2-5}$$

From this matrix, the position $\boldsymbol{p}_e$ and rotation matrix R_{IE} corresponding to the orientation of the end-effector can be directly extracted. As we choose quaternions as parametrization of the orientation we can convert the rotation matrix R_{IE} to quaternions and combine the absolute position and orientation in geometric coordinate vector $\boldsymbol{x}_e = \boldsymbol{k}(\boldsymbol{q})$.

Similarly, the positions of the arm CoG is found by evaluating the transformation chain up to the arm. The inertial velocities of the CoG of the arm and base are essential for dynamic model derivation. These are related to the generalized velocity $\boldsymbol{v}$ via geometric Jacobians. For a point of interest i the velocity is:

$$\begin{pmatrix} \boldsymbol{v}_i \\ \boldsymbol{\omega}_i \end{pmatrix} = J_i(\boldsymbol{q}) \boldsymbol{v} \tag{2-6}$$

where $J_i(\boldsymbol{q}) \in \mathbb{R}^{6\times 8}$ is the geometric Jacobian matrix associated with point i. These Jacobian map the floating base velocities and joint rates to the inertial velocities of each of point of interest. It also common to split up the geometric Jacobian in a translational Jacobian and rotational Jacobian:

$$\boldsymbol{v}_i = J_{P,i}(\boldsymbol{q}) \boldsymbol{v} \tag{2-7}$$

and

$$\boldsymbol{\omega}_i = J_{R,i}(\boldsymbol{q}) \boldsymbol{v} \tag{2-8}$$

The translational Jacobian of a point i can be derived by taking the time derivative of the absolute position vector $\boldsymbol{p}_i$, which is the inertial velocity $\boldsymbol{v}_i$ and then taking the partial derivative to the chosen generalized coordinates:

$$J_{P,i}(\boldsymbol{q}) = \frac{\partial \dot{\boldsymbol{p}}_i}{\partial \boldsymbol{v}} \tag{2-9}$$

The rotational Jacobian of a point i is based on the reference frame of point i. Unlike velocities, angular rates relative to each other can be summed if expressed in the same reference frame and thus the inertial angular rate can easily be obtained by adding up successive angular rates. Again by taking the partial derivative to the generalized coordinates we can obtain the rotational Jacobian:

$$J_{R,i}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{v}} \tag{2-10}$$

For the next section we will leave out the explicit notation that the Jacobian are dependent on the generalized coordinates $\boldsymbol{q}$. An analytical derivation of the Jacobian of the DSAM are given in the appendix.

2-2 Dynamic Model

In this section, the derivation of the dynamic model for the DSAM is presented using the PNEM. This method combines the Newton-Euler equations of the individual rigid bodies with the kinematic constraints derived in Section 2-1 to obtain the Equations of Motion (EoM) in the generalized coordinate space.

The system dynamics of the DSAM can be written in the standard canonical form:

$$M(\boldsymbol{q})\dot{\boldsymbol{v}} + C(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \tag{2-11}$$

where $M(\boldsymbol{q}) \in \mathbb{R}^{8\times8}$ is the generalized mass matrix, $C(\boldsymbol{q}, \boldsymbol{v}) \in \mathbb{R}^{8\times8}$ is the Coriolis and centrifugal matrix, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^8$ is the vector of gravitational forces, $\boldsymbol{\tau}$ is the vector of actuation torques/forces, and $\boldsymbol{\tau}_{ext}$ represents external wrenches. How the actuation torques and forces and external wrenches are related to the propeller speeds and interaction with the environment will be explored in the next sections.

We consider the system as a combination of two rigid bodies $i \in \{B, A\}$, representing the quadrotor base and the arm link. For each body i, we have the mass m_i and the inertia tensor $\mathcal{I}_i$ expressed in the body's local frame.

The dynamics of the coupled system can be obtained using the PNEM [51] by projecting the unconstrained Newton-Euler equations of each link onto the generalized coordinates using the geometric Jacobian derived in Appendix A. Using the PNEM we can derive the dynamics in mixed frames to simplify the equations. The geoetric translational Jacobian of all bodies should be defined in the same reference frame and the geometric rotational Jacobian of all bodies should also all be defined in the same reference frame. With this method each of the matrices of the canonical EoM can be obtained in a systematic way, which is next introduced.

The generalized mass matrix is calculated by summing the masses and inertia of all bodies, mapped to the generalized coordinates using their respective Jacobian in their respective reference frame. For the DSAM system we will choose the world frame Σ_I to express the inertial translational velocity and the base frame Σ_B to express the inertial angular rates. We obtain the following mass matrix:

$$M(\boldsymbol{q}) = \sum_{i\in\{B,A\}} \left({}_I J_{P,i}^T \cdot m_i \cdot {}_I J_{P,i} + {}_B J_{R,i}^T \cdot {}_B \mathcal{I}_i \cdot {}_B J_{R,i} \right) \tag{2-12}$$

In Appendix A an analytical expression is derived for the generalized mass matrix of the DSAM.

The Coriolis and centrifugal terms arise from the changing effective inertia and the rotation of the reference frames. In the projected method, this is derived from the time derivative of the body and arm Jacobian and the gyroscopic moments:

$$C(\boldsymbol{q},\boldsymbol{v}) = \sum_{i\in\{B,A\}} \Big({}_I J_{P,i}^T \cdot m_i \cdot {}_I \dot{J}_{P,i} \cdot \boldsymbol{v} + {}_B J_{R,i}^T \Big({}_B \mathcal{I}_i \cdot {}_B \dot{J}_{R,i} \cdot \boldsymbol{v} + {}_B \boldsymbol{\Omega}_i \times {}_B \mathcal{I}_i \cdot {}_B \boldsymbol{\Omega}_i \Big) \Big) \tag{2-13}$$

where ${}_B\boldsymbol{\Omega}_i$ is the inertial angular velocity of body i expressed in base frame Σ_B.

The generalized gravity vector is derived by projecting the gravitational acceleration vector $\boldsymbol{g}_I = [0, 0, -g]^T$ acting on each mass into the generalized coordinate space:

$$\boldsymbol{g}(\boldsymbol{q}) = - \sum_{i\in\{B,A\}} {}_I J_{P,i}^T \cdot m \cdot \boldsymbol{g}_I \tag{2-14}$$

In this framework forces due to actuators are incorporated in the same way as other external forces. The sources of external forces and moments that are relevant in the dynamical description in this section are the body torques ${}_B\boldsymbol{\tau}_b$, the total thrust ${}_B\boldsymbol{T}$ and the actuator torques of the two joints τ_{η_1} and τ_{η_2}. Each of these forces and moments can be projected to the generalized coordinate space by using the translational and rotational Jacobian of the point or body on which they act. In the DSAM system there are actuators that move the arm relative to the base. So the moments due to the actuation are applied to both the base and the arm.

The generalized force vector for the DSAM is defined as follows:

$$\boldsymbol{\tau} = \begin{pmatrix} R_{IB}\, {}_B\boldsymbol{T} \\ \boldsymbol{\tau}_b \\ \boldsymbol{\tau}_\eta \end{pmatrix} \tag{2-15}$$

Note that the first three entries of the generalized force vector is the thrust. Due to the underactuated nature of the quadrotor, this thrust force can not be freely controlled and this needs to be considered in controller design for such systems. Also the actual inputs of the DSAM are motor speeds of the propellers and desired joint positions, not the generalized forces. These can however be related to each other via control allocation, which will be introduced in Section 2-4.

2-3 Aerodynamic Forces

In the previous section we have derived the dynamics of a floating base multi-body system given external forces and torques. In this section we will dive deeper into what forces and torques are relevant for the DSAM. Furthermore also the relation between propeller speeds and the generalized forces are explored.

To properly model our aerial manipulator it is essential to understand the aerodynamics of a multi-rotor. In this work, the aerodynamics are based on [40]. The most important force in multi-rotor dynamics is the thrust force generated by the propellers. A simple model for the generated thrust force is given as:

$$T_i = b_i \omega_{p,i}^2 \tag{2-16}$$

in which b_i is the thrust coefficient and $\omega_{p,i}$ is the angular velocity of propeller i. This thrust force is only applied in the rotation direction of the propeller. For standard non tilted multi-rotors, this direction is ${}_B\boldsymbol{e}_z$ for all rotors, while for tiltable multi-rotors the direction of the thrust of each propeller can be controlled. Each propeller also generates a drag force, which can be modeled as:

$$Q_i = d_i \omega_{p,i}^2 \tag{2-17}$$

in which d_i is the drag coefficient and $\omega_{p,i}$ is the angular velocity of propeller i. This drag force is applied in the direction which resists the rotation of the propeller, which results in a torque on the multi-rotor body opposing the rotation speed of the propeller. In a standard quadrotor configuration, two propellers rotate in opposite directions compared to the other two, so no moment is generated in total if all four propellers rotate at the same velocity which should occur during hover. Furthermore this will lead to a decoupled heading control from the rest of the quadrotor dynamics.

Commonly only these main aerodynamic forces are considered for control, especially when aerial platforms operate near hover with low velocities. The relations between propeller speeds and forces are approximations which can be identified using static tests. Better models of the dynamics include many other effects such as blade flapping, induced drag [40] and rotor interaction forces [34]. Also it might be possible that dynamics and parameters change during operation. For example the aerodynamics of the platform changes when flying closer to a surface [49] or damage to the propeller blades could also influence the blade aerodynamic properties. Lastly, especially for relatively high-speed motions, drag force is also typically considered. These effects can significantly influence the mapping from propeller speeds to the generalized forces of the dynamic system. It is practically impossible to model all the dynamics of an aerial manipulator accurately for a real flight and this suggests the use of robust control methods for multi-rotors and thus also for aerial manipulators. Furthermore common disturbances in uncontrolled environments such as wind gusts have a significant impact on the dynamics of a multi-rotor. Therefore disturbance rejection is also an important property for control in aerial manipulation.

2-4 Control Allocation

In the previous sections the EoM of the aerial manipulation system is described with generalized forces, which are based on joint torques, body torques and forces. Also the main contributions in the relation between propeller speeds and these generalized forces were summarized. For control of any multi-rotor system, the body forces and torques need to be mapped to rotor speeds. The relation between the propeller speeds and the body forces and torques of a multi-rotor is known as control allocation. In our work the DSAM has reconfigurable rotors, where each rotor can be tilted around the body x-axis. Also the front rotors are different from the back rotors, so the ordering of the rotors is of importance. Define the the first rotor to be

the front left rotor that rotates Counterclockwise (CCW), the second rotor to be the back right rotor which also rotates CCW, the third rotor to be the back left rotor that rotates Clockwise (CW) and finally the fourth rotor to be the front right rotor that also rotates CW. Define n_i to be 1 for the CCW propellers and -1 for the CW propellers to account for the rotation direction. Next define angle θ_i as the rotation angle of propeller i around the body x-axis. Then the thrust vector of propeller i denoted in the body frame B is calculated as:

$$ {}_B\boldsymbol{T}_i = R_{BP_i}\, {}_{P_i}\boldsymbol{T}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_i & \sin\theta_i \\ 0 & -\sin\theta_i & \cos\theta_i \end{bmatrix} b_i w_{p,i}^2 \boldsymbol{e}_3 \tag{2-18} $$

and the drag moment of propeller i is calculated as:

$$ {}_B\boldsymbol{Q}_i = n_i R_{BP_i}\, {}_{P_i}\boldsymbol{Q}_i = n_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_i & \sin\theta_i \\ 0 & -\sin\theta_i & \cos\theta_i \end{bmatrix} d_i w_{p,i}^2 \boldsymbol{e}_3 \tag{2-19} $$

Control allocation considering only propeller thrust and drag then takes the form of:

$$ \begin{pmatrix} {}_B\boldsymbol{\tau} \\ {}_B\boldsymbol{F} \end{pmatrix} = \boldsymbol{G}_1 \boldsymbol{\omega}_p \circ \boldsymbol{\omega}_p \tag{2-20} $$

with $\circ$ the Hadamard product and

$$ \boldsymbol{G}_1 = \begin{bmatrix} n_1 R_{BP_1} d_1 \boldsymbol{e}_3 & n_2 R_{BP_2} d_2 \boldsymbol{e}_3 & n_3 R_{BP_3} d_3 \boldsymbol{e}_3 & n_4 R_{BP_4} d_4 \boldsymbol{e}_3 \\ R_{BP_1} b_1 \boldsymbol{e}_3 & R_{BP_2} b_2 \boldsymbol{e}_3 & R_{BP_3} b_3 \boldsymbol{e}_3 & R_{BP_4} b_4 \boldsymbol{e}_3 \end{bmatrix} \in \mathbb{R}^{6\times 4}, \quad \boldsymbol{\omega}_p = \begin{pmatrix} \omega_{p,1} \\ \omega_{p,2} \\ \omega_{p,3} \\ \omega_{p,4} \end{pmatrix} \tag{2-21} $$

in which d_i is the drag coefficient and b_i is the thrust coefficient of propeller i. Note that only four inputs are possible, so to invert matrix $\boldsymbol{G}_1$, either a pseudo-inverse needs to be used, some quadratic optimization on the propeller speeds or we restrict the force and torques we try to allocate to only consider the force in the body z-direction. We will take the last approach in this work and define the actual the reduced control allocation matrix

$$ \tilde{\boldsymbol{G}}_1 = \begin{bmatrix} n_1 R_{BP_1} d_1 \boldsymbol{e}_3 & n_2 R_{BP_2} d_2 \boldsymbol{e}_3 & n_3 R_{BP_3} d_3 \boldsymbol{e}_3 & n_4 R_{BP_4} d_4 \boldsymbol{e}_3 \\ \boldsymbol{e}_3^T R_{BP_1} b_1 \boldsymbol{e}_3 & \boldsymbol{e}_3^T R_{BP_2} b_2 \boldsymbol{e}_3 & \boldsymbol{e}_3^T R_{BP_3} b_3 \boldsymbol{e}_3 & \boldsymbol{e}_3^T R_{BP_4} b_4 \boldsymbol{e}_3 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{2-22} $$

such that

$$ \boldsymbol{\omega}_p \circ \boldsymbol{\omega}_p = \tilde{\boldsymbol{G}}_1^{-1} \begin{pmatrix} {}_B\boldsymbol{\tau} \\ || \, {}_B\boldsymbol{F} || \end{pmatrix} \tag{2-23} $$

For now we have only considered the rotor thrust and drag for the control allocation. In literature it has also been studied to include other effects, such as the rotor dynamics and gyroscopic forces of the propellers. In [54] these effects are added and summarized in additional matrices $\boldsymbol{G}_2$ corresponding to the propeller dynamics and $\boldsymbol{G}_3$ corresponding to the gyroscopic terms. It has been found that neglecting the gyroscopic terms $\boldsymbol{G}_3$ has little influence to the control performance of the multi-rotor in [54] using Incremental Nonlinear Dynamic Inversion (INDI), so this will also be neglected in this work. However neglecting the $\boldsymbol{G}_2$ matrix

related to the spin-up torque seemed to greatly influence the INDI control performance. For that reason this term will also be included in this work. However in this work the matrix was derived for a standard multi-rotor with non tilted rotors. For this work we will have to derive the $\boldsymbol{G}_2$ matrix with tilted rotors.

The inertial torques of each rotor can be summarized as

$$\mathcal{I}_{p,i}\dot{\omega}_{p,i}\,{}_{P_i}\boldsymbol{e}_z = n_i\tau_{p,i}\,{}_{P_i}\boldsymbol{e}_z \tag{2-24}$$

where $\mathcal{I}_{p,i}$ is the rotor inertia tensor in the rotor frame and ${}_{P_i}\boldsymbol{e}_z$ is aligned with the thrust direction of the propellor. Again we want to know the torques applied in the body frame B so we write the following relation for the inertial torque of propeller i:

$${}_B\boldsymbol{\tau}_{\text{in}} = \mathcal{I}_{p,i}R_{BP_i}\dot{\omega}_{p,i}\boldsymbol{e}_3 \tag{2-25}$$

which leads to the following $\boldsymbol{G}_2$ matrix:

$$\boldsymbol{G}_2 = \begin{bmatrix} n_1\mathcal{I}_{p,1}R_{BP_1}\boldsymbol{e}_3 & n_2\mathcal{I}_{p,2}R_{BP_2}\boldsymbol{e}_3 & n_3\mathcal{I}_{p,3}R_{BP_3}\boldsymbol{e}_3 & n_4\mathcal{I}_{p,4}R_{BP_4}\boldsymbol{e}_3 \\ \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times1} \end{bmatrix} \in \mathbb{R}^{6\times4} \tag{2-26}$$

Now the full nonlinear mapping from propeller speeds and its derivative to body forces and torques is given as

$$\begin{pmatrix} {}_B\boldsymbol{\tau} \\ {}_B\boldsymbol{F} \end{pmatrix} = \boldsymbol{G}_1\boldsymbol{\omega}\circ\boldsymbol{\omega} + \boldsymbol{G}_2\dot{\boldsymbol{\omega}} \tag{2-27}$$

We also define the reduced $\boldsymbol{G}_2$ matrix as:

$$\tilde{\boldsymbol{G}}_2 = \begin{bmatrix} \mathcal{I}_{p,1}R_{BP_1}\boldsymbol{e}_3 & \mathcal{I}_{p,2}R_{BP_2}\boldsymbol{e}_3 & \mathcal{I}_{p,3}R_{BP_3}\boldsymbol{e}_3 & \mathcal{I}_{p,4}R_{BP_4}\boldsymbol{e}_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \tag{2-28}$$

such that we can approximate the mapping as:

$$\begin{pmatrix} {}_B\boldsymbol{\tau} \\ ||\,{}_B\boldsymbol{F}|| \end{pmatrix} = \tilde{\boldsymbol{G}}_1\boldsymbol{\omega}\circ\boldsymbol{\omega} + \tilde{\boldsymbol{G}}_2\dot{\boldsymbol{\omega}} \tag{2-29}$$

2-5 Contact Modeling

Aerial manipulators are specifically designed to interact with their environment, which introduces unique challenges in both simulation and control. Unlike traditional aerial vehicles, these systems must maintain stability while exerting or receiving forces through physical contact at the end-effector during Aerial Physical Interaction (APhI) tasks. In this section we will discuss how contact is incorporated in both control models and simulation.

There are two primary approaches to modeling contact: hard contact models and soft contact models. In the soft contact model, interactions are represented through continuous contact forces. The most common and easiest soft contact model is by modeling the environment as a mass-spring-damper system. This is done in impedance control and admittance control and is commonly applied in robotics [51]. For aerial manipulation the soft contact model is also often applied, including in impedance controllers [48, 44, 66] and in Nonlinear Model

Predictive Control (NMPC) where the contact model is explicitly included in the dynamics [59]. The main challenge with soft contact models in simulation is that the combined differential equations of the multibody system and contact dynamics of the simulation can become highly stiff [21]. This stiffness leads to either poor numerical accuracy or long computational times when solving the differential equations.

The hard contact model treats contact as a kinematic constraint rather than a force-displacement relationship. Instead of computing contact forces based on penetration depth at the contact point, forces are calculated based on a kinematic constraint related to the contact with a contact Jacobian. This can be easily included in optimization based control methods by including this kinematic contact constraint [46]. This approach can offer computational advantages especially in simulation but requires careful handling of contact transitions and impulses as it introduces non continuity in the dynamics.

Contact modeling is crucial not only in the control algorithms but also for achieving realistic simulation behavior. Accurately modeling physical contact phenomena such as friction, impact and compliance present significant challenges, and simplifications are commonly done in practice. Simplifications however introduce discrepancies between simulated and real-world behavior and a balance has to be found between computational efficiency and realistic contact behavior. Many physics engines and simulation environments support contact modeling for robotic systems. In this work we consider Gazebo [30] for implementation of the final flight stack and MATLAB SimScape [58] for controller prototyping. Gazebo provides a comprehensive robotics simulation stack with ROS integration, but lacks contact realism. Controller implementation and modeling in SimScape is easier, but there is larger gap from the controller implementation in this software and the actual flight stack.

As we utilize MATLAB SimScape for controller prototyping for APhI tasks, the following part will describe SimScape's approach to contact modeling. For contact between two 3D rigid bodies, it will be assumed to occur at a single contact point with an associated reference frame Σ_C. According to Newton's third law, equal and opposite forces are applied to both bodies, which consists of a normal contact force and a friction force.

Let d denote the penetration depth of the two bodies at the contact point. The normal contact force is applied along the z-direction of the contact frame and opposes the penetration direction. The friction force acts within the contact xy-plane and opposes the relative tangential velocity between the bodies.

The normal force in SimScape is modeled as:

$$ {}_C\boldsymbol{F}_n = s(d, w) \left(k_P d + k_D \dot{d} \right) {}_C\boldsymbol{e}_z \tag{2-30} $$

where w is the transition width parameter, $s(d, w)$ is a smoothing function for the contact force, k_P and k_D are the contact stiffness and damping coefficients, respectively, and $\dot{d}$ is the rate of change of penetration depth. SimScape employs a soft contact model, where the smoothing function $s(d, w)$ is introduced to prevent discontinuities in the system dynamics. This function satisfies: $s(0) = 0$, $s(w) = 1$, and increases monotonically and continuously on the interval $[0, w]$. The transition width parameter is also introduced to prevent numerical issues. To simulate a rigid surface, high contact stiffness gains and a small transition width must be chosen. A balance must therefore be found between solver performance and realistic rigid body contact.

Next, the friction force is calculated according to the smooth stick-slip method. In this model, the friction force acts opposite to the relative tangential velocity at the contact point, with a magnitude that depends on the effective friction coefficient μ_{eff}. This coefficient transitions smoothly between the static coefficient μ_s and the dynamic coefficient μ_d as a function of the tangential slip velocity:

$$\mu_{\text{eff}}(v) = \mu_d + (\mu_s - \mu_d)\, s(v_{\text{crit}},\, v) \tag{2-31}$$

where $v = |\,{}_C\boldsymbol{v}_{\parallel}|$ is the magnitude of the relative tangential velocity at the contact point, v_{crit} is the critical velocity threshold, and $s(v_{\text{crit}}, v)$ is a smoothing function that satisfies $s(v_{\text{crit}}, 0) = 1$ and $s(v_{\text{crit}}, v) \to 0$ for $v \gg v_{\text{crit}}$. The resulting friction force vector in the contact plane is then given by:

$${}_C\boldsymbol{F}_f = -\mu_{\text{eff}}\Big(|\,{}_C\boldsymbol{v}_{\parallel}|\Big)\ |{}_C\boldsymbol{F}_n|\ \frac{{}_C\boldsymbol{v}_{\parallel}}{\max\Big(|\,{}_C\boldsymbol{v}_{\parallel}|,\, \epsilon\Big)} \tag{2-32}$$

where ${}_C\boldsymbol{v}_{\parallel}$ is the relative tangential velocity vector at the contact point in the contact plane, $|{}_C\boldsymbol{F}_n|$ is the normal force magnitude, and ϵ is a small regularization constant to avoid division by zero. This formulation ensures a smooth transition between the sticking regime, where $\mu_{\text{eff}} \approx \mu_s$, and the sliding regime, where $\mu_{\text{eff}} \approx \mu_d$, avoiding discontinuities that could destabilize the numerical integration.

2-6 Summary

In this Chapter the kinematic and dynamic modeling of the DSAM is presented. The different relevant aerodynamic forces for the platform are introduced and the control allocation is derived. Lastly a brief introduction is given into contact modeling. Specifically the SimScape Multibody [58] contact modeling is explored in detail, as this simulator is used for prototyping the controller design proposed in this work. Using the models and relations derived in this Chapter it is possible to design model-based controllers for the platform to perform APhI tasks. In the next Chapter the nonlinear control methods considered in this thesis are formally introduced.

Chapter 3

Preliminaries: Nonlinear Control

In this Chapter the nonlinear control methods that are applied in the proposed controller designs of this thesis are introduced. First a theoretical introduction to Nonlinear Dynamic Inversion (NDI) and Incremental Nonlinear Dynamic Inversion (INDI) is given and the advantages and disadvantages of these methods. As we are interested in robust disturbance rejection properties, disturbance estimation used in nonlinear control is introduced and this will allow for another view on the relation between NDI and INDI. Finally Nonlinear Model Predictive Control (NMPC) is also introduced.

3-1 Nonlinear Dynamic Inversion

Consider a class of Single-Input Single-Output (SISO) nonlinear systems of the form:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})u \tag{3-1}$$
$$y = h(\boldsymbol{x}) \tag{3-2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and the mappings f, g and h are sufficiently smooth in a domain $D \subset \mathbb{R}^n$. If there exists a diffeomorphism $T : D \to \mathbb{R}^n$ and the change of variables $\boldsymbol{z} = T(\boldsymbol{x})$ transforms system (3-1)-(3-2) into the form:

$$\boldsymbol{z} = A\boldsymbol{z} + B\beta(\boldsymbol{x})(u - \alpha(\boldsymbol{x})) \tag{3-3}$$

then under certain conditions [29] the system is feedback linearizable and NDI can be applied by choosing $u = \beta(\boldsymbol{x})^{-1}\nu + \alpha(\boldsymbol{x})$. This method is referred to as full-state linearization and transforms a nonlinear system to a linear one:

$$\boldsymbol{z} = A\boldsymbol{z} + B\nu \tag{3-4}$$

in which ν the virtual control input can be chosen as $\nu = -K\boldsymbol{z}$ with K such that $A - BK$ is Hurwitz. Applying this diffeomorphism however does not mean that the output description $y = h(T(\boldsymbol{z}))$ is a linear equation. If the output equation is nonlinear, reference tracking is

still difficult to achieve. Under certain conditions the input-output relation can be linearized with feedback which is called input-ouput linearization. For this we look at the derivatives of the output and introduce the relative degree. We start by introducing some notation. The derivative of the output can be written as

$$\dot{y} = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}}[f(\boldsymbol{x}) + g(\boldsymbol{x})u] = L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})u \tag{3-5}$$

where $L_f h(\boldsymbol{x})$ and $L_g h(\boldsymbol{x})$ are the Lie derivatives of h with respect to vector field f of g respectively. The following other notations will be used:

$$L_g L_f h(\boldsymbol{x}) = \frac{\partial (L_f h)}{\partial \boldsymbol{x}} g(\boldsymbol{x}) \tag{3-6}$$

$$L_f^2 h(\boldsymbol{x}) = L_f L_f h(\boldsymbol{x}) = \frac{\partial (L_f h)}{\partial \boldsymbol{x}} f(\boldsymbol{x}) \tag{3-7}$$

$$L_f^k h(\boldsymbol{x}) = L_f L_f^{k-1} h(\boldsymbol{x}) = \frac{\partial (L_f^{k-1} h)}{\partial \boldsymbol{x}} f(\boldsymbol{x}) \tag{3-8}$$

$$L_f^0 h(\boldsymbol{x}) = h(\boldsymbol{x}) \tag{3-9}$$

If $L_g h(\boldsymbol{x}) = 0$, then $\dot{y}$ is independent of the control input u. By continuing with the next derivative,

$$y^{(2)} = \frac{\partial (L_f h)}{\partial \boldsymbol{x}}[f(\boldsymbol{x}) + g(\boldsymbol{x})u] = L_f^2 h(\boldsymbol{x}) + L_g L_f h(\boldsymbol{x})u \tag{3-10}$$

if $L_g L_f h(x) = 0$, then the second derivative of the output is also independent of the control. By repeating this process if at some point:

$$L_g L_f^{i-1} h(x) = 0, \quad i = 1, 2, \ldots, \rho - 1 \tag{3-11}$$

and

$$L_g L_f^{\rho-1} h(\boldsymbol{x}) \neq 0 \tag{3-12}$$

then we know that u does not appear in the expressions of $y, \dot{y}, \ldots, y^{(\rho-1)}$, but does appear in $y^{(\rho)}$ with a nonzero coefficient:

$$y^{(\rho)} = L_f^\rho h(\boldsymbol{x}) + L_g L_f^{\rho-1} h(\boldsymbol{x})u \tag{3-13}$$

The integer ρ is called the *relative degree* and the input u can chosen as

$$u = \frac{\nu - L_f^\rho h(\boldsymbol{x})}{L_g L_f^{\rho-1} h(\boldsymbol{x})} \tag{3-14}$$

such that the resulting system becomes:

$$y^{(\rho)} = \nu \tag{3-15}$$

which is a chain of ρ integrators, with ν the virtual control input. Note that we have a direct relation between the output and the virtual input, therefore this form is called input-output linearization.

We can extend this idea to Multiple-Input Multiple-Output (MIMO) systems. The system is now of the form:

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x})\end{aligned} \tag{3-16}$$

in which $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^p$. The function $\boldsymbol{f}$ is a smooth mapping $\mathbb{R}^n \to \mathbb{R}^n$, $\boldsymbol{g}$ a smooth mapping $\mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^n$ and $\boldsymbol{h}$ a smooth mapping $\mathbb{R}^n \to \mathbb{R}^p$. If $p < m$, so there are more inputs than outputs, then the problem is overdetermined. If $p > m$, so more outputs than inputs, then the problem is underdetermined and possibly not all outputs can be controlled. We will assume $p = m$ for the rest of the derivation of the input-output linearization. By denoting the elements of $\boldsymbol{h}$ as h_i, $i = 1, 2, \ldots, m$ and the columns vectors of $\boldsymbol{g}$ as $\boldsymbol{g}_j$, $j = 1, 2, \ldots, m$, we can define an extension to the relative degree ρ to MIMO systems. The vector relative degree $\boldsymbol{\rho} = [\rho_1\, \rho_2\, , \ldots, \rho_p]^T$ contains the relative degree of each output channel $i \in \{1, 2, \ldots, p\}$, where the relative degree ρ_i of an output channel i is defined as the smallest integer such that $\forall \boldsymbol{x} \in \mathbb{R}^n$, at least one input channel $j \in \{1, 2, \ldots, m\}$ satisfies:

$$L_{g_j} L_f^{\rho-1} h_i(x) \neq 0 \tag{3-17}$$

If furthermore the vector relative degree satisfies:

$$\rho = ||\boldsymbol{\rho}||_1 = \sum_{i=1}^{m} \rho_i \leq n \tag{3-18}$$

then the output dynamics of the system can be represented as:

$$\begin{bmatrix} y_1^{(\rho_1)} \\ y_2^{(\rho_2)} \\ \vdots \\ y_m^{(\rho_m)} \end{bmatrix} = \begin{bmatrix} \mathcal{L}_f^{\rho_1} h_1(x) \\ \mathcal{L}_f^{\rho_2} h_2(x) \\ \vdots \\ \mathcal{L}_f^{\rho_m} h_m(x) \end{bmatrix} + \begin{bmatrix} \mathcal{L}_{g_1}\mathcal{L}_f^{\rho_1-1} h_1(x) & \mathcal{L}_{g_2}\mathcal{L}_f^{\rho_1-1} h_1(x) & \cdots & \mathcal{L}_{g_m}\mathcal{L}_f^{\rho_1-1} h_1(x) \\ \mathcal{L}_{g_1}\mathcal{L}_f^{\rho_2-1} h_2(x) & \mathcal{L}_{g_2}\mathcal{L}_f^{\rho_2-1} h_2(x) & \cdots & \mathcal{L}_{g_m}\mathcal{L}_f^{\rho_2-1} h_2(x) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_{g_1}\mathcal{L}_f^{\rho_m-1} h_m(x) & \mathcal{L}_{g_2}\mathcal{L}_f^{\rho_m-1} h_m(x) & \cdots & \mathcal{L}_{g_m}\mathcal{L}_f^{\rho_m-1} h_m(x) \end{bmatrix} \boldsymbol{u} \tag{3-19}$$

which can be compactly written as

$$\boldsymbol{y}^{(\rho)} = \boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u} \tag{3-20}$$

If $\rho = n$ then the system is full-state feedback linearizable and there are no internal dynamics. Otherwise there are $n - \rho$ internal dynamics and stability of the total system can be shown using a nonlinear state transformation. If there are internal dynamics then there exist smooth functions

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \phi_1(\boldsymbol{x}) \\ \phi_2(\boldsymbol{x}) \\ \vdots \\ \phi_{n-\rho}(\boldsymbol{x}) \end{bmatrix} \tag{3-21}$$

such that

$$\frac{\partial \phi_k}{\partial x} g_j(x) = 0, \quad \forall k \in \{1, 2, \ldots, n-\rho\}, \quad \forall j \in \{1, 2, \ldots, m\}, \quad \forall x \in D_0. \tag{3-22}$$

Then using the diffeomorphism

$$\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x}) = [\boldsymbol{T_1}(\boldsymbol{x}), \boldsymbol{T_2}(\boldsymbol{x})]^T = [\boldsymbol{\eta}, \boldsymbol{\xi}]^T \tag{3-23}$$

with

$$\begin{aligned} \boldsymbol{\eta} &= \boldsymbol{\phi}(\boldsymbol{x}) && (3\text{-}24)\\ \boldsymbol{\xi} &= [\boldsymbol{\xi_1^T}, \boldsymbol{\xi_2^T}, \dots, \boldsymbol{\xi_m^T}]^T && (3\text{-}25)\\ \boldsymbol{\xi_i} &= [h_i(\boldsymbol{x}), \mathcal{L}_f h_i(\boldsymbol{x}), \dots, \mathcal{L}_f^{\rho_i - 1} h_i(\boldsymbol{x})]^T, \quad i = 1, 2, \dots, m && (3\text{-}26) \end{aligned}$$

the nonlinear system can be transformed into

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{f_0}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \left.\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})\right|_{\boldsymbol{x} = \boldsymbol{T^{-1}}(\boldsymbol{z})}\\ \dot{\boldsymbol{\xi}} &= \boldsymbol{A_c}\boldsymbol{\xi} + \boldsymbol{B_c}[\boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})u]\\ \boldsymbol{y} &= \boldsymbol{C_c}\boldsymbol{\xi} \end{aligned} \tag{3-27}$$

where $\boldsymbol{A_c} = \text{diag}\{A_i^0\}$, $\boldsymbol{B_c} = \text{diag}\{B_i^0\}$, $\boldsymbol{C_c} = \text{diag}\{C_i^0\}$, $i = 1, 2, \dots, m$, and (A_i^0, B_i^0, C_i^0) is a canonical form representation of a chain of ρ_i integrators. The NDI control is then chosen as

$$u = \boldsymbol{\mathcal{B}^{-1}}(\boldsymbol{x})(\boldsymbol{\nu} - \boldsymbol{\alpha}(\boldsymbol{x})) \tag{3-28}$$

so that the resulting closed-loop system is:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{f_0}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \left.\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})\right|_{\boldsymbol{x} = \boldsymbol{T^{-1}}(\boldsymbol{z})}\\ \dot{\boldsymbol{\xi}} &= \boldsymbol{A_c}\boldsymbol{\xi} + \boldsymbol{B_c}\boldsymbol{\nu}\\ \boldsymbol{y} &= \boldsymbol{C_c}\boldsymbol{\xi} \end{aligned} \tag{3-29}$$

This system has $n - \rho$ internal dynamics and m decoupled channels. The relation between each input channel v_i and each output channel y_i is a chain of ρ_i integrators. The NDI control input stabilizes the system when the internal dynamics are Input-to-State Stable (ISS) and the virtual control input $\boldsymbol{\nu}$ is chosen such that the linearized system is stable.

Applications of NDI or feedback linearization are commonly found in aerospace applications as the dynamics are nonlinear. It has been applied in aircraft [19] and in multirotors [16]. Note that by assumption of the dimensions, NDI can not directly be applied on an underactuated system where the number of inputs is smaller than the degrees-of-freedom (DoF). A specific controller needs to be designed to deal with this underactuation. For example in quadrotors the input has to be transformed to total thrust and moments in order to apply feedback linearization or a hierarchal control architecture can be applied. In (aerial) manipulation tasks NDI is often used without explicit mention, for example in impedance control [26, 28, 66, 35]. In [65] NDI is included to be able to apply H_∞-controller synthesis.

The NDI is a useful tool that transforms a nonlinear system into a decoupled linear system using feedback. Linear control can again be applied on the feedback linearized system such as PID-control [41] or linear robust control [65]. The main drawback is that the method is dependent on an accurate model description $\boldsymbol{\alpha}(\boldsymbol{x})$ and $\boldsymbol{\mathcal{B}}(\boldsymbol{x})$ to cancel the nonlinearities in the system. In reality it is impossible to obtain an perfectly accurate model due to many factors

such as model simplifications and external disturbances. The main drawbacks of NDI led to research to improve the robustness of this approach. For example combining linear robust control techniques such as $\mathcal{H}_\infty$ synthesis with NDI [65]. Another example is INDI, which uses measurements to reduce model dependency, which will be elaborately explained in the section 3-2.

3-2 Incremental Nonlinear Dynamic Inversion

Consider again the following class of nonlinear systems:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) \end{aligned} \tag{3-30}$$

in which $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^p$. The function $\boldsymbol{f}$ is a smooth mapping $\mathbb{R}^n \to \mathbb{R}^n$, $\boldsymbol{g}$ a smooth mapping $\mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^n$ and $\boldsymbol{h}$ a smooth mapping $\mathbb{R}^n \to \mathbb{R}^p$. Using input-output linearization as described in section 3-1 the model can be rewritten in the form:

$$\boldsymbol{y}^{(\rho)} = \boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u} \tag{3-31}$$

And if the vector relative degree of the system has the property $\rho \leq n$ then it can be rewritten in canonical form as in section 3-1:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{f_0}(\boldsymbol{\eta}, \boldsymbol{\xi}) \\ \dot{\boldsymbol{\xi}} &= \boldsymbol{A_c}\boldsymbol{\xi} + \boldsymbol{B_c}[\boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u}] \\ \boldsymbol{y} &= \boldsymbol{C_c}\boldsymbol{\xi} \end{aligned} \tag{3-32}$$

Suppose now the state, output derivatives and input can be measured at time t_0, which is Δt from the current time t. The output can be estimated then using a Taylor approximation around t_0. The measurements at t_0 of the state, output derivatives and input are $\boldsymbol{x_0}$, $\boldsymbol{y_0^{(\rho)}}$, $\boldsymbol{u_0}$ respectively. Then using a first order Taylor-expansion we estimate:

$$\boldsymbol{y}^{(\rho)} \approx \boldsymbol{y_0^{(\rho)}} + \frac{\partial(\boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u})}{\partial \boldsymbol{x}}\Big|_{\substack{\boldsymbol{x}=\boldsymbol{x_0} \\ \boldsymbol{u}=\boldsymbol{u_0}}} (\boldsymbol{x} - \boldsymbol{x_0}) + \mathcal{B}(\boldsymbol{x_0})(\boldsymbol{u} - \boldsymbol{u_0}) \tag{3-33}$$

If we assume $\boldsymbol{x} - \boldsymbol{x_0} \approx 0$ for sufficiently high sampling time Δt due to continuity of the state (this also known as the time-scale separation assumption [54]), then we can rewrite the relation. Defining $\boldsymbol{\Delta u} = \boldsymbol{u} - \boldsymbol{u_0}$, the INDI control law is then chosen as:

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{\Delta u} = \boldsymbol{u_0} + \boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{\nu} - \boldsymbol{y_0^{(\rho)}}) \tag{3-34}$$

The input increments with $\boldsymbol{\Delta u}$ relative to $\boldsymbol{u}_0$, which is why it is referred to as an incremental control method. Furthermore the matrix $\boldsymbol{\mathcal{B}}^{-1}$ is often referred to as the control effectiveness matrix. Recall the NDI control law was defined as:

$$\boldsymbol{u} = \boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{\nu} - \boldsymbol{\alpha}(\boldsymbol{x}))$$

When comparing the two expressions we notice that the INDI is not dependent on $\boldsymbol{\alpha}$ anymore, but instead on the output measurement $\boldsymbol{y_0^\rho}$ and measured input $\boldsymbol{u_0}$. INDI is less dependent

on the complete model, but requires additional output and input measurements. This is why INDI is also referred to as a sensor-based approach.

INDI is inherently a discrete control method so a sampling time Δt should be chosen for the control method. The stability of the control method also depends on this sampling time. Intuitively the stability of INDI can be linked to the so called time-scale separation [54, 52], though this assumption is not mathematically rigorous. In [62] a rigorous proof using Lyapunov methods has been given. Furthermore in [27] stability conditions on model errors and time delays are given.

INDI has appeared mostly in aerospace application. In [54] an adaptive INDI method has been applied on control of a multi-rotor and in [47] on an aerial manipulator. Robustness of INDI is verified in many simulations and experiments [37, 36]. In [61] a comparison is made between NDI and INDI for reference tracking for a rigid aircraft with different modeling errors and it is shown that INDI outperforms NDI greatly in robustness. The main drawback of INDI is the use of acceleration measurements which can be done using direct measurements [11] or indirect derivation methods [54, 37]. For indirect derivation methods the state derivative has to be calculated, which amplifies high frequency noise. For this reason filters have to be applied, however this influences the time delay between control and measurements and therefore also stability and performance of the closed-loop system. In [36] a nondelay differentiator design is proposed for the state derivative calculation to combat this time delay.

3-3 Disturbance Estimation

Consider a nonlinear system written in input-output form

$$\boldsymbol{y}^{(\boldsymbol{\rho})} = \boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u}, \tag{3-35}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the output vector, $\boldsymbol{\rho}$ denotes the vector of relative degrees. Systems expressed in dynamic canonical form, as introduced in Chapter 2, are examples of this structure, where typically $\boldsymbol{\rho} = \mathbf{1}$.

Assume now that there is an additive disturbance $\boldsymbol{d} \in \mathbb{R}^m$, which affects each output channel:

$$\boldsymbol{y}^{(\boldsymbol{\rho})} = \boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}. \tag{3-36}$$

Suppose that the output derivatives $\boldsymbol{y}^{(\boldsymbol{\rho})}$, the state $\boldsymbol{x}$, and the control input $\boldsymbol{u}$ are known at a given time instant t_0. Then, an estimate of the disturbance at time t_0 can be obtained as

$$\hat{\boldsymbol{d}}_0 = \boldsymbol{y}_0^{(\boldsymbol{\rho})} - \hat{\boldsymbol{\alpha}}(\boldsymbol{x}_0) - \hat{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\boldsymbol{u}_0, \tag{3-37}$$

where $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\mathcal{B}}}$ denote the model estimates evaluated at t_0.

3-3-1 Disturbance Observer

Rather than directly computing the disturbance from instantaneous measurements, it is common to construct a disturbance observer as in [14]. A common assumption is that the disturbance varies slowly compared to the system dynamics:

$$\dot{\boldsymbol{d}} \approx \mathbf{0}. \tag{3-38}$$

Define the disturbance estimation error as

$$\boldsymbol{e}_d = \boldsymbol{d} - \hat{\boldsymbol{d}}. \tag{3-39}$$

We will use first-order error dynamics,

$$\dot{\boldsymbol{e}}_d + L\boldsymbol{e}_d = \boldsymbol{0}, \tag{3-40}$$

with $L \in \mathbb{R}^{m\times m}$ Hurwitz. This yields the observer dynamics

$$\dot{\hat{\boldsymbol{d}}} = -L\hat{\boldsymbol{d}} + L\left(\boldsymbol{y}^{(\rho)} - \boldsymbol{\alpha}(\boldsymbol{x}) - \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u}\right). \tag{3-41}$$

Under the assumption of bounded model errors and sufficiently slow disturbance variation, the estimation error $\boldsymbol{e}_d$ converges to zero.

3-3-2 Disturbance Observer Based Control

Disturbance estimates can be incorporated into the control law to improve robustness against external disturbances and model uncertainties. Note that since the disturbance estimate is model-based, unmodeled dynamics and parametric uncertainties are lumped into the estimated disturbance.

Consider applying NDI to system (3-36), which includes additive disturbances. If a disturbance estimate $\hat{\boldsymbol{d}}$ is available, the control law can be chosen as

$$\boldsymbol{u} = \hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{x})\left(\boldsymbol{\nu} - \hat{\boldsymbol{\alpha}}(\boldsymbol{x}) - \hat{\boldsymbol{d}}\right), \tag{3-42}$$

which results in linear output dynamics with virtual control input $\boldsymbol{\nu}$ if the model is completely correct.

Now consider using the direct disturbance estimate (3-37) evaluated at time t_0. Substituting $\hat{\boldsymbol{d}}_0$ into the NDI control law gives

$$\boldsymbol{u} = \hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{x}_0)\left(\boldsymbol{\nu} - \boldsymbol{y}_0^{(\rho)} + \hat{\boldsymbol{\mathcal{B}}}(\boldsymbol{x}_0)\boldsymbol{u}_0\right), \tag{3-43}$$

which can be rewritten as the incremental control law

$$\Delta\boldsymbol{u} = \hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{x}_0)\left(\boldsymbol{\nu} - \boldsymbol{y}_0^{(\rho)}\right). \tag{3-44}$$

This expression is exactly the INDI control law.

Now consider the disturbance observer dynamics in (3-41). This can be written as

$$\dot{\hat{\boldsymbol{d}}} = L\left(\boldsymbol{y}^{(\rho)} - \boldsymbol{\alpha}(\boldsymbol{x}) - \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u} - \hat{\boldsymbol{d}}\right), \tag{3-45}$$

which reveals that the observer acts as a first-order filter of the instantaneous disturbance estimate. We can take the direct disturbance estimate as in (3-37).

For sufficiently large observer gain L, the observer dynamics become fast compared to the plant dynamics. Introducing $L = \varepsilon^{-1}I$ with $0 < \varepsilon \ll 1$, the observer equation can be written as

$$\varepsilon\dot{\hat{\boldsymbol{d}}} = \hat{\boldsymbol{d}}_0 - \hat{\boldsymbol{d}}. \tag{3-46}$$

In the limit $\varepsilon \to 0$, the fast dynamics result in the disturbance estimate

$$\hat{\boldsymbol{d}} = \hat{\boldsymbol{d}}_0, \tag{3-47}$$

which shows that a disturbance observer approaches the direct disturbance estimate obtained from instantaneous measurements in the high-gain limit.

This result further clarifies the relationship between NDI with disturbance observer and INDI. While NDI explicitly relies on model-based cancellation and dynamic disturbance estimation, INDI implicitly assumes a high-bandwidth disturbance estimation by directly using incremental measurements. Consequently, INDI can be interpreted as the high-gain limit of NDI with a disturbance observer.

3-4 Nonlinear Model Predictive Control

Rather than solving a nonlinear control problem analytically, an optimal solution can be obtained numerically through the use of optimization algorithms. This is achieved by including the nonlinear system dynamics into a nonlinear Optimal Control Problem (OCP) as equality constraints. NMPC is an advanced control strategy that solves this OCP iteratively in a receding horizon fashion.

Consider a continuous-time nonlinear system described by:

$$\dot{\boldsymbol{x}}(t) = f_c(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{3-48}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ is the state vector and $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ is the control input vector. To formulate the OCP, the system is discretized with a sampling time T_s, leading to the following discrete-time model:

$$\boldsymbol{x}_{k+1} = f_d(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{3-49}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ denote the state and input at discrete time step k, respectively.

At any current time step t, given the measured or estimated current state $\boldsymbol{x}(t)$, a finite-horizon OCP over a prediction horizon N is formulated as follows:

$$\begin{aligned}
\min_{\boldsymbol{X},\boldsymbol{U}} \quad & \sum_{i=0}^{N-1} \ell(\boldsymbol{x}_i, \boldsymbol{u}_i) + V_f(\boldsymbol{x}_N) \\
\text{s.t.} \quad & \boldsymbol{x}_0 = \boldsymbol{x}(t) \\
& \boldsymbol{x}_{i+1} = f_d(\boldsymbol{x}_i, \boldsymbol{u}_i), && i = 0, \dots, N-1 \\
& g(\boldsymbol{x}_i, \boldsymbol{u}_i) \leq 0, && i = 0, \dots, N-1 \\
& h(\boldsymbol{x}_i, \boldsymbol{u}_i) = 0, && i = 0, \dots, N-1 \\
& \boldsymbol{x}_N \in \mathcal{X}_f
\end{aligned} \tag{3-50}$$

Here, $\boldsymbol{U} = \{\boldsymbol{u}_0, \dots, \boldsymbol{u}_{N-1}\}$ is the sequence of control inputs, and $\boldsymbol{X} = \{\boldsymbol{x}_0, \dots, \boldsymbol{x}_N\}$ is the corresponding sequence of predicted states. The objective function consists of a stage cost $\ell(\cdot)$ and a terminal cost $V_f(\cdot)$. The optimization is subject to the initial condition constraint $\boldsymbol{x}_0 = \boldsymbol{x}(t)$, the nonlinear system dynamics $f_d(\cdot)$, generic inequality constraints $g(\cdot)$, equality

constraints $h(\cdot)$, and optionally a terminal set constraint $\mathcal{X}_f$, which is usually included for stability properties.

Solving (3-50) yields an optimal input sequence $\boldsymbol{U}^* = \{\boldsymbol{u}_0^*, \boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_{N-1}^*\}$. The receding horizon principle then is to only apply the first element of this optimal sequence to the system:

$$\boldsymbol{u}(t) = \boldsymbol{u}_0^* \tag{3-51}$$

At the next control instant, the horizon is shifted with the NMPC control time step, the state is measured again, and the optimization problem is solved again.

The NMPC can be used as described above as direct feedback controller. NMPC can also be used in a cascaded architecture as a high-level trajectory planner. In this paradigm, rather than exclusively using the first control input, the predicted optimal state trajectory $\boldsymbol{X}^*$ can be passed to a high-frequency inner-loop controller as a reference. Furthermore, the optimal control sequence $\boldsymbol{U}^*$ can be injected as a feedforward term to enhance the trajectory tracking performance of the inner loop.

NMPC has seen widespread use in aerial robotics. For example, Lunni et al. [39] implemented NMPC on an onboard computer to enable a physical aerial manipulator to track a 3D reference with its end-effector. Similarly, Tzoumanikas et al. [59] utilized NMPC to control an aerial manipulator for complex aerial writing tasks. Sun et al. [55] successfully applied NMPC for agile drone flight, analyzing its performance both as a standalone controller and within a cascaded control architecture. Furthermore, Marti-Saumell et al. [42] demonstrated the effectiveness of NMPC in executing aggressive maneuvers across various simulated environments.

A primary advantage of NMPC is its systemic ability to explicitly handle state and input constraints within complex systems. Furthermore, the predictive nature of the receding horizon allows the system to incorporate future dynamics and constraints into current control decisions. This is particularly beneficial for systems that need to navigate highly dynamic or constrained environments. Moreover, the increasing computational capabilities of lightweight embedded computers and more efficient optimization algorithms have made real-time onboard NMPC feasible for autonomous robots. However, the approach has notable drawbacks: solving non-convex, nonlinear optimization problems in real-time is computationally demanding, and the performance is highly dependent on the accuracy of the dynamic model and the rigorous tuning of the cost function.

Finally, guaranteeing the recursive feasibility and asymptotic stability of the OCP for non nominal cases remains a challenging theoretical problem. For linear systems, robust linear Model Predictive Control (MPC) paradigms, such as tube-based MPC, have been well-established to guarantee stability and recursive feasibility under uncertainties [43]. While robust NMPC frameworks exist, proving recursive feasibility and designing robust invariant sets for highly nonlinear dynamics requires significantly more effort and is part of ongoing research.

3-5 Summary

In this Chapter, the nonlinear control methods NDI, INDI and NMPC are formally introduced. With these nonlinear control tools and the modeling approaches in Chapter 2, we can introduce

the controller designs proposed in this work for the Differential Shoulder Aerial Manipulator (DSAM) for performing Aerial Physical Interaction (APhI) tasks. This will be done in the following Chapter.

Chapter 4

Controller Design

In this chapter we will introduce all the parts of the novel controller designs proposed in this work. The goal of the proposed controllers in this work is to achieve simultaneous force and end-effector pose tracking in Aerial Physical Interaction (APhI) tasks. Incremental Nonlinear Dynamic Inversion (INDI) will be considered to enable robust tracking performance and furthermore we consider Nonlinear Model Predictive Control (NMPC) as outer loop similar to [55]. The following control schemes will be proposed and compared:

1. A novel baseline NMPC formulation adapted from [55] and where the dynamics are modified based on the Differential Shoulder Aerial Manipulator (DSAM) dynamic model derived in Chapter 2 and states related to the manipulator are added to the cost. The formulation will be contact-aware. External force feedback (FF) is used in the cost of the NMPC and the desired reference force is included in the NMPC dynamics (NMPC-FF).

2. The NMPC-FF controller augmented with a INDI inner loop attitude controller, similar to the work of [55] (NMPC-FF-INDI)

3. The NMPC-FF controller augmented with a cascaded INDI inner loop controller (NMPC-FF-CINDI), where the cascaded controller is based on [53].

The proposed controller formulations are valid for general (underactuated) multibody aerial manipulator systems and can also be applied to other aerial manipulator configurations by changing the multibody dynamics.

We will assume all generalized coordinates and velocities to be available from a Extended Kalman Filter (EKF) algorithm combining VICON motion capture measurements and Inertial Measurement Unit (IMU) measurements. Also the acceleration of the base is available from the IMU of the DSAM. In the next sections we will explore each of the proposed controllers in more detail.

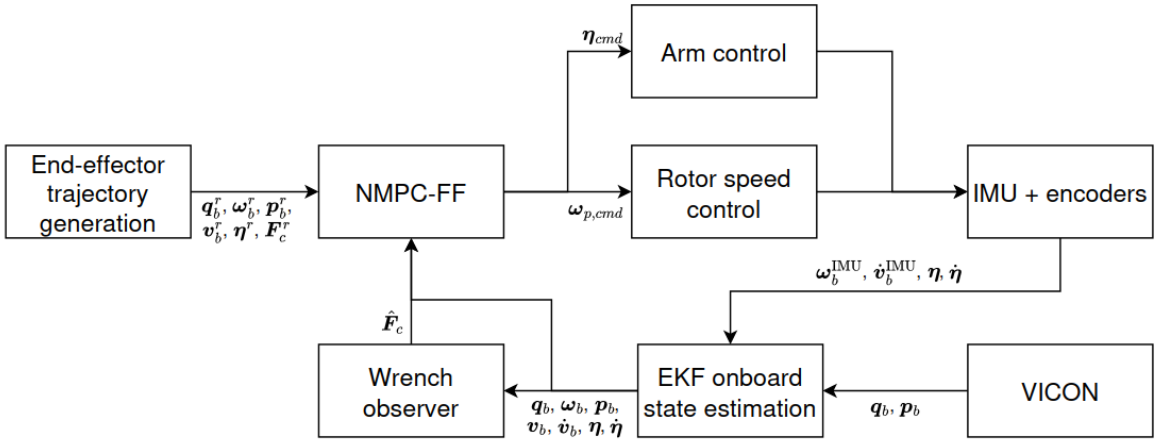


Figure 4-1: A schematic of the NMPC-FF control loop.

4-1 Baseline NMPC formulation

A schematic of the high level NMPC-FF control loop is given in Figure 4-1. In this section we will define the Optimal Control Problem (OCP) of the baseline NMPC controller. Our baseline controller will have some similarities to the control design in [59], however there are some significant hardware differences such as a different manipulator and aerial platform. For this reason it is not possible to exactly copy the controller and the baseline might not perform as well as for the original application, however it provides us with an initial design. One significant hardware difference between our aerial manipulator and the one in [59] is the Delta arm and in our case the two-degrees-of-freedom (DoF) arm. Also in [59] they attached a spring to the pen to add compliance. A linear spring contact model is used in the NMPC. This model is an accurate description of their aerial manipulator because of the compliance in the hardware. We will not have this compliance in our manipulator. In this section we will reconsider the contact modeling to try to improve performance in contact-based tasks.

For the proposed NMPC controller a model containing the Equations of Motion (EoM) of the aerial manipulator system in free flight is used and extended with a model of the contact dynamics. We will assume knowledge on wall position and the normal vector of the surface. This information can be estimated using sensors such as cameras, or be obtained from prior knowledge of the environment.

4-1-1 Contact modeling

At every time instant in which the aerial manipulator interacts with the environment, we assume this interaction will be on a single point C located at the contact position $\boldsymbol{p}_c$ of a surface with normal vector $\boldsymbol{e}_n$. The values $\hat{\boldsymbol{p}}_c$ and $\hat{\boldsymbol{e}}_n$ are the estimated contact position and normal vector at that timestep, which depend on the reference trajectory and can change each time step. For example with vision information it is possible to obtain a parametrization of a surface and derive these values as done in [6].

The estimation of parameters of the contact surface is not the focus of this work. For simplicity we assume surfaces to have simple geometries. Specifically we assume in the contact-based

tasks that the surface location is prior knowledge and we assume that the surface has a constant normal vector over its entire surface. Considering these simplifications, only the projected distance on the normal direction of the wall is relevant. As the surface is a flat surface, $\hat{\boldsymbol{p}}_c$ can be chosen to be any arbitrary point on this surface.

In [59] a linear spring model is used to predict the contact force $\boldsymbol{F}_c$. This force acts in the normal direction of the surface at point C and is dependent on the penetration depth of the end-effector d_{pen}. The penetration depth is calculated as follows:

$$d_{pen} = (\hat{\boldsymbol{p}}_c - \boldsymbol{p}_e)^T \hat{\boldsymbol{e}}_n \tag{4-1}$$

The contact force is then calculated as:

$$\boldsymbol{F}_c = k_{p,\text{contact}} d_{pen} \hat{\boldsymbol{e}}_n \tag{4-2}$$

where $k_{p,\text{contact}}$ is the chosen stiffness of the contact model. This contact model is used to predict contact and anticipate it in a given trajectory. It is used in the both the dynamics and the cost of their proposed NMPC formulation. For our aerial manipulator design this model is not an accurate description of reality as mentioned before and it shows oscillatory behavior especially at the moment of contact. In the results section a brief comparison will be made with this way of contact modeling and our proposed controller in simulation for the DSAM.

In our proposed controller the contact force enters the NMPC formulation in two different ways. In the dynamics the desired force is directly used. In the cost a combination between an estimated contact force and the linear spring model is used. The external contact force can for example be directly measured using a force/torque sensor at the end-effector, or be directly estimated from the dynamics. We will choose the latter due to hardware constraints.

We want to estimate the external wrench $\boldsymbol{w}_{\text{ext}}$ acting on the aerial manipulator. This estimate is obtained by comparing the filtered acceleration measurements with the model-based accelerations computed from the filtered control inputs. The external wrench consists of the external force $\boldsymbol{F}_{\text{ext}}$ and the external moment $\boldsymbol{M}_{\text{ext}}$:

$$\boldsymbol{w}_{\text{ext}} = \begin{bmatrix} \boldsymbol{F}_{\text{ext}} \\ \boldsymbol{M}_{\text{ext}} \end{bmatrix} \tag{4-3}$$

By assuming quasi-static motion of the arm, the contributions of the arm dynamics to the total force can be neglected. The external force estimate in the inertial frame is then given by:

$$\hat{\boldsymbol{F}}_{\text{ext}} = \boldsymbol{R}_{IB} \left(m_{\text{total}} \boldsymbol{a}_f - \boldsymbol{F}_{b,f} \right) \tag{4-4}$$

where m_{total} is the total mass of the aerial manipulator, $\boldsymbol{a}_f$ is the filtered body acceleration measurement, and $\boldsymbol{F}_{b,f}$ is the estimated force acting on the base computed from the filtered motor speeds $\boldsymbol{u}_f$. The rotation matrix $\boldsymbol{R}_{IB}$ transforms the force from the body frame to the inertial frame.

Similarly, under the quasi-static arm assumption, the contribution of the arm angular dynamics to the total body moment can be neglected. The external moment estimate in the body frame is then:

$$\hat{\boldsymbol{M}}_{\text{ext}} = \boldsymbol{I}_{\text{total }B} \dot{\boldsymbol{\omega}}_{b,f} + {}_B\boldsymbol{\omega}_{b,f} \times (\boldsymbol{I}_{\text{total }B} \boldsymbol{\omega}_{b,f}) - \boldsymbol{\tau}_{b,f} \tag{4-5}$$

where $\boldsymbol{I}_{\text{total}}$ is the total inertia matrix of the aerial manipulator in the current configuration, ${}_B\dot{\boldsymbol{\omega}}_{b,f}$ is the filtered angular acceleration measurement, $\boldsymbol{\tau}_{b,f}$ is the estimated torque acting on the body computed from the filtered motor speeds, $\boldsymbol{\omega}_{b,f}$ is the filtered angular velocity and ${}_B\boldsymbol{\omega}_{b,f} \times (\boldsymbol{I}_{\text{total}}\, {}_B\boldsymbol{\omega}_{b,f})$ is the gyroscopic term. In this work this estimated external torque was not used as the sliding tasks applies no external moments on the end-effector.

For the NMPC formulation we will assume the estimated external force to be the contact force when contact is expected. We take $\hat{\boldsymbol{F}}_c$ as the estimated external force $\boldsymbol{F}_{ext}$. The contact force used in the cost will be based on the sum of the estimated external force and the linear spring model:

$$\boldsymbol{F}_{c,\text{cost},k} = \max\left(0, i_{c,k}\hat{\boldsymbol{F}}_c^T\hat{\boldsymbol{e}}_n + k_{p,\text{contact}}d_{\text{pen}}\right)\hat{\boldsymbol{e}}_n \tag{4-6}$$

where $\boldsymbol{p}_{e,k}$ is the predicted end-effector position at timestep k and $i_{c,k}$ is a contact indicator variable at time-step k. Define our reference penetration depth at time step k as:

$$d^r_{\text{pen},k} = (\hat{\boldsymbol{p}}_c - \boldsymbol{p}^r_{e,k})^T\hat{\boldsymbol{e}}_n \tag{4-7}$$

Then we define the contact variable as follows:

$$i_{c,k} = \begin{cases} 0 & \text{if } d^r_{\text{pen},k} \leq -\dfrac{w_t}{2} \\ \dfrac{1}{2}\left(1 - \cos\left(\pi\dfrac{d^r_{\text{pen},k} + \frac{w_t}{2}}{w_t}\right)\right) & \text{if } \left|d^r_{\text{pen},k}\right| < \dfrac{w_t}{2} \\ 1 & \text{if } d^r_{\text{pen},k} \geq \dfrac{w_t}{2} \end{cases} \tag{4-8}$$

where w_t is a contact transition width which is chosen to be 1 cm. This modeling approach is chosen to avoid discontinuities at the exact contact position.

The reason we multiply the external force with a contact indicator variable is because the external force can be due to different reasons, for example disturbances and modeling errors. By introducing the contact indicator we only assume the measured external force is a contact force when contact is expected in the reference trajectory. We also take the dot product between the external force and the normal direction of the surface, because we want to track the normal contact force for our application. There might be friction forces due to contact and in this way they will be ignored in the cost. The contact indicator is dependent on the reference penetration depth. This design choice was taken to avoid switching behavior in contact-based tasks.

Also note the use of the maximum operator. We assume that the contact force applied can only be positive. The intuition behind this is that a contact force will never 'pull' the aerial manipulator towards the surface, only push it away due to contact. The spring model is added so the planner is able to relate force with base motion. By planning a trajectory into a contact surface, the system will apply more force and the estimated contact force should increase. By moving away from the wall, the force decreases and the estimated contact force should also decrease. The intuition behind this combination between estimated contact force and spring model was to create a force tracking ability of the controller. The addition of this cost term will also allow us to combine the NMPC controller with the cascaded INDI controller for physical interaction in any direction, which will be further explored in the experimental results.

The contact force in the dynamics will be defined differently:

$$\boldsymbol{F}_{c,\mathrm{dyn},k} = i_{c,k}\boldsymbol{F}^{r}_{c,k} \tag{4-9}$$

where $\boldsymbol{F}^{r}_{c,k}$ is the reference force which can be directly used from the reference trajectory.

The reason for introducing the contact force differently in the dynamics and cost will also be explored in more detail in the results section, where we will compare the proposed controller to modified controllers using the same contact force modeling for the cost and dynamics. However one important reason of not using the external force estimate in the dynamics is that this estimate is based on accelerometer data, which is found to be very noisy in the physical DSAM system.

4-1-2 NMPC-FF formulation

Now that we have a model for the contact force, we want to include it in the NMPC formulation to make it contact-aware. The contact force is applied on the end-effector when it is interacting with the environment. This external force on the end-effector can be related to the dynamics in generalized coordinates using the end-effector Jacobian J_E. This Jacobian can be obtained as described in Section 2-1. The contact force is introduced in the dynamics as follows:

$$M(\boldsymbol{q})\dot{\boldsymbol{v}} + C(\boldsymbol{v},\boldsymbol{q}) + G(\boldsymbol{q}) = \boldsymbol{\tau} + J_E^T\boldsymbol{F}_{c,\mathrm{dyn}} \tag{4-10}$$

The state and input variables of the system are the generalized variables and velocities of an aerial manipulator system described in Chapter 2. The state vector at each time step is defined as

$$\boldsymbol{x} = [\boldsymbol{p}_b^T\ \boldsymbol{v}_b^T\ \boldsymbol{q}_b^T\ {}_B\boldsymbol{\omega}_b^T\ \boldsymbol{\eta}^T\ \dot{\boldsymbol{\eta}}^T]^T \tag{4-11}$$

where $\boldsymbol{p}_b^T$ is position of the base in the world frame, $\boldsymbol{v}$ is the velocity of the base in the world frame, $\boldsymbol{q}_b$ the quaternion parametrization of the base orientation relative to the world frame, ${}_B\boldsymbol{\omega}_b$ the body angular velocity in the body frame and $\boldsymbol{\eta}$ the manipulator joint angles. The input vector is given as:

$$\boldsymbol{u} = [\boldsymbol{\omega}_p^T\ \boldsymbol{\eta}_r^T]^T \tag{4-12}$$

where $\boldsymbol{\omega}_p$ is the vector of the rotor speeds and $\boldsymbol{\eta}_r$ are the desired joint angles. These desired joint angles will be used as reference in the low level controller of the manipulator. Note that in the dynamics we derived in Chapter 2, we did not have joint reference positions $\boldsymbol{\eta}_r$ as input, but instead the arm torques. The reason for this modification is due to hardware constraints where the arm is controlled using position control. The dynamics of the arm in the whole-body dynamics and thus the NMPC is modified to an approximation:

$$\ddot{\boldsymbol{\eta}} = k_{p,\mathrm{arm}}(\boldsymbol{\eta}_r - \boldsymbol{\eta}) - k_{d,\mathrm{arm}}\dot{\boldsymbol{\eta}} \tag{4-13}$$

The actual position control of the arm is not a PD controller in the DSAM, so this part will introduce some model mismatch into the whole-body dynamics.

Furthermore the propeller speeds $\boldsymbol{\omega}_p$ also need a mapping to body torques and forces for the multibody dynamics. For the NMPC formulation the inertial forces of the rotors are neglected and the direct mapping:

$$\begin{pmatrix} {}_B\boldsymbol{\tau} \\ {}_B\boldsymbol{F} \end{pmatrix} = \boldsymbol{G}_1\boldsymbol{\omega}_p \circ \boldsymbol{\omega}_p \tag{4-14}$$

is used.

The NMPC will find the optimal control sequence $U^* = [\boldsymbol{u}_0^*, \boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_N^*]$ and state trajectory $X^* = [\boldsymbol{x}_0^*, \boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_N^*]$ by minimizing the sum of stage costs $L_k(\boldsymbol{x}_k, \boldsymbol{u}_k)$ for $k \in [0, 1, \ldots, N]$ and the terminal cost $\Phi_N(\boldsymbol{x}_N)$), where the solution satisfies the imposed dynamics and constraints. The mathematical formulation of the OCP proposed in this work has the following form:

$$U^* = \arg\min_U \quad \sum_{k=0}^{N-1} L(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{x}_{k,r}) + \Phi(\boldsymbol{x}_N, \boldsymbol{x}_{N,r}) \tag{4-15}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = f_d(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{F}_{c,\text{dyn},k}), \tag{4-16}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}(t), \tag{4-17}$$

$$\boldsymbol{u} \in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}] \tag{4-18}$$

in which N is the discrete horizon length and f_d is the discretized dynamics. The input bounds $\boldsymbol{u}_{\min} = [0, 0, 0, 0, -\frac{7}{12}\pi, -\pi]^T$ and $\boldsymbol{u}_{\max} = [2000, 2000, 2000, 2000, \frac{7}{12}\pi, \pi]^T$ were chosen for the NMPC corresponding to the minimum and maximum rotor speeds and joint angles.

The stage costs L consist of the error between the reference and predicted state and the magnitude of the predicted control input. The terminal cost is only based on the error in predicted and reference state. The following errors are defined:

$$\boldsymbol{e}_{p,B} = \boldsymbol{p}_B - \boldsymbol{p}_{B,r}, \tag{4-19}$$

$$\boldsymbol{e}_v = \boldsymbol{v} - \boldsymbol{v}_r, \tag{4-20}$$

$$\boldsymbol{e}_{q,B} = [\boldsymbol{q}_{IB}^{-1} \otimes \boldsymbol{q}_{IB,r}]_{1:3}, \tag{4-21}$$

$$\boldsymbol{e}_{\boldsymbol{\omega}} = {}_B\boldsymbol{\omega}_b - R_{BB_r}\, {}_{B_r}\boldsymbol{\omega}_{b,r} \tag{4-22}$$

$$\boldsymbol{e}_\eta = \boldsymbol{\eta} - \boldsymbol{\eta}_r \tag{4-23}$$

$$\boldsymbol{e}_{\dot{\eta}} = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_r \tag{4-24}$$

$$\boldsymbol{e}_{p,E} = \boldsymbol{p}_E - \boldsymbol{p}_{E,r} \tag{4-25}$$

$$\boldsymbol{e}_{n,E} = {}_I\boldsymbol{e}_x^E - {}_I\boldsymbol{e}_x^{E_r} \tag{4-26}$$

$$\boldsymbol{e}_u = \boldsymbol{u} - \boldsymbol{u}_r, \tag{4-27}$$

$$e_f = ||\boldsymbol{F}_{c,\text{cost}}||_2 - ||\boldsymbol{F}_c^r||_2 \tag{4-28}$$

where ${}_I\boldsymbol{e}_x^E$ is the unit x-vector of the end-effector frame Σ_E expressed in the world frame Σ_I and ${}_I\boldsymbol{e}_x^{E_r}$ is the unit x-vector of the reference end-effector frame Σ_{E_r} expressed in the world frame Σ_I. Each of the errors has a corresponding weighting matrix $Q_{p,B}$, Q_v, $Q_{q,B}$, Q_ω, Q_η, $Q_{\dot{\eta}}$, $Q_{p,E}$, $Q_{n,E}$, Q_u and Q_f which are parameters that have to be chosen. The stage cost is then defined as

$$L(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{x}_{k,r}) = \sum_j \boldsymbol{e}_j^T Q_j \boldsymbol{e}_j \tag{4-29}$$

with $\boldsymbol{e}_j \in \{\boldsymbol{e}_{p,B}, \boldsymbol{e}_v, \boldsymbol{e}_{q,B}, \boldsymbol{e}_{\boldsymbol{\omega}}, \boldsymbol{e}_\eta, \boldsymbol{e}_{\dot{\eta}}, \boldsymbol{e}_{p,E}, \boldsymbol{e}_{n,E}, \boldsymbol{e}_u, e_f\}$ and the terminal cost

$$\Phi(\boldsymbol{x}_N, \boldsymbol{x}_{N,r}) = \sum_j \boldsymbol{e}_j^T Q_j \boldsymbol{e}_j \tag{4-30}$$

with $\boldsymbol{e}_j \in \{\boldsymbol{e}_{p,B}, \boldsymbol{e}_v, \boldsymbol{e}_{q,B}, \boldsymbol{e}_{\boldsymbol{\omega}}, \boldsymbol{e}_\eta, \boldsymbol{e}_{\dot{\eta}}, \boldsymbol{e}_{p,E}, \boldsymbol{e}_{n,E}, e_f\}$.

This optimization problem is solved each time step in a receding horizon fashion. The optimal input sequence obtained each timestep will be the desired rotor speed commands and the desired joint angles of the arm.

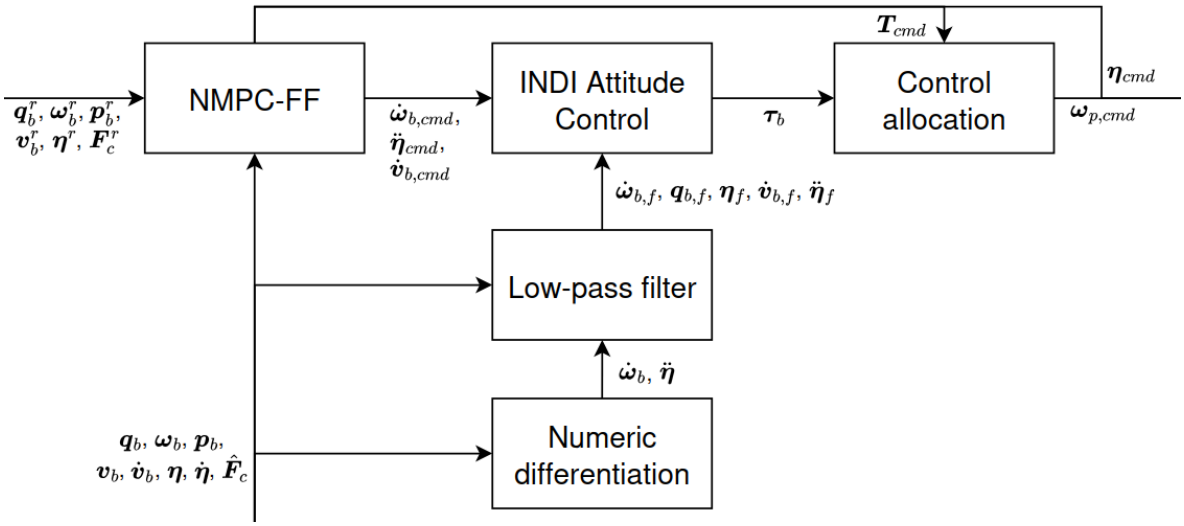


Figure 4-2: A schematic of the NMPC-FF-INDI control block

4-2 NMPC-FF-INDI formulation

In this section a controller will be proposed which consists of the previously introduced NMPC controller extended with an inner loop INDI motion controller similar to [55]. This work uses an INDI attitude controller designed for a quadrotor system and it needs to be adapted to our aerial manipulator system. We will refer to this controller as NMPC-FF-INDI.

4-2-1 High-level Overview

The control scheme of this controller will be the same as in Figure 4-1, but the NMPC-FF control block is replaced with the extended schematic given in Figure 4-2. One notable difference between the NMPC-FF controller is that besides the optimal control sequence of the NMPC, also the optimal state sequence is used. In this way the NMPC is also used as a trajectory planner consistent with the multibody dynamics of the aerial manipulator. This optimal state sequence generates the virtual control inputs for the inner loop INDI attitude controller. The INDI controller should then follow this generated reference and reject unmodeled disturbances. The total thrust is obtained from the NMPC optimal input sequence. The combination of the thrust and torques is then used in INDI control allocation to obtain the desired rotor speed commands.

4-2-2 Relating Multibody Dynamics and INDI

First we relate our DSAM dynamic equations to the general nonlinear system used introduced in Section 3-2. More generally the following description holds for any dynamic equation that is written in canonical form. Recall our class of nonlinear systems:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) \end{aligned} \tag{4-31}$$

The canonical dynamic equations are given as:

$$M(\boldsymbol{q})\dot{\boldsymbol{v}} + C(\boldsymbol{q},\boldsymbol{v})\boldsymbol{v} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

and the relation between generalized velocities $\boldsymbol{v}$ and generalized coordinates $\boldsymbol{q}$ is given as:

$$\dot{\boldsymbol{q}} = E(\boldsymbol{q})\boldsymbol{v} \tag{4-32}$$

By choosing states $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{q}\\ \boldsymbol{v}\end{bmatrix}$, output $\boldsymbol{y} = \boldsymbol{v}$ and input $\boldsymbol{u} = \boldsymbol{\tau}$ we see that

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} E(\boldsymbol{q})\boldsymbol{v} \\ M(\boldsymbol{q})^{-1}(-C(\boldsymbol{x})\boldsymbol{v} - G(\boldsymbol{q}))\end{bmatrix}, \boldsymbol{G}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{0}_{n_q \times n_u} \\ M(\boldsymbol{q})^{-1}\end{bmatrix} \tag{4-33}$$

and furthermore that the canonical dynamic equations are already written in output dynamics where the vector relative degree is one for each output channel. Recall the output dynamics

$$\boldsymbol{y}^{(\rho)} = \boldsymbol{\alpha}(\boldsymbol{x}) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u} \tag{4-34}$$

where now $\boldsymbol{\alpha}(\boldsymbol{x}) = M(\boldsymbol{q})^{-1}(-C(\boldsymbol{x})\boldsymbol{v} - G(\boldsymbol{q}))$ and $\boldsymbol{\mathcal{B}}(\boldsymbol{x}) = M(\boldsymbol{q})^{-1}$. Note that our input $\boldsymbol{u}$ in the canonical equations are the generalized forces. In the derived system this includes all translational directions, while our actual quadrotor base can only apply thrust in the body z direction. In Sections 3-1 and 3-2 the assumption for input-output linearization was that $m = p$ which does not hold for our underactuated aerial platform.

As applying feedback linearization directly will not be possible, an assumption will be made on the dynamics of the DSAM for applying a INDI inner loop. A common assumption is that the attitude and arm control is much faster than the positional control for an aerial manipulator. By separating the translational dynamics and angular and arm dynamics it is possible to design a INDI attitude controller similar to [55] or even to create a cascaded control structure, similar to quadrotor control in [53]. However one important difference between quadrotor control and aerial manipulator control is that the angular dynamics is completely decoupled from the translational dynamics of the quadrotor and the presence of an actuated arm. To take into account these differences, we will need to redesign the INDI controllers to make them work for the aerial manipulator system. For this purpose we will partition the whole-body dynamics as follows:

$$\begin{bmatrix}\boldsymbol{T}\\ \boldsymbol{\tau}_b\\ \boldsymbol{\tau}_\eta\end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13}\\ M_{12}^T & M_{22} & M_{23}\\ M_{13}^T & M_{23}^T & M_{33}\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{v}}_b\\ {}_B\dot{\boldsymbol{\omega}}_b\\ \ddot{\boldsymbol{\eta}}\end{bmatrix} + \begin{bmatrix}\boldsymbol{C}_1\\ \boldsymbol{C}_2\\ \boldsymbol{C}_3\end{bmatrix} \tag{4-35}$$

More detail on the submatrices of the generalized mass matrix M can be found in the Appendix A, but they are all only dependent on generalized coordinates $\boldsymbol{q}$. The $\boldsymbol{C}_i$ matrices are the sum of the Coriolis and gravity forces for the relevant generalized velocities and are dependent on the generalized coordinates $\boldsymbol{q}$ and velocities $\boldsymbol{v}$. For the following sections we will continue with this partitioned dynamics to design our attitude INDI controller and the cascaded INDI controller.

4-2-3 INDI Attitude Control

For the attitude control we want to control our aerial manipulator state such that a desired body angular acceleration and joint accelerations are achieved. These can be obtained from numerical differentiation of the optimal state sequence of the NMPC. These terms will be the virtual control input in our nonlinear attitude controller. The virtual input for the INDI whole-body attitude control are both the body angular and arm accelerations:

$$\boldsymbol{\nu}_{inner} = \begin{bmatrix} {}_B\dot{\boldsymbol{\omega}}_{b,\text{cmd}} \\ \ddot{\boldsymbol{\eta}}_{\text{cmd}} \end{bmatrix} \tag{4-36}$$

The desired arm accelerations are derived from a PD law:

$$\ddot{\boldsymbol{\eta}}_{\text{cmd}} = k_{p,\text{arm}}(\boldsymbol{\eta}_r - \boldsymbol{\eta}) + k_{d,\text{arm}}(\dot{\boldsymbol{\eta}}_r - \dot{\boldsymbol{\eta}}) \tag{4-37}$$

To achieve this desired attitude and arm accelerations we apply INDI on the dynamics corresponding to these states from the partitioned EoM (4-35):

$$\begin{bmatrix} \boldsymbol{\tau}_b \\ \boldsymbol{\tau}_\eta \end{bmatrix} = \begin{bmatrix} M_{22} & M_{23} \\ M_{23}^T & M_{33} \end{bmatrix} \begin{bmatrix} {}_B\dot{\boldsymbol{\omega}}_b \\ \ddot{\boldsymbol{\eta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_2 \\ \boldsymbol{C}_3 \end{bmatrix} + \begin{bmatrix} M_{12}^T \\ M_{13}^T \end{bmatrix} \dot{\boldsymbol{v}}_b \tag{4-38}$$

The input-output dynamics can be described then as:

$$\dot{\boldsymbol{y}} = \boldsymbol{\alpha}(\boldsymbol{x}, \dot{\boldsymbol{v}}_b) + \boldsymbol{\mathcal{B}}(\boldsymbol{x})\boldsymbol{u} \tag{4-39}$$

with $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{\omega}_b \\ \dot{\boldsymbol{\eta}} \end{bmatrix}$, $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{\tau}_b \\ \boldsymbol{\tau}_\eta \end{bmatrix}$ and

$$\boldsymbol{\alpha}(\boldsymbol{x}, \dot{\boldsymbol{v}}_b) = \begin{bmatrix} M_{22} & M_{32} \\ M_{23}^T & M_{33} \end{bmatrix}^{-1} \left(- \begin{bmatrix} \boldsymbol{C}_2 \\ \boldsymbol{C}_3 \end{bmatrix} - \begin{bmatrix} M_{12}^T \\ M_{13}^T \end{bmatrix} \dot{\boldsymbol{v}}_b \right), \quad \boldsymbol{\mathcal{B}}(\boldsymbol{x}) = \begin{bmatrix} M_{22} & M_{32} \\ M_{23}^T & M_{33} \end{bmatrix}^{-1} \tag{4-40}$$

Whole-body Attitude INDI

Our INDI whole-body control law is then defined as:

$$\begin{bmatrix} \boldsymbol{\tau_b} \\ \boldsymbol{\tau_\eta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau_{b,f}} \\ \boldsymbol{\tau_{\eta,f}} \end{bmatrix} + \begin{bmatrix} M_{22} & M_{23} \\ M_{23}^T & M_{33} \end{bmatrix} \left(\begin{bmatrix} {}_B\dot{\boldsymbol{\omega}}_{b,\text{cmd}} \\ \ddot{\boldsymbol{\eta}}_{\text{cmd}} \end{bmatrix} - \begin{bmatrix} {}_B\dot{\boldsymbol{\omega}}_{b,f} \\ \ddot{\boldsymbol{\eta}}_f \end{bmatrix} \right) + \begin{bmatrix} M_{12}^T \\ M_{13}^T \end{bmatrix} (\dot{\boldsymbol{v}}_b - \dot{\boldsymbol{v}}_{b,f}) \tag{4-41}$$

where we use the filtered measurements $\dot{\boldsymbol{y}}_f$, $\boldsymbol{x_f}$, $\boldsymbol{u}_f$ and $\dot{\boldsymbol{v}}_{b,f}$ and filtered in put $\boldsymbol{\tau_{b,f}}$ to replace part of the nonlinear dynamic model. Also as $\dot{\boldsymbol{v}}_b$ is not directly available, we choose $\dot{\boldsymbol{v}}_b = \dot{\boldsymbol{v}}_{b,\text{cmd}}$, which can also be obtained via numerical differentiation of the optimal NMPC state sequence. We do not have direct access to the body torques, but instead the filtered body torque is extracted from:

$$\begin{bmatrix} \boldsymbol{\tau}_{b,f} \\ T_f \end{bmatrix} = \tilde{\boldsymbol{G}}_1 \boldsymbol{\omega}_{p,f} \circ \boldsymbol{\omega}_{p,f} \tag{4-42}$$

where the filtered propeller speeds $\boldsymbol{\omega}_{p,f}$ are available.

In Section 3-2 INDI was performed using a Taylor expansion around a previous measurement at time t_0. In practice we perform this Taylor expansion around a previous filtered measurement

corresponding to time t_f, where the filter applied is a second-order discrete low pass filter with cut-off frequency f_{att}. It is important that the filter applied to all measurements correspond to the same amount of delay.

(4-41) calculates the desired joint torques also based on the filtered joint torques. In the actual implementation in this thesis we do not use joint torque control, but position control due to the hardware of the DSAM. For that reason we neglect the joint torques in the implementation of this control law.

If we look at (4-41) we notice the term:

$$\boldsymbol{\delta}_{lin} = \begin{bmatrix} M_{12}^T \\ M_{13}^T \end{bmatrix} (\dot{\boldsymbol{v}}_b - \dot{\boldsymbol{v}}_{b,f}) \tag{4-43}$$

which is the effect of the change in linear acceleration on the angular base and arm dynamics. This term does not arise in normal quadrotor control. Given that the input torques and forces are not necessarily continuous (and later in the cascaded INDI control the linear acceleration is even used similarly in the INDI framework) this term can not simply be ignored.

By choosing a sufficiently high sampling time, the derivatives in the state and higher order derivatives in the Taylor expansion of $\dot{\boldsymbol{y}}$ at t_f vanish. In mathematical terms:

$$||\frac{\partial \boldsymbol{\alpha}(\boldsymbol{x}, \dot{\boldsymbol{v}}_b)}{\partial \boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{x}_f) + O(\Delta \boldsymbol{x}^2)|| \approx 0 \tag{4-44}$$

As mentioned in Section 3-2, the intuition behind this assumption is that the states are continuous while the input is not, so it theoretically possible to choose a high enough filtering frequency f such that $\lim_{f\to\infty}(\boldsymbol{x} - \boldsymbol{x}_f) \approx 0$ while the matrix $\frac{\partial \boldsymbol{\alpha}(\boldsymbol{x}, \dot{\boldsymbol{v}}_b)}{\partial \boldsymbol{x}}$ evaluated at t_f remains bounded. We will see that it is however not possible to choose a very high filtering frequency in practice because of noisy accelerometer data. The cut-off frequency of the acceleration measurements is a trade-off between reducing noise and INDI disturbance rejection performance.

A keen eye might notice that (4-41) can also be obtained by applying INDI on the whole-body dynamics, assuming that also the linear acceleration can be controlled:

$$\begin{bmatrix} \boldsymbol{T} \\ \boldsymbol{\tau}_b \\ \boldsymbol{\tau}_\eta \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_f \\ \boldsymbol{\tau}_{b,f} \\ \boldsymbol{\tau}_{\eta,f} \end{bmatrix} + \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{12}^T & M_{22} & M_{23} \\ M_{13}^T & M_{23}^T & M_{33} \end{bmatrix} \left(\begin{bmatrix} \dot{\boldsymbol{v}}_{b,\text{cmd}} \\ {}_B\dot{\boldsymbol{\omega}}_{b,\text{cmd}} \\ \ddot{\boldsymbol{\eta}}_{\text{cmd}} \end{bmatrix} - \begin{bmatrix} \dot{\boldsymbol{v}}_{b,f} \\ {}_B\dot{\boldsymbol{\omega}}_{b,f} \\ \ddot{\boldsymbol{\eta}}_f \end{bmatrix} \right) \tag{4-45}$$

This calculation leads to the same desired torques and additionally a desired thrust. Calculating the thrust in this way however may include non zero terms in the underactuated directions of the quadrotor base, which is infeasible for the actual system. This equivalence however supports the choice of $\dot{\boldsymbol{v}}_b = \dot{\boldsymbol{v}}_{b,\text{cmd}}$ in (4-41). From this control law we obtain the modified desired body torques $\boldsymbol{\tau}_b$ which are used in control allocation as described later.

Decoupled Attitude INDI

This INDI attitude controller uses the whole-body dynamics of the aerial manipulator. In literature it is also common to ignore many coupling terms, such as the dynamic arm forces and the linear acceleration coupling terms [9]. These terms require a precise model of each

component of the aerial manipulator which may not be easily obtained in practice. In decoupled control we assume that the manipulator and aerial base are two separate systems with independent controllers. More commonly, partially coupled control is applied with some information or estimates from the manipulator configuration, which can be included in the floating base dynamics. The use of INDI could help to reject coupling disturbances in the decoupled or partially coupled control framework. For this reason also a partially coupled INDI controller is proposed which does use the state-dependent total inertia of the aerial manipulator, but does not include dynamic interaction forces between the linear dynamics and the arm dynamics.

We start again with our dynamics but now only the body angular dynamics are evaluated:

$$\boldsymbol{\tau}_b = M_{22}\,{}_B\dot{\boldsymbol{\omega}}_b + M_{12}^T\dot{\boldsymbol{v}}_b + M_{23}\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_2 \tag{4-46}$$

We look closer at the submatrices, and specifically M_{22} which is derived explicitly in Appendix A:

$$M_{22} = {}_B\mathcal{I}_b + {}_B\mathcal{I}_a - m_a[{}_B\boldsymbol{p}_{ba}]_\times[{}_B\boldsymbol{p}_{ba}]_\times \tag{4-47}$$

This matrix is the total inertia of the DSAM expressed in the base frame B and is the same as adding the inertia of the two bodies and applying the parallel axis theorem for the offset in arm Center of Gravity (CoG). This equation exactly describes the Euler's rotation equation for the base where $\boldsymbol{\delta}_{arm} = M_{12}^T\dot{\boldsymbol{v}}_b + M_{23}\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_2$ contain all disturbance moments due to the presence of the manipulator. Now in the decoupled approach we will assume $\boldsymbol{\delta}_{arm} = 0$. Note that M_{22} still uses information on the joint angles, but we will still refer to this approach as the decoupled attitude controller.

The INDI decoupled attitude control law for the body torques is then defined as:

$$\boldsymbol{\tau}_b = \boldsymbol{\tau}_{b,f} + M_{22}({}_B\dot{\boldsymbol{\omega}}_{b,\text{cmd}} - {}_B\dot{\boldsymbol{\omega}}_{b,f}) \tag{4-48}$$

In the results section, a brief comparison between the decoupled and whole-body attitude INDI controller is done simulation. Unless explicitly stated, a whole-body attitude INDI controller is used in the NMPC-FF-INDI control scheme.

4-3 NMPC-FF-CINDI formulation

The NMPC-FF-CINDI controller will build upon the previously proposed NMPC-FF similar to the NMPC-FF-INDI control scheme. Instead of only using an INDI attitude controller, the NMPC-FF controller is extended with an inner loop cascaded INDI controller. This INDI motion controller will be based on [53]. Again, this cascaded INDI controller was designed for a quadrotor system and we need to adjust it to make it work with our aerial manipulator system.

4-3-1 High-level Overview

In this control scheme the same NMPC-FF as in Section 4-1 is used to generate an optimal state sequence which is then used in our inner loop cascaded INDI controller. The cascaded

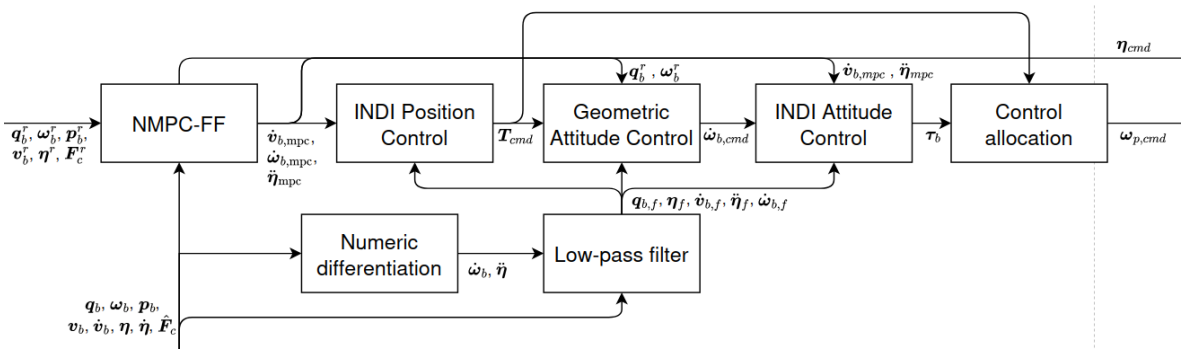


Figure 4-3: A schematic of the NMPC-FF-CINDI control block.

inner loop consists of a INDI position controller and a INDI attitude controller. The INDI attitude controller is the same as in Subsection 4-2-3, but the computation of the virtual control input in this INDI controller will be different. The position controller requires a reference acceleration and outputs a desired thrust. This reference acceleration is obtained using numerical differentiation of the predicted velocity and the current velocity in the optimal state sequence. The position INDI controller is added to add robustness in control of the translational states.

By combining the desired thrust vector from the position INDI controller and the optimal state sequence it is possible to calculate the desired body acceleration, which is the virtual input to the INDI attitude controller. A geometric PD attitude control law is used to obtain the desired angular acceleration. A schematic of the NMPC-FF-CINDI control block is given in Figure 4-3. In the next sections the cascaded INDI control is explained in more detail.

4-3-2 INDI Position Control

To apply INDI for position control of the aerial manipulator we look at dynamics related to the linear position in (4-35). This results in:

$$\boldsymbol{T} = M_{11}\dot{\boldsymbol{v}}_b + M_{12}\,{}_B\dot{\boldsymbol{\omega}}_b + M_{13}\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_1. \tag{4-49}$$

Whole-body Position INDI

Similar to the attitude INDI control described in 4-2-3, we apply a Taylor expansion around $t_{f,\text{outer}}$ and invert the dynamics. This leads to the following INDI formulation:

$$\begin{aligned}\boldsymbol{T}_{\text{cmd}} = \boldsymbol{T}_{f,\text{outer}} &+ M_{11}\cdot(\dot{\boldsymbol{v}}_{b,\text{mpc}} - \dot{\boldsymbol{v}}_{b,f,\text{outer}}) \\ &+ M_{12}({}_B\dot{\boldsymbol{\omega}}_{b,\text{mpc}} - {}_B\dot{\boldsymbol{\omega}}_{b,f,\text{outer}}) + M_{13}(\ddot{\boldsymbol{\eta}}_{\text{mpc}} - \ddot{\boldsymbol{\eta}}_{f,\text{outer}})\end{aligned} \tag{4-50}$$

The commanded accelerations $\dot{\boldsymbol{v}}_{b,\text{mpc}}$, ${}_B\dot{\boldsymbol{\omega}}_{b,\text{mpc}}$ and $\ddot{\boldsymbol{\eta}}_{\text{mpc}}$ are obtained by differentiating the first time step of the optimal state velocities. The filtered measurements ${}_B\dot{\boldsymbol{\omega}}_{b,f,\text{outer}}$, $\ddot{\boldsymbol{\eta}}_{f,\text{outer}}$ and $\dot{\boldsymbol{v}}_{b,f,\text{outer}}$ and filtered input $\boldsymbol{T}_{f,\text{outer}}$ are used, which correspond to time $t_{f,\text{outer}}$. Again $\boldsymbol{T}_{f,\text{outer}}$ is extracted from the simplified reduced control allocation equation:

$$\begin{bmatrix}\boldsymbol{\tau}_{b,f,\text{outer}} \\ T_{f,\text{outer}}\end{bmatrix} = \tilde{\boldsymbol{G}}_1\boldsymbol{\omega}_{p,f,\text{outer}} \circ \boldsymbol{\omega}_{p,f,\text{outer}} \tag{4-51}$$

The same second order discrete time low pass filter is applied on all measurements with cut-off frequency f_{outer}.

For this control layer we assume that the aerial manipulator follows the reference the NMPC generates for the body angular acceleration and the arm accelerations. Again this can also be seen as applying INDI to the whole-body dynamics equations (4-35) and ignoring the torques generated from this control law. This assumption actually does not hold for our implemented cascaded control scheme, as we modify our tracked angular acceleration and joint acceleration based on the desired thrust direction from this control layer. In the results section we will do some experimental tests to study the consequences of this 'whole-body' control with invalid assumption.

Another approach would be to use the decoupled INDI position control law. To derive this decoupled control law, we can start from a simple force balance at the CoG of the quadrotor (note that this does not coincide with the CoG of the aerial manipulator):

$$\sum_i \boldsymbol{F}_i = m\boldsymbol{a} \tag{4-52}$$

This can be described by the following forces:

$$M_{11}\dot{\boldsymbol{v}}_b = \boldsymbol{T} + \boldsymbol{F}_{arm} + \boldsymbol{F}_g \tag{4-53}$$

where $\boldsymbol{T}$ is the thrust force, $\boldsymbol{F}_{arm}$ are all the forces on the base due to the presence of the arm and $\boldsymbol{F}_g$ is the gravity force. If we compare this equation with (4-49), the arm force is then described by the terms $M_{12}\,{}_B\dot{\boldsymbol{\omega}}_b + M_{13}\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_1$. The magnitude of these terms depend on the specific task requirements, the mass of the arm and the distance of the CoG of the arm to the CoG of the base. In literature it is common to assume quasi-static forces from the arm, in other words $\frac{d\boldsymbol{F}_{arm}}{dt} \approx 0$. This assumption limits the applicability of such control laws. However with this assumption, we can construct a decoupled control formulation for position control of the aerial manipulator. This would be given as follows:

$$\boldsymbol{T}_{\text{cmd}} = \boldsymbol{T}_{f,\text{outer}} + M_{11} \cdot (\dot{\boldsymbol{v}}_{b,\text{mpc}} - \dot{\boldsymbol{v}}_{b,f,\text{outer}}) \tag{4-54}$$

The terms $M_{12}\,{}_B\dot{\boldsymbol{\omega}}_b + M_{13}\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_1$ will be lumped together and be approximated by acceleration measurement $\dot{\boldsymbol{v}}_{b,f,\text{outer}}$. The change over time of this force is assumed to be zero. In mathematical terms, the additional INDI assumption for the decoupled control would be:

$$||M_{12}({}_B\dot{\boldsymbol{\omega}} - {}_B\dot{\boldsymbol{\omega}}_{b,f,\text{outer}}) + M_{13}(\ddot{\boldsymbol{\eta}} - \ddot{\boldsymbol{\eta}}_{f,\text{outer}})|| \approx 0 \tag{4-55}$$

for $t - t_{f,\text{outer}} \to 0$, next to the standard assumption.

$$||\frac{\partial \boldsymbol{\alpha}(\boldsymbol{x}, \dot{\boldsymbol{\omega}}_b, \ddot{\boldsymbol{\eta}})}{\partial \boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{x}_{f,\text{outer}}) + O(\Delta x^2)|| \approx 0 \tag{4-56}$$

4-3-3 Geometric Attitude Control

To use the calculated desired thrust from the INDI position control in the angular acceleration command control law we will use a similar approach as described in [55]. The thrust vector $\boldsymbol{T}_{\text{cmd}}$ will be used to construct the desired orientation together with the yaw axis from the

optimal state sequence. The NMPC predicted orientation quaternion of the base at the second time step will be denoted as $\boldsymbol{q}_b^r$. From this we can obtain the rotation matrix corresponding to this quaternion:

$$R_{IB^r} = R(\boldsymbol{q}_b^r) \tag{4-57}$$

where $R(\cdot)$ converts a quaternion to a rotation matrix. We obtain the predicted yaw axis in inertial frame from the first column of R_{IB^R}:

$$\hat{\boldsymbol{x}}_c^r = R_{IB^r}\boldsymbol{e}_1 \tag{4-58}$$

By using the cross product between the normalized thrust vector and the yaw axis we can calculate the last axis:

$$\hat{\boldsymbol{y}}_b^r = \frac{\boldsymbol{T}_{\text{cmd}}}{||\boldsymbol{T}_{\text{cmd}}||} \times \hat{\boldsymbol{x}}_c^r \tag{4-59}$$

and to make it a valid orientation we re-project the yaw axis:

$$\hat{\boldsymbol{x}}_b^r = \hat{\boldsymbol{y}}_b^r \times \hat{\boldsymbol{z}}_b^r \tag{4-60}$$

where $\hat{\boldsymbol{z}}_b^r = \dfrac{\boldsymbol{T}_{\text{cmd}}}{||\boldsymbol{T}_{\text{cmd}}||}$. Now the quaternion parametrization of this desired orientation will be referred to as $\boldsymbol{q}_d$.

Now the DSAM is reconfigurable in which the propellors can be tilted. When the rotors are not tilted, meaning they produce thrust only in the body z-direction, there is significantly more roll and pitch authority than heading. Heading can be independently controlled from the tilt and tilt-prioritized control was introduced in [10] for quadrotor control. In [55] the equations for the reduced attitude error are summarized:

$$[q_{e,w}, q_{e,x}, q_{e,y}, q_{e,z}]^T = \boldsymbol{q}_d \otimes \boldsymbol{q}^{-1} \tag{25}$$

$$\tilde{\boldsymbol{q}}_{e,\text{red}} = \frac{1}{\sqrt{q_{e,w}^2 + q_{e,z}^2}} \begin{bmatrix} q_{e,w}q_{e,x} - q_{e,y}q_{e,z} \\ q_{e,w}q_{e,y} + q_{e,x}q_{e,z} \\ 0 \end{bmatrix} \tag{26}$$

$$\tilde{\boldsymbol{q}}_{e,\text{yaw}} = \frac{1}{\sqrt{q_{e,w}^2 + q_{e,z}^2}} \begin{bmatrix} 0 & 0 & q_{e,z} \end{bmatrix}^T. \tag{27}$$

The angular acceleration command used in then be a weighted sum of this reduced attitude error vector and the error in angular velocity $\boldsymbol{\omega}_b^r - \boldsymbol{\omega}_b$, where $\boldsymbol{\omega}_b^r$ is the predicted angular rate at the second time step in the NMPC optimal solution. Finally also a feedforward angular acceleration term ${}_B\dot{\boldsymbol{\omega}}_{b,mpc}$ also obtained from the optimal state sequence of the NMPC is included, leading to the following angular acceleration command:

$${}_B\dot{\boldsymbol{\omega}}_{b,\text{cmd}} = k_{q,\text{red}}\tilde{\boldsymbol{q}}_{e,\text{red}} + k_{q,\text{yaw}}\text{sgn}(q_{e,w})\tilde{\boldsymbol{q}}_{e,\text{yaw}} + K_{\text{rate}}(\boldsymbol{\omega}_b^r - \boldsymbol{\omega}_b) + {}_B\dot{\boldsymbol{\omega}}_{b,mpc} \tag{4-61}$$

where $k_{q,\text{red}}$ and $k_{q,\text{yaw}}$ are the gains for the reduced-attitude tilt and yaw control and K_{rate} is the diagonal weight matrix for the rate control. This angular acceleration command is then used in INDI attitude control as described in Section 4-2-3 instead of the angular acceleration reference from the MPC. In [55] it was found that the feedforward term ${}_B\dot{\boldsymbol{\omega}}_{b,mpc}$ did not influence the physical system. In our work it is included in the simulation as it was able to obtain better performance with this term, however in the physical experiments this feed forward term was also removed.

One key difference between the control schemes augmented with only an attitude INDI controller and the cascaded INDI controller is the effect of the tilt-prioritized PD attitude control layer on the desired orientation. In NMPC-FF-INDI, the INDI attitude controller directly tracks the optimal angular acceleration from the NMPC state trajectory. However, in NMPC-FF-CINDI, the tilt-prioritized control layer modifies the desired angular acceleration command before it is passed to the INDI controller. As a result, the actual body orientation may deviate from the orientation predicted by the NMPC.

This discrepancy must be accounted for in the arm control. Since the joint angle references computed by the NMPC are based on the assumption that the NMPC-predicted orientation is being tracked, a correction term is required to compensate for the modified attitude command. As only the direction of the end-effector in the NMPC is of importance and also for the tasks in this work, only the first joint is considered. For future work where the exact end-effector orientation should be tracked, including roll around the arm x-axis, this part would also need to be modified. The NMPC-commanded joint angles $\eta_{1,r,\text{MPC}}$ are adjusted as follows:

$$\eta_{1,r} = \eta_{1,r,\text{MPC}} - \boldsymbol{e}_2^T \frac{{}_B\dot{\boldsymbol{\omega}}_{b,\text{cmd}} - {}_B\dot{\boldsymbol{\omega}}_{b,\text{mpc}}}{k_{p,\text{arm}}} \tag{4-62}$$

where ${}_B\dot{\boldsymbol{\omega}}_{b,\text{cmd}}$ is the angular acceleration command generated by the tilt-prioritized cascaded controller, ${}_B\dot{\boldsymbol{\omega}}_{b,\text{mpc}}$ is the angular acceleration from the NMPC trajectory, and $k_{p,\text{arm}}$ is the proportional gain of the arm controller. This correction takes into account the attitude deviation introduced by the tilt-prioritized layer in the joint motion, ensuring that the arm motion remains consistent with the actual body orientation rather than the NMPC-predicted one.

4-4 Control Allocation Implementation

The current outputs of our control schemes are the desired body torques, joint torques and body forces. Our actual control input are the propellor speeds and these can be obtained using control allocation. Consider the reduced control allocation equation derived in Section 2-4 where the propeller gyroscopic terms are neglected:

$$\begin{pmatrix} {}_B\boldsymbol{\tau} \\ ||\,{}_B\boldsymbol{F}\,|| \end{pmatrix} = \tilde{\boldsymbol{G}}_1\boldsymbol{\omega}_p \circ \boldsymbol{\omega}_p + \tilde{\boldsymbol{G}}_2\dot{\boldsymbol{\omega}}_p \tag{4-63}$$

where $\boldsymbol{\omega}_p$ is the vector of propeller speeds. This relation between body torques and thrust and the propellor speeds is nonlinear. This nonlinear system is solved directly using Newton's method, while approximating the motor inertial torques as a first order linear system similar to [55].

Note that the control allocation only solves for the body z-thrust and the three body torques, resulting in four equations for four unknown motor speeds. Due to the tilted rotor configuration of the DSAM, each rotor also generates a force component in the body y-direction. Including this additional force component would result in an overdetermined system of five desired outputs with only four control inputs, which in general has no exact solution. Therefore, the body y-force is omitted from the allocation. This simplification is taken as the body z-thrust is the primary contributor to the total lift and is thus more important to track accurately.

The unmodeled body y-force contributions are small relative to the body z-thrust due to the small tilt angles. The resulting neglected force will be handled as disturbance in the outer loop controllers.

Define the vector of desired body torques and thrust in the body z direction as:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\tau}_{b,\text{des}} \\ T_{\text{des}} \end{bmatrix} \tag{4-64}$$

The goal of the control allocation is to find the motor speeds $\boldsymbol{\omega}_p$ that satisfy:

$$\boldsymbol{\mu} = \tilde{\boldsymbol{G}}_1 \boldsymbol{\omega}_p \circ \boldsymbol{\omega}_p + \tilde{\boldsymbol{G}}_2 \dot{\boldsymbol{\omega}}_p \tag{4-65}$$

The motor dynamics are approximated as a first-order system, such that the angular acceleration of the propellors can be expressed as:

$$\dot{\boldsymbol{\omega}}_p \approx \boldsymbol{K}_m^{-1}(\boldsymbol{\omega}_p - \boldsymbol{\omega}_{p,f,\text{allo}}) \tag{4-66}$$

where $\boldsymbol{K}_m^{-1} = \text{diag}(k_{m,1}^{-1},\, k_{m,2}^{-1},\, k_{m,3}^{-1},\, k_{m,4}^{-1})$ is the diagonal matrix of inverse motor time constants and $\boldsymbol{\omega}_{p,f,\text{allo}}$ are the filtered motor speeds. For $\boldsymbol{\omega}_{p,f,\text{allo}}$ we again apply a second-order low pass filter on the propeller speeds with a cut-off frequency f_{allo}. The time constants of each of the motors can be experimentally identified by applying a step reference rotor speed to each of the propellers and measuring the rise time in the step response.

Substituting this into the allocation equation the nonlinear mapping to be solved becomes:

$$\boldsymbol{F}(\boldsymbol{\omega}_p) = -\boldsymbol{\mu} + \tilde{\boldsymbol{G}}_1\, c_t\, \boldsymbol{\omega}_p^{\circ 2} + \boldsymbol{K}_m^{-1}\tilde{\boldsymbol{G}}_2(\boldsymbol{\omega}_p - \boldsymbol{\omega}_{p,f,\text{allo}}) = \boldsymbol{0} \tag{4-67}$$

This nonlinear equation is solved iteratively using Newton's method. The Jacobian of $\boldsymbol{F}$ with respect to $\boldsymbol{\omega}_p$ is:

$$\boldsymbol{J}(\boldsymbol{\omega}_p) = 2\,\tilde{\boldsymbol{G}}_1\, c_t\, \text{diag}(\boldsymbol{\omega}_p) + \boldsymbol{K}_m^{-1}\tilde{\boldsymbol{G}}_2 \tag{4-68}$$

The Newton update step is then:

$$\boldsymbol{\omega}_p^{(k+1)} = \boldsymbol{\omega}_p^{(k)} - \boldsymbol{J}^{-1}(\boldsymbol{\omega}_p^{(k)})\boldsymbol{F}(\boldsymbol{\omega}_p^{(k)}) \tag{4-69}$$

The iteration is initialized with the motor speeds $\boldsymbol{\omega}_p$ obtained when neglecting $\boldsymbol{G}_2$ and directly inverting the control allocation matrix $\boldsymbol{G}_1$ in (4-65). The algorithm terminates when convergence is reached:

$$\|\boldsymbol{\omega}_p^{(k+1)} - \boldsymbol{\omega}_p^{(k)}\| < \epsilon \tag{4-70}$$

where $\epsilon = 10^{-3}$ is the chosen convergence tolerance. A maximum of $n = 10$ iterations is allowed. If the algorithm does not converge, instead the propellor speeds are taken by directly inverting the reduced control allocation $\tilde{\boldsymbol{G}}_1$ matrix.

4-5 Summary

In this section the NMPC-FF, NMPC-FF-INDI and NMPC-FF-CINDI controllers were formally introduced. Finally also a control allocation algorithm for the INDI-based schemes was introduced. The proposed controllers will be compared in simulation and real world experiments to study the effectiveness of INDI in presence of disturbances and uncertainties for the DSAM. This will be explored in the next Chapter.

Chapter 5

Experimental Results

In this chapter we will describe the implementation details and experiments that were done in this thesis related to the research goal of investigating the benefit of using Incremental Nonlinear Dynamic Inversion (INDI) in aerial manipulator control for simultaneous force and end-effector pose tracking. Three main controller will be compared: The baseline Nonlinear Model Predictive Control (NMPC)-FF, NMPC-FF-INDI and NMPC-FF-CINDI. The controllers are described in Chapter 4.

Experiments will be done in free flight and in Aerial Physical Interaction (APhI) tasks in both simulation and in real world experiments. In simulation the effect of disturbances and model mismatch will be studied on the three controllers. Also different friction parameters on the contact surface will be tested to study robustness of the controllers in the aerial sliding task. A brief comparison with the NMPC formulation of [59] will be done. Ablation of certain contact terms in the our proposed NMPC formulation will be studied in APhI tasks. Finally whole-body and decoupled inner loop control will be briefly compared in simulation.

In the real world experiments the controller designs are validated and compared to each other with similar free flight experiments and aerial writing tasks. The aerial writing tasks will be done using a marker attached to the end-effector on a whiteboard and a crayon attached to the end-effector on a blackboard. Also external wind will be added to show the robust tracking performance of the proposed controllers. The different surfaces are chosen to show sliding is possible on surfaces with different friction properties.

5-1 Simulation Results

5-1-1 Implementation Details

The physical parameters of the Differential Shoulder Aerial Manipulator (DSAM) are given in Table 5-1 and are used as nominal model. The simulations will be done in the Matlab SimScape environment [58]. The model is built using the SimScape Multibody package.

Table 5-1: Nominal parameters of the DSAM used in simulation.

Parameter	Value	Unit
Quadrotor Center of Gravity (CoG) to root ($_B\boldsymbol{r}_{br}$)	$[0.045, 0, -0.01]^T$	m
Root to arm CoG ($_A\boldsymbol{r}_{ra}$)	$[0.075, 0, 0]^T$	m
Root to end-effector ($_A\boldsymbol{r}_{re}$)	$[0.19, 0, 0]^T$	m
Quadrotor Mass (m_B)	0.775	kg
Quadrotor Moment of Inertia ($\mathcal{I}_b$)	$\text{diag}(0.00254, 0.00220, 0.00515)$	kg·m^2
Arm Moment of Inertia ($\mathcal{I}_a$)	$\text{diag}(0.001, 0.001, 0.001)$	kg·m^2
Arm Mass (m_A)	0.2	kg
Rotor tilt angles ($\boldsymbol{\theta}$)	$[-15, 15, 15, -15]^T$	degrees

The controllers require solving a nonlinear Optimal Control Problem (OCP) each timestep. For solving the OCP each iteration we use the Acados package [60]. The NMPC control frequency is $f_{\text{outer}} = 100$ Hz. The control and prediction horizon is $N = 20$ time steps with one time step chosen as $\Delta t_{\text{NMPC}} = 0.05$ s. The control frequency of the inner loops can run faster than the NMPC. A control frequency of $f_{\text{inner}} = 500$ Hz is chosen for the inner loop.

We assume that position, orientation, joint angles, velocities and linear acceleration can be directly measured in the simulation with added Gaussian noise. This would simulate the information we can obtain from a Inertial Measurement Unit (IMU) and motion capture system, which will be used in the real world experiments. For both INDI control and the external wrench estimate we need acceleration data. The angular acceleration data is obtained by numerically differentiating the angular rate signal from a simulated gyroscope. Even though we numerically differentiate the angular rate signal which would amplify high frequency noise, in reality the linear acceleration data is more noisy than the angular acceleration data in small aerial multi-rotors. To account for the different noise levels, two separate cut-off frequencies will be chosen for the acceleration filters. A second order low-pass filter is chosen with a cut-off frequency of $f_{\text{outer}} = 30$ rad/s for the linear accelerations. Similarly a second order low-pass filter is chosen with a cut-off frequency of $f_{\text{att}} = 60$ rad/s for the angular accelerations and the same cut-off frequency is chosen for propeller speeds filter f_{allo} in the control allocation.

The tuning of the INDI parameters are given in Table 5-2. The tuning of the NMPC weights and parameters are given in Table 5-4. Finally the cut-off frequencies for each of the control layers is given in Table 5-3.

In the experiments described in this Chapter, the reference trajectory will be generated using an algebraic reference trajectory of the end-effector position $\boldsymbol{p}_e^r$, end-effector direction $\boldsymbol{n}_e^r$ and joint angles $\boldsymbol{\eta}^r$. In case of singularities of the joint angle at $\eta_1^r = \pm\frac{1}{2}\pi$ a desired base yaw ψ^r is also necessary for the trajectory generation. The offline trajectory generation is based on the work in [63].

Lastly the DSAM consists of two joints, but in this work we are only interested in the direction $\boldsymbol{n}_e$, which is direction the end-effector is pointing. This makes the second joint η_2 irrelevant for the evaluation of the results in this thesis. Also because the inertia of the arm around this joint angle is very small compared to base, control of this joint angle will have little effect on the results. The reference $\eta_2^r(t) = 0$ is chosen for the second joint for all the experiments, but results related to this joint will not be discussed in the rest of the results.

Table 5-2: PD attitude and angular rate control parameters in simulation

Parameter	Value	Description
$k_{q,\text{red}}$	200	Attitude gain
K_{rate}	$\text{diag}([0,0,0])$	Rate gains (roll, pitch, yaw)
$k_{q,\text{yaw}}$	150	Yaw gain

Table 5-3: Filter values for each of the control layers

Filter	Value	Unit
f_{att}	60	rad s^{-1}
f_{outer}	30	rad s^{-1}
f_{allo}	60	rad s^{-1}

Table 5-4: NMPC weights and parameters in simulation

Parameter	Value
End-Effector Position and Attitude Weights	
$Q_{p,E}$	$\text{diag}([100,100,100])$
$Q_{q,E}$	$\text{diag}([200,200,200])$
Base Position and Velocity Weights	
$Q_{p,B}$	$\text{diag}([10,10,10])$
Q_v	$\text{diag}([1,1,1])$
Attitude and Body Rate Weights	
$Q_{q,B}$	$\text{diag}([5,5,5])$
Q_ω	$\text{diag}([1,1,1])$
Joint Position and Velocity Weights	
Q_η	$\text{diag}([1\cdot10^{-3},10])$
$Q_{\dot{\eta}}$	$\text{diag}([0.1,0.1])$
Contact Force and Control Input Weights	
Q_f	3.0
Q_u	$\text{diag}([0.6,0.6,0.6,0.6])$
Joint Acceleration Weights	
$Q_{\ddot{\eta}}$	$\text{diag}([1.0,1.0])$
Parameters	
$k_{p,\text{arm}}$	100
$k_{d,\text{arm}}$	20
$k_{p,\text{contact}}$	1

5-1-2 Free Flight Experiments

We compare the proposed control schemes augmented with an INDI inner loop with the baseline NMPC-FF controller for an experiment with external disturbances on base. This could for example due to wind gusts. The end-effector should track an eight-like trajectory in free flight. Also some model mismatch is introduced to simulate the uncertainties of reality. We choose to change the actual DSAM arm mass to 0.25 kg, the actual arm inertia to 1.1 times the original arm inertia and the rotor thrust coefficients to 1.1 times the original values.

The end-effector translational trajectory is parametrized as follows:

$$\begin{aligned} e_x(t) &= \sin(2\pi f_r) \\ e_y(t) &= \cos(2\pi f_r)\sin(2\pi f_r) \\ e_z(t) &= 0 \end{aligned} \tag{5-1}$$

where f_r is the loop frequency with chosen $f_r = 0.2$ Hz. The trajectory duration is 20 seconds, with a smooth start and ending, leading to tracking of two full loops with velocities up to 1.7 ms^{-1} and accelerations up to 3.3 ms^{-2}. The end-effector orientation trajectory is parametrized as:

$$\begin{aligned} n_x(t) &= 0 \\ n_y(t) &= 1 \\ n_z(t) &= 0 \end{aligned} \tag{5-2}$$

A visual representation of the reference in 3D can be seen in Figure 5-1 and in Figure 5-3 the base velocities during the experiment are plotted. In Figure 5-2 the end-effector position tracking errors are plotted. In Figure 5-4 the absolute end-effector position error $||\boldsymbol{e}||$ and attitude error ϕ_{err} are shown. Here ϕ_{err} is magnitude of the angular error in the angle-axis representation. Throughout the experiment, several disturbances are applied. The disturbance forces are applied directly on the base. The disturbance force over time is plotted in Figure 5-5. In Figure 5-6 the commanded rotor speeds are plotted over time.

The cascaded INDI inner loop significantly increases disturbance rejection of the NMPC controller without deteriorating tracking performance in this experiment. The attitude INDI inner loop does not seem to improve the performance of the baseline NMPC-FF controller in this experiment. This is because of the disturbance forces are directly applied on the base, so no large disturbance torques are generated. In the rotor speeds plot, the inputs from the two INDI control schemes are slightly more noisy than the baseline NMPC, however it does not seem problematic.

To study purely disturbance rejection more closely, we will also compare the controllers for a disturbance rejection test with a static hover reference. In Figure 5-8 the position trajectory is plotted. The corresponding disturbance force over time is plotted in Figure 5-10. In Figure 5-8 we also notice that the direction in which the disturbance is applied has influence especially for the attitude tracking of the end-effector. This can also be seen in the absolute position and attitude error plots in Figure 5-9. In Figure 5-1 a 3D representation of the end-effector position during the hover experiment is shown.

The disturbance in the world y-direction at $t = 7$ s, which is the body x-direction of the aerial manipulator, has a significant impact on the attitude tracking error ϕ_{err} of the DSAM for the NMPC-FF-CINDI control scheme compared to the other two controllers.

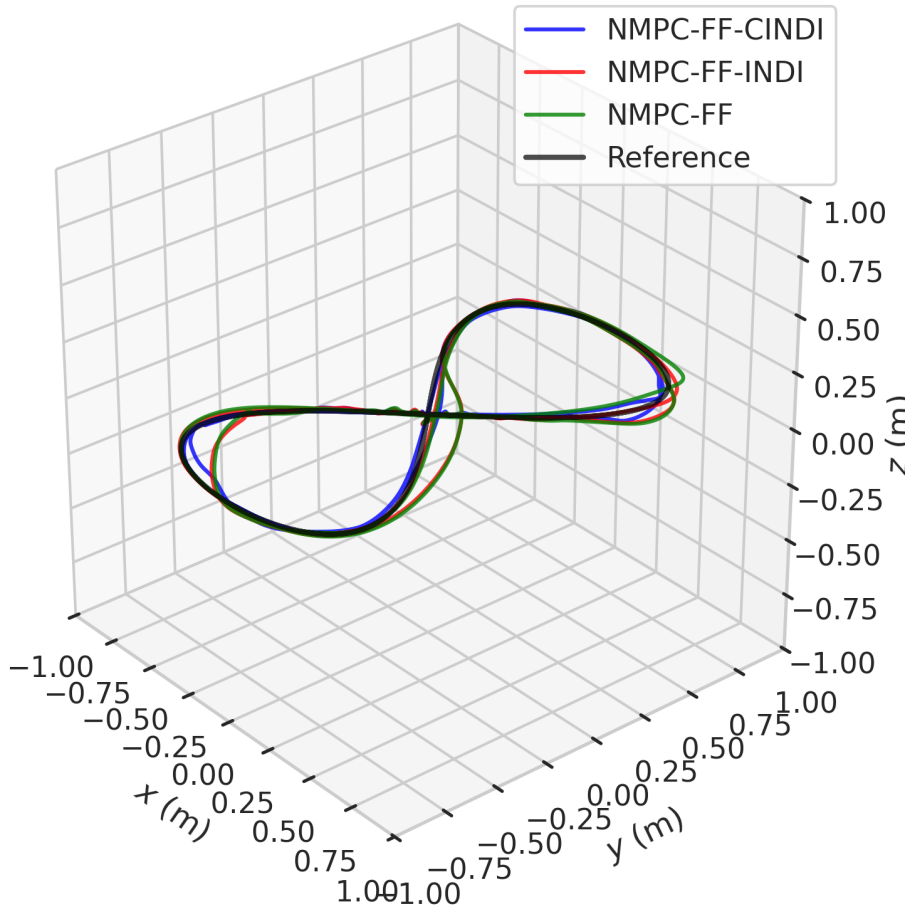


Figure 5-1: 3D representation of the end-effector trajectory during the eight-like free flight experiment with disturbances.

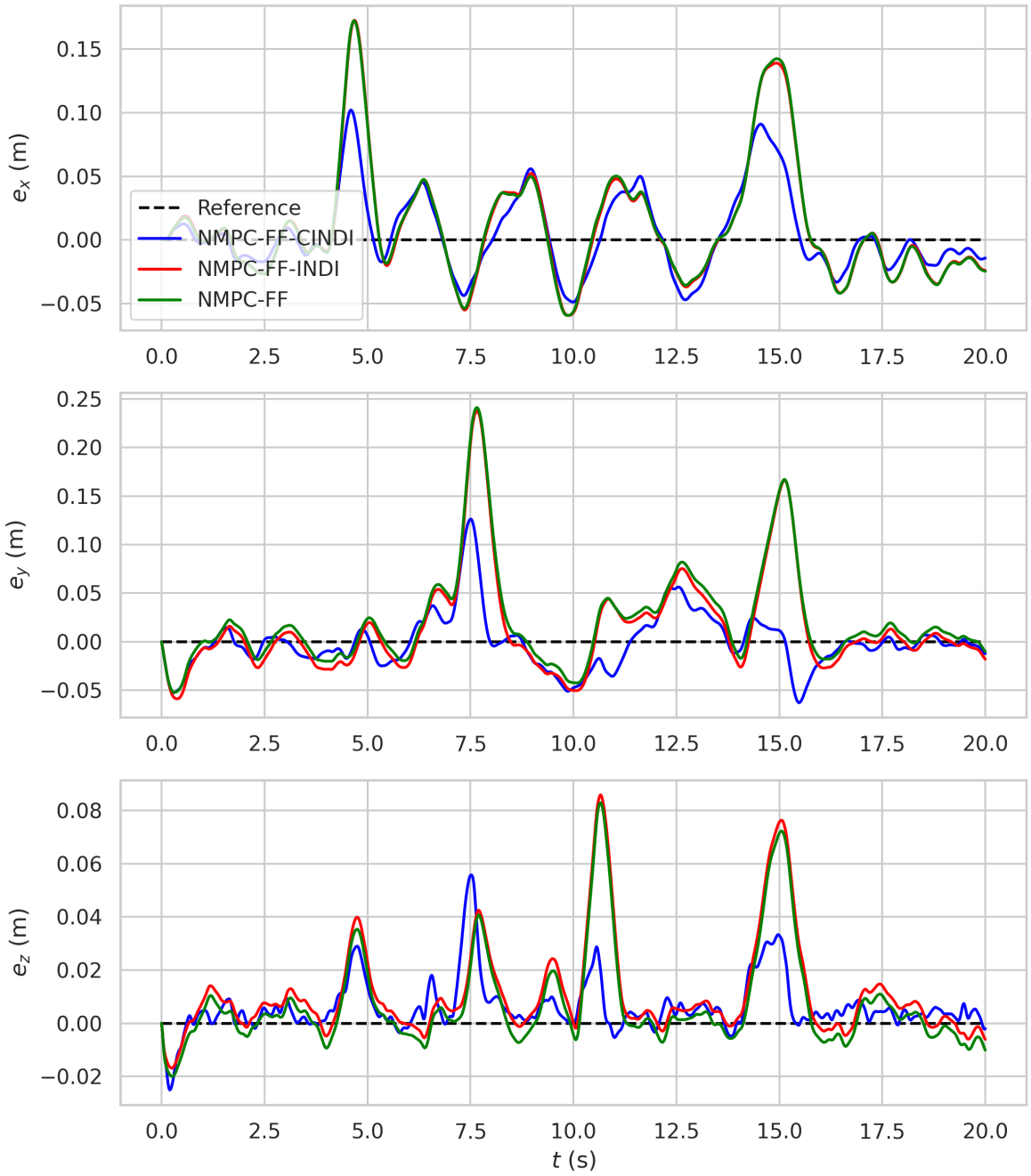


Figure 5-2: The end-effector position tracking errors during the eight-like free flight experiment with disturbances.

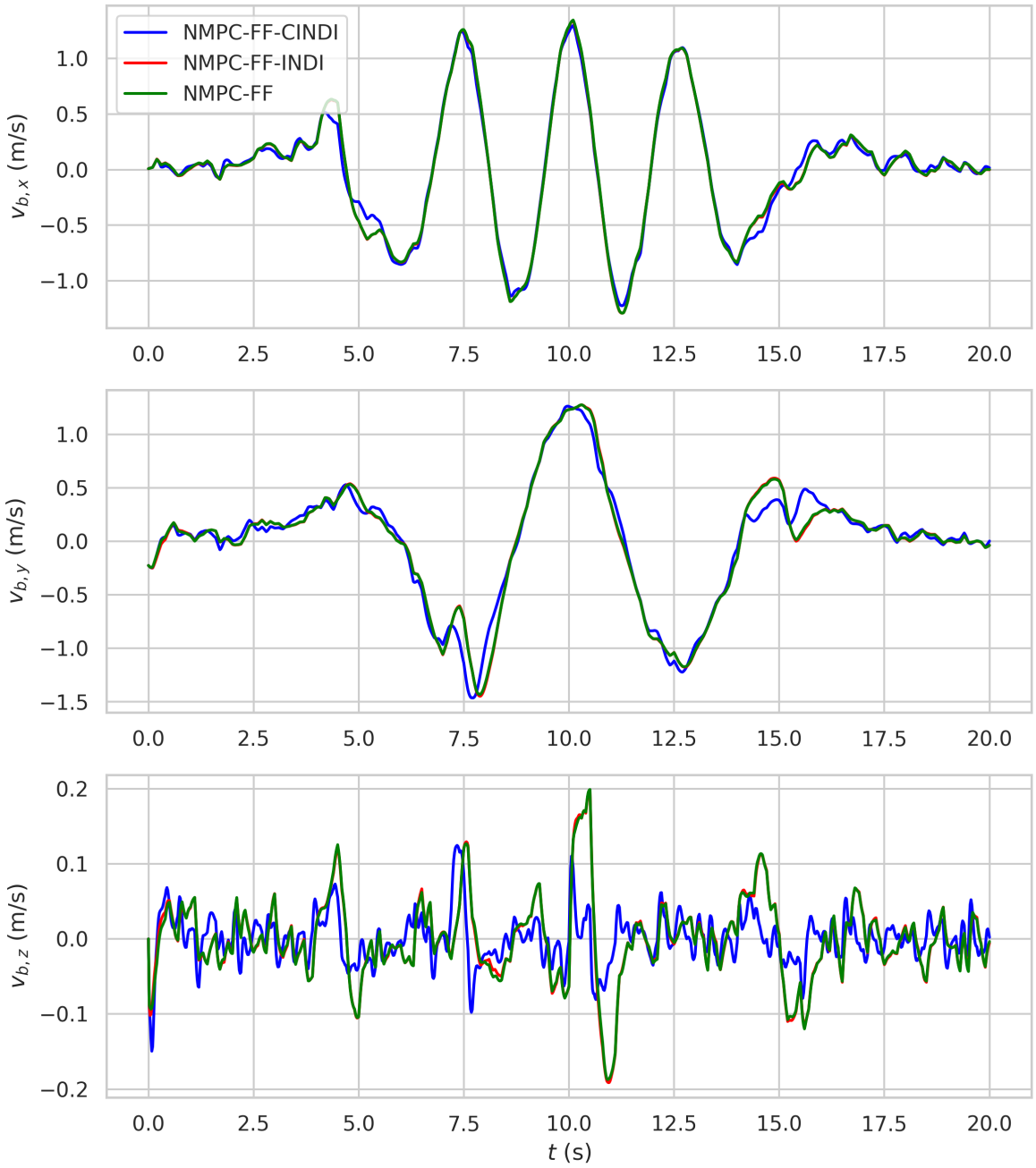


Figure 5-3: The base velocities during the eight-like free flight experiment with disturbances.

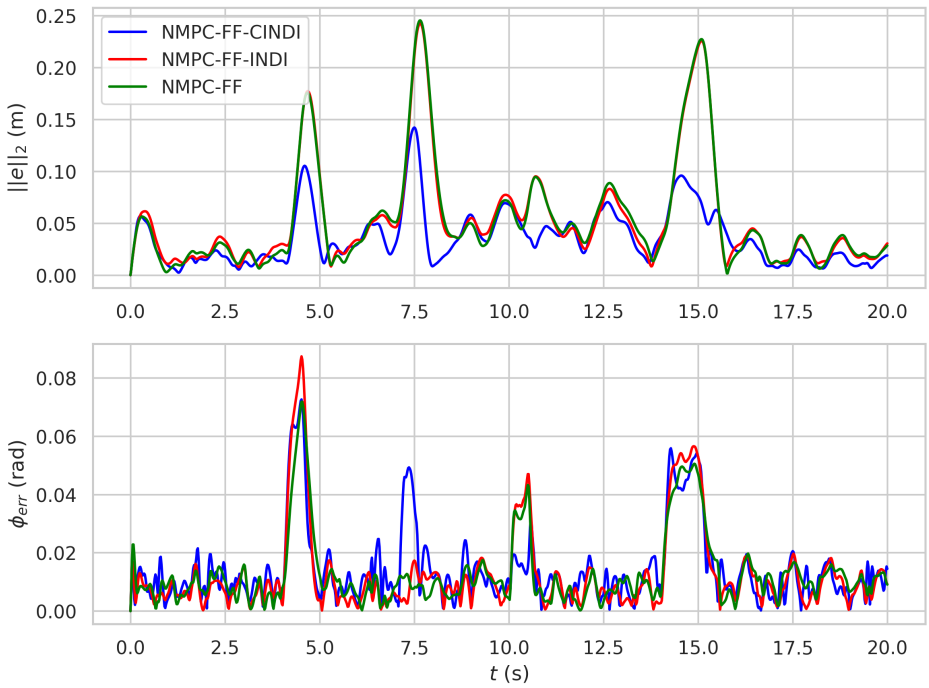


Figure 5-4: The absolute position and attitude tracking error during the eight-like free flight experiment with disturbances.

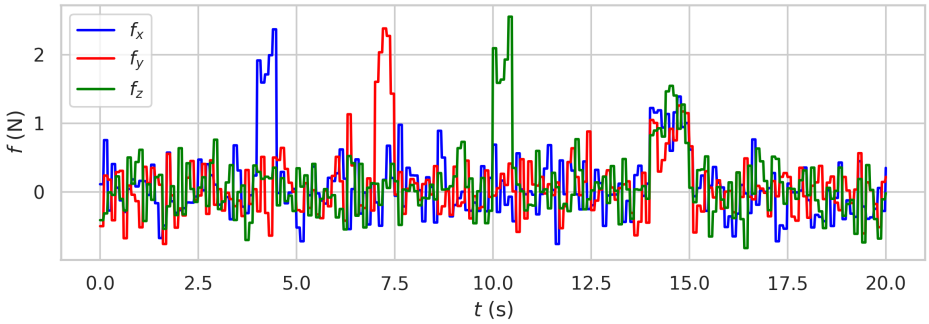


Figure 5-5: The disturbance forces applied during the eight-like free flight experiment.

Figure 5-6: The commanded propellor speeds during the eight-like free flight experiment with disturbances.

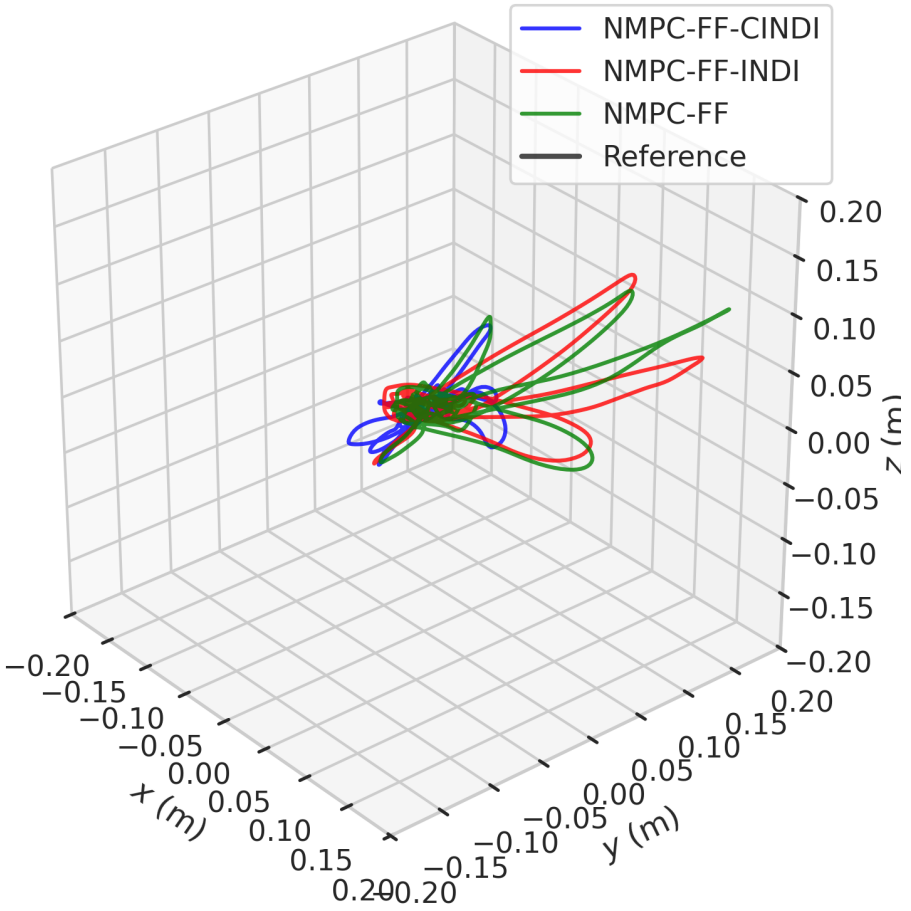


Figure 5-7: 3D representation of the end-effector trajectory during the hover experiment with disturbances.

This observation can be explained. A disturbance from the body x or y body axes is rejected by changing the attitude (the tilt) of the base. The NMPC-FF formulation is aware of the arm and changes its tilt and joint angle accordingly to reject a disturbance force in the body x-direction. Both the NMPC-FF and NMPC-FF-INDI controller thus do not show any significant attitude tracking errors when this disturbance is applied. However the NMPC-FF-CINDI controller has the additional position INDI and geometric attitude control layers. These control layers calculate a new desired thrust vector and modifies the desired attitude of the base obtained from the NMPC. As described in Section 4-3-3 this change in tilt outside of the NMPC is also taken into account by calculating an offset in the first joint. However this is based on the PD arm control model, which is not how it is actually controlled in simulation (and in physical experiments). This mismatch will lead to a larger offset in the end-effector attitude tracking. Removing the offset in η_1 due to changing the tilt of the base in the NMPC-FF-CINDI increased the attitude tracking error significantly more, validating the addition of this term in simulation.

Finally also different tunings of the NMPC controller are tested to show that the cascaded INDI increases disturbance rejection also with other tunings. In Figure 5-12 a boxplot of the

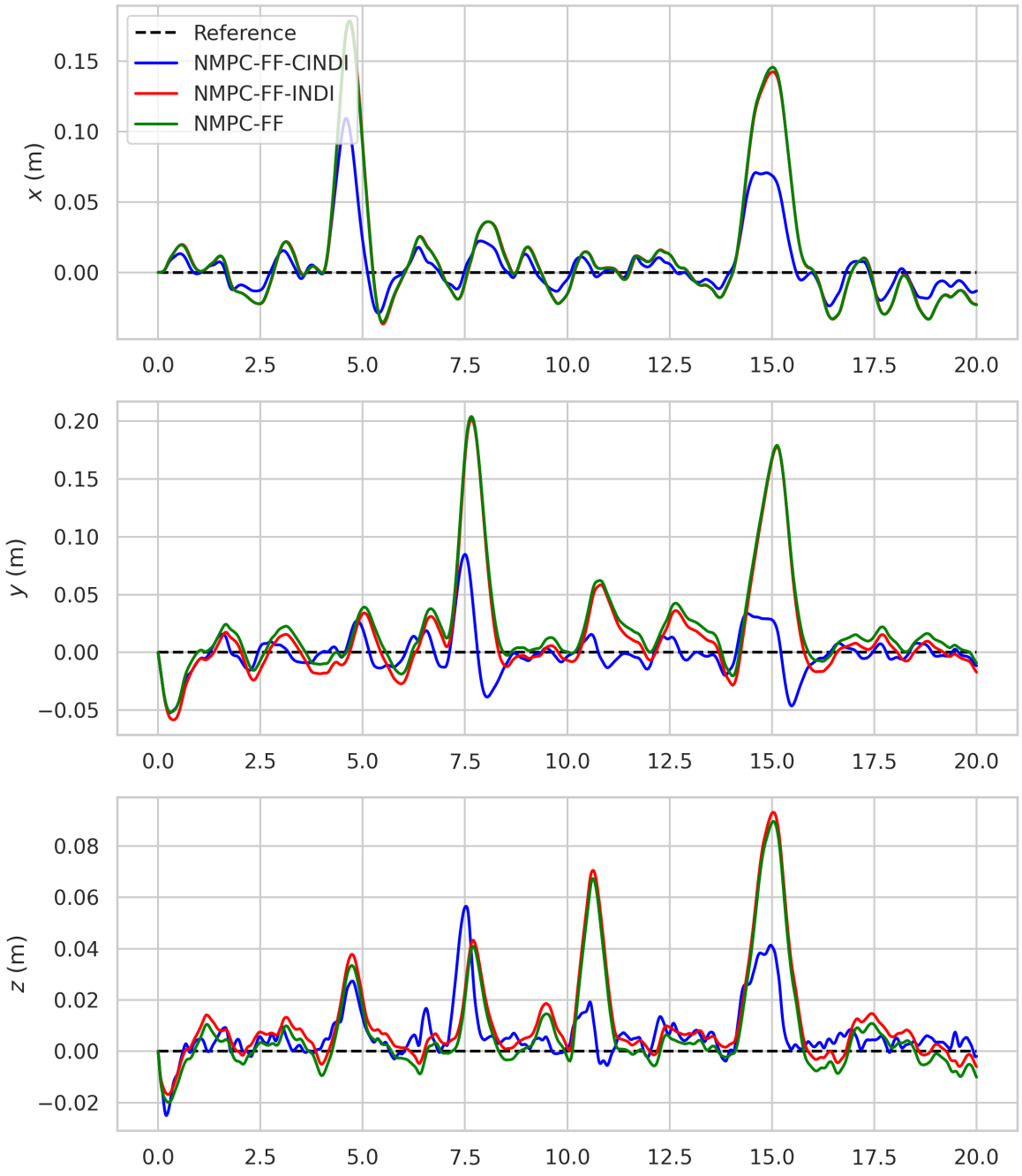


Figure 5-8: The end-effector position trajectory during the hover experiment with disturbances.

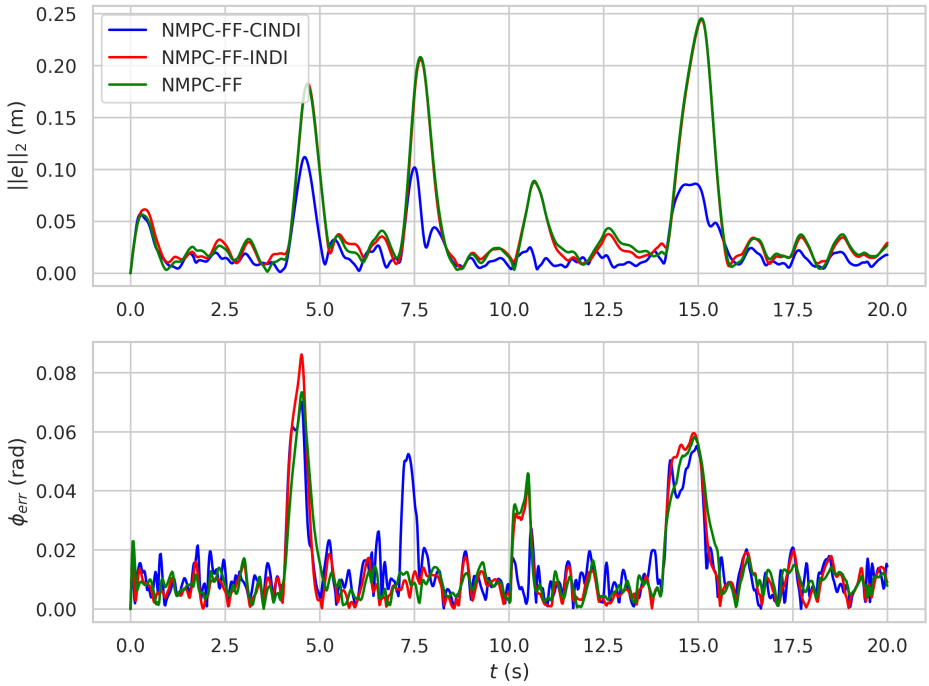


Figure 5-9: The absolute position and attitude error during the end-effector in the hover experiment with disturbances.

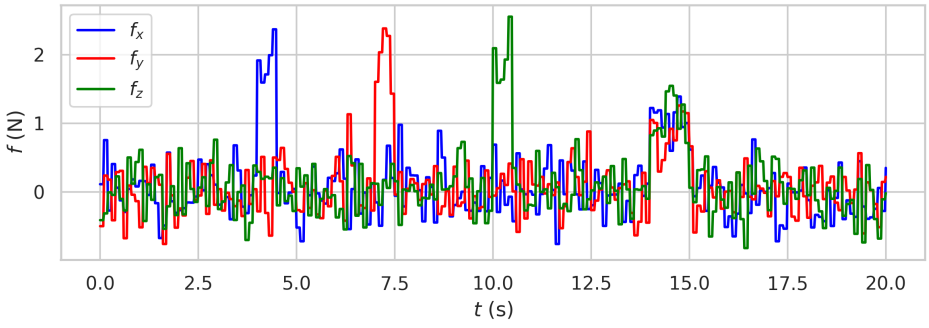


Figure 5-10: The disturbance forces applied during the hover experiment.

Figure 5-11: The commanded propellor speeds during the hover experiment with disturbances.

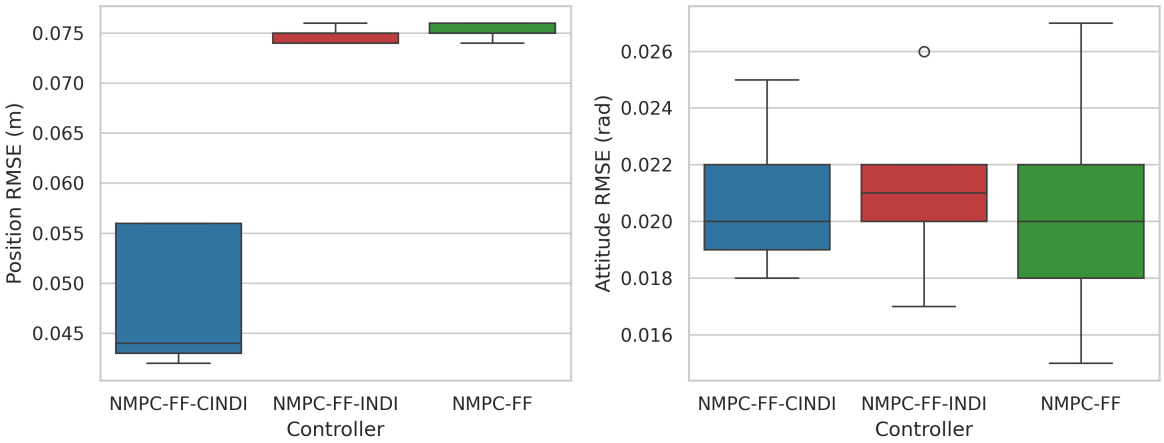


Figure 5-12: RMSE of the position and attitude for different NMPC weights for the eight flight experiment with disturbances.

RMSE of the end-effector position and attitude tracking are shown for five different NMPC weights for the same eight-like flight tracking experiment with disturbances as described before. As can be seen the position RMSE of the NMPC-FF-CINDI control scheme is significantly lower for all weights compared to the other two controllers. The exact changes in NMPC tunings of the experiments are the weights on the end-effector position, base position and end-effector orientation and can be found in Appendix B.

5-1-3 Aerial Writing

In this section APhI sliding experiments are performed in simulation. The experiments will be designed such that the contact force is tracked in different directions of the world frame Σ_I. One experiment will track a contact normal force in the horizontal direction with $\hat{\boldsymbol{e}}_n = [0\ -1\ 0]^T$, another experiment will track contact forces in the vertical direction with $\hat{\boldsymbol{e}}_n = [0\ 0\ 1]^T$ and also lastly an experiment will track contact forces on a 45-degree inclined surface with $\hat{\boldsymbol{e}}_n = \frac{1}{\sqrt{2}}[0\ -1\ 1]^T$. For each experiment we will present the results given a nominal control model with no external disturbances besides contact forces (including friction forces). As described in Section 2-5, friction in simulation is modeled with three main parameters: the static friction coefficient μ_s, the kinetic friction coefficient μ_k and the critical velocity v_{min}. The goal of this work is to perform aerial writing with the DSAM and this will be validated in physical experiments with a whiteboard marker and whiteboard. Although the parameters vary per marker and whiteboard surface, an estimate is made of the parameters to be used in the simulations, where $\mu_s = 0.5$, $\mu_k = 0.3$ and $v_{min} = 0.1$ ms^{-1} are chosen. Lastly the obtained friction f_{fr} and contact normal forces f_n are directly extracted from a rigid body block in SimScape, allowing for direct measurement of the actual contact forces in simulation.

Vertical Surface

In our first experiment we will move the aerial manipulator horizontally towards a vertical surface. Then the DSAM will track a reference trajectory and simultaneously a contact normal force. The end-effector translational trajectory is parametrized as follows:

$$\begin{aligned} e_x(t) &= \begin{cases} 0 & \text{if } t < t_{\text{write}} \\ 0.3\sin(2\pi f_r t) & \text{else} \end{cases} \\ e_y(t) &= \begin{cases} 0.96\sin\left(\dfrac{\pi}{2}\dfrac{t}{t_{\text{move}}}\right) & \text{if } t < t_{\text{move}} \\ 0.96 & \text{else} \end{cases} \\ e_z(t) &= \begin{cases} 0 & \text{if } t < t_{\text{write}} \\ 0.3\cos(2\pi f_r t)\sin(2\pi f_r t) & \text{else} \end{cases} \end{aligned} \tag{5-3}$$

where f_r is the loop frequency with chosen $f_r = 0.1$ Hz. This leads to a trajectory with sliding accelerations up to 0.25 ms^{-2} and velocities up to 0.27 ms^{-1}. The timing parameters are chosen to be $t_{\text{move}} = 3$ s and $t_{\text{write}} = 5$ s. The end-effector orientation trajectory is parametrized as:

$$\begin{aligned} n_x(t) &= 0 \\ n_y(t) &= 1 \\ n_z(t) &= 0 \end{aligned} \tag{5-4}$$

The wall is positioned at $y = 0.95$ with the normal vector of the surface being $\hat{\boldsymbol{e}}_n = -[0\ 1\ 0]^T$. The desired normal contact force is set to 1 N. Also the heading and joint angle η_1 is defined beforehand:

$$\begin{aligned} \psi &= \frac{\pi}{2} \\ \eta_1 &= 0 \end{aligned} \tag{5-5}$$

In Figure 5-13 a visual representation of the 3D end-effector trajectories of this experiment is shown. In Figure 5-14 a side view of the end-effector position during the eight drawing experiment is shown. In Figure 5-15 end-effector tracking errors are shown. In Figure 5-17 the absolute position and attitude tracking error of the end-effector are plotted over time. In Figure 5-18 normal contact forces and friction forces are plotted. In Figure 5-19 the commanded rotor speeds are plotted.

At the instant of contact a large contact force peak is observed for all controllers, but all controllers are able to remain stable. After the initial contact transients, the NMPC-FF has large end-effector pose tracking errors during sliding, while both control schemes with INDI inner loop are significantly less affected. In Figure 5-19 it can be seen that the commanded rotor speed $\omega_{p,2}$ of the NMPC-FF results hit the saturation limit at around $t = 13$ s. This is also where the large tracking error occurs in the experiment. The position error of NMPC-FF-CINDI is lower than NMPC-FF-INDI, where the absolute tracking error reaches up to 7 cm for cascaded INDI and 9 cm for the attitude INDI inner loop.

On average the NMPC-FF-CINDI tracks a force of 1.2 N and the NMPC-FF-CINDI tracks a force of around 1.0 N which is also the reference contact force. Both controllers track this force consistently, except for certain time windows in which the controller seems to behave very noisy for both cascaded INDI and attitude INDI inner loop. This noisy behavior occurs during the peak velocity of the sliding motion and could either be attributed to the undesired controller behavior or simulation inaccuracies. To explore this observation further, the same experiment will be done in real world experiments later in this Chapter. The difference in applied contact force between NMPC-FF-INDI and NMPC-FF-CINDI can be attributed to the position INDI layer, which will add more contact force due to the penetration depth of the reference trajectory. From this experiment it seems the INDI control layers enable the DSAM to simultaneously track end-effector pose and contact forces consistently, while the NMPC-FF controller has significantly degraded tracking performance during sliding.

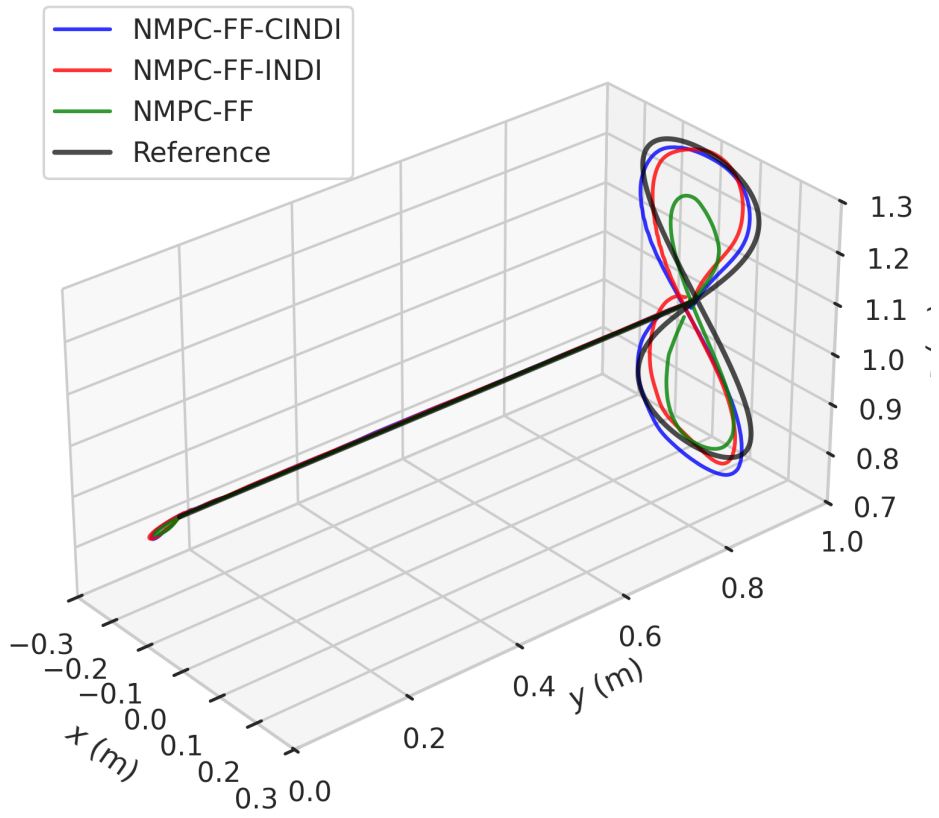


Figure 5-13: 3D representation of the end-effector position trajectory during the eight writing experiment on a vertical surface.

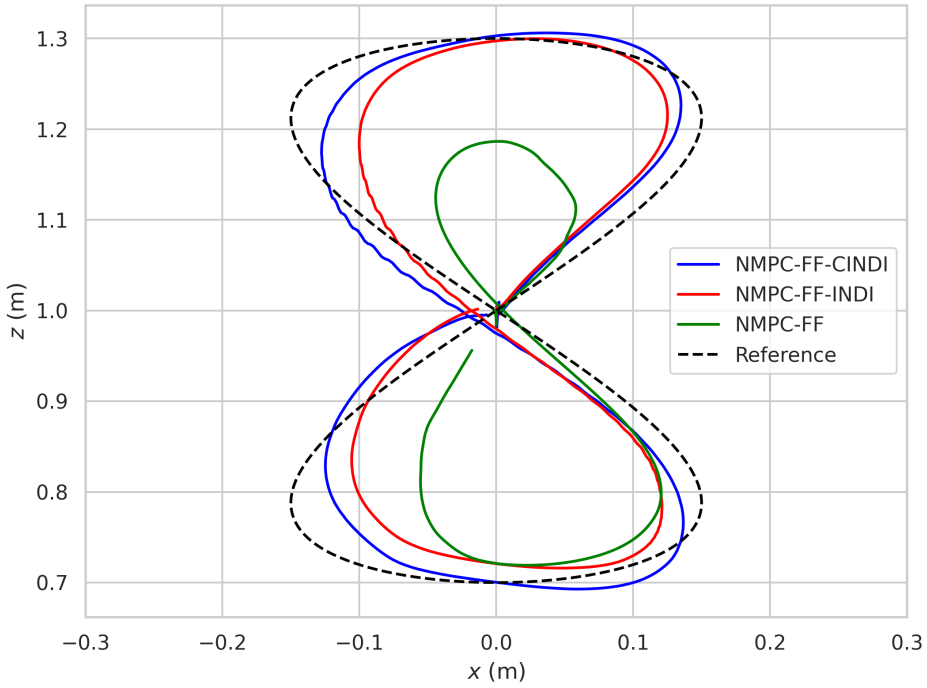


Figure 5-14: Side view of the end-effector position during the eight drawing experiment on a vertical surface.

Horizontal Surface

In our next experiment we will move the aerial manipulator vertically towards a horizontal surface. During contact, the DSAM will again perform a eight writing experiment but now in a different direction. The end-effector position trajectory is parametrized as follows:

$$
\begin{aligned}
e_x(t) &= \begin{cases} 0 & \text{if } t < t_{\text{write}} \\ 0.3\sin(2\pi f_r t) & \text{else} \end{cases} \\
e_y(t) &= \begin{cases} 0 & \text{if } t < t_{\text{write}} \\ 0.3\cos(2\pi f_r t)\sin(2\pi f_r t) & \text{else} \end{cases} \\
e_z(t) &= \begin{cases} 0.3 - 0.3\sin\left(\dfrac{\pi}{2}\dfrac{t}{t_{\text{move}}}\right) & \text{if } t < t_{\text{move}} \\ 0 & \text{else} \end{cases}
\end{aligned}
\tag{5-6}
$$

where f_r is the loop frequency with $f_r = 0.1$ Hz. This leads to a trajectory with reference sliding accelerations up to 0.25 ms^{-2} and velocities up to 0.27 ms^{-1}. The timing parameters are chosen to be $t_{\text{move}} = 3$ s and $t_{\text{write}} = 5$ s. The end-effector orientation trajectory is parametrized as:

$$
\begin{aligned}
n_x(t) &= 0 \\
n_y(t) &= 0 \\
n_z(t) &= -1
\end{aligned}
\tag{5-7}
$$

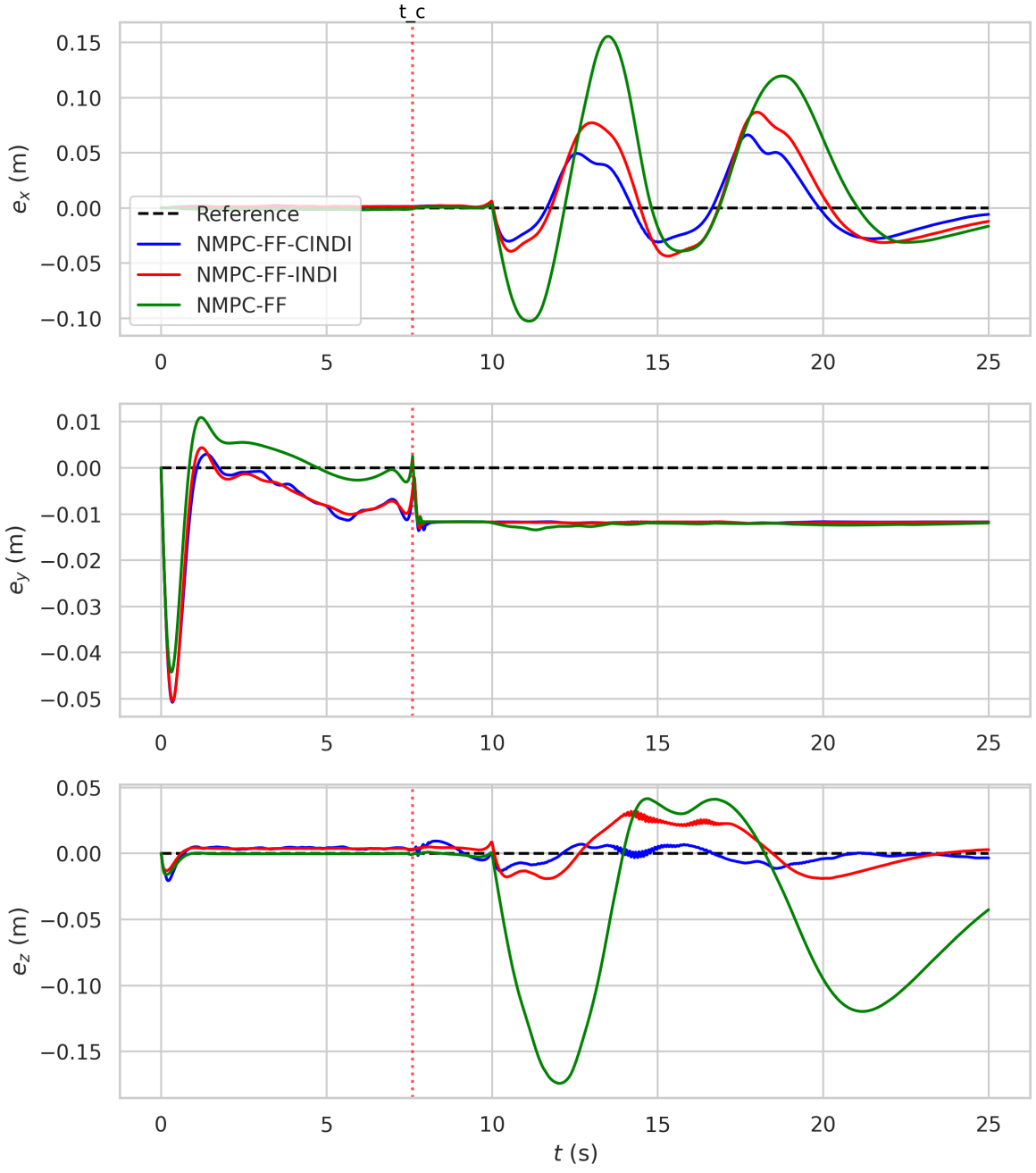


Figure 5-15: The end-effector position tracking error during the eight writing experiment on a vertical surface in simulation.

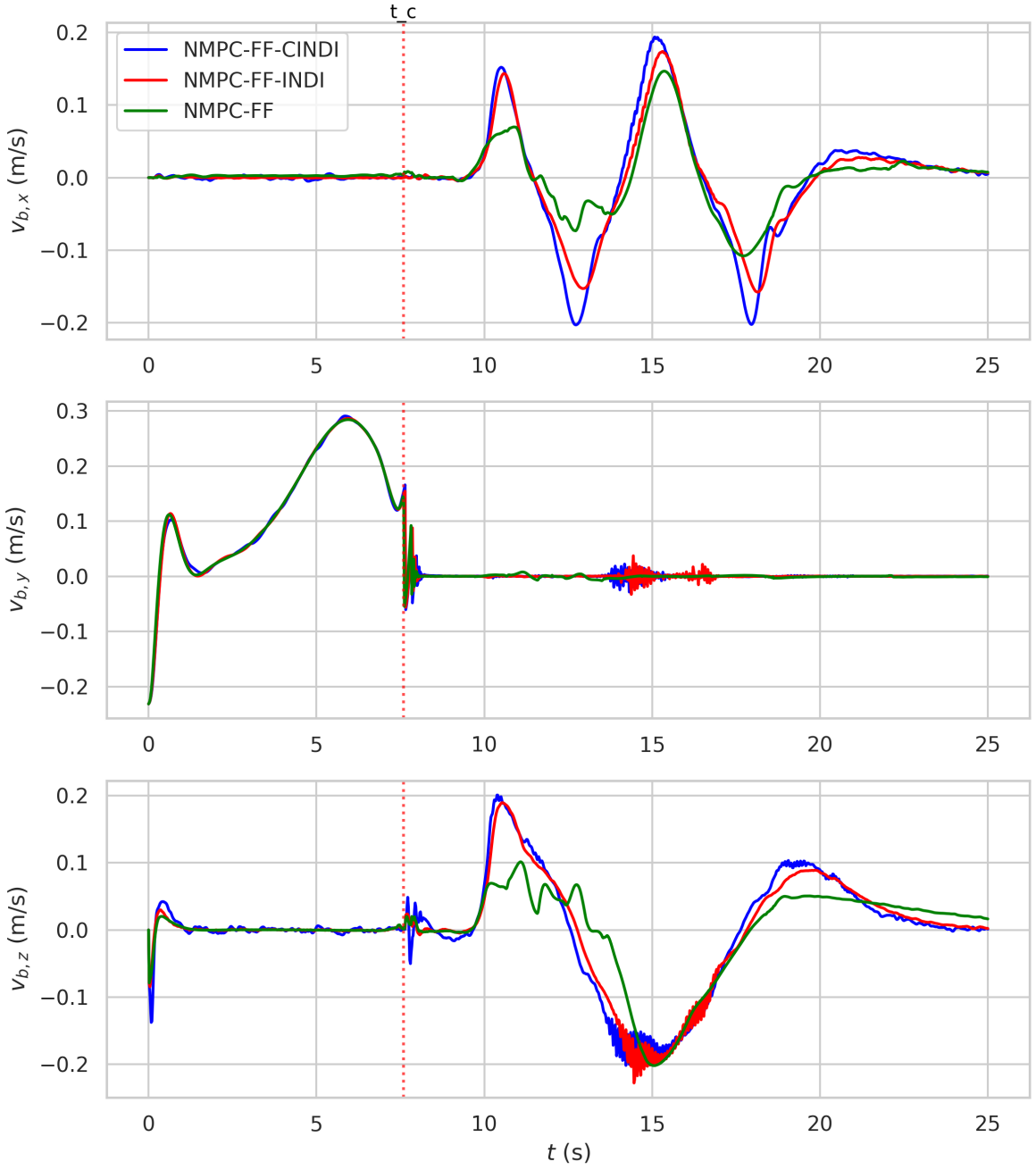


Figure 5-16: The base velocity during the eight writing experiment on a vertical surface in simulation.

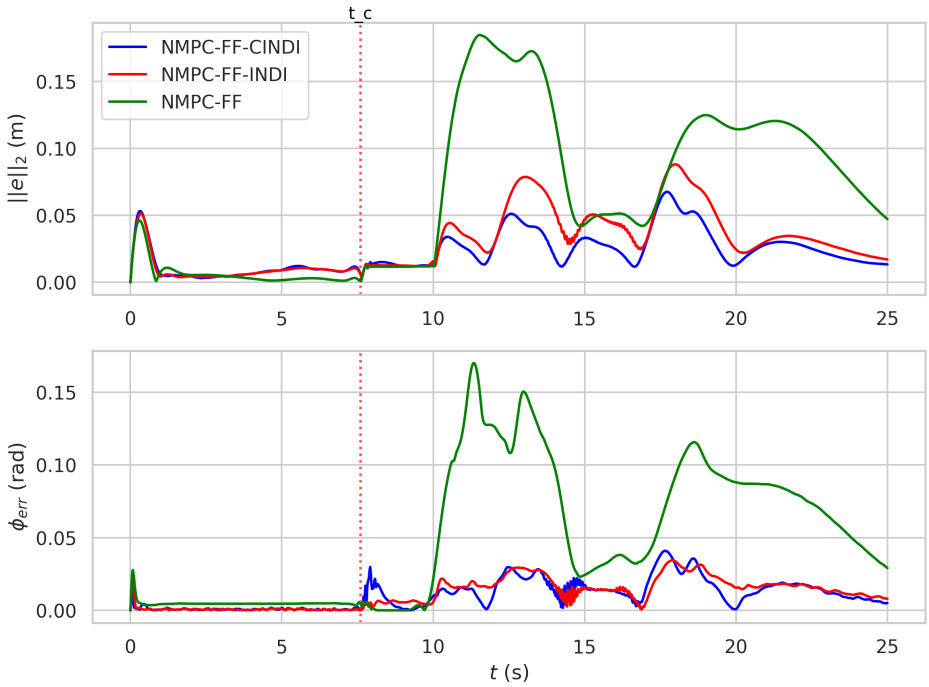


Figure 5-17: The absolute position and attitude error of the end-effector during the eight writing experiment on a vertical surface in simulation.

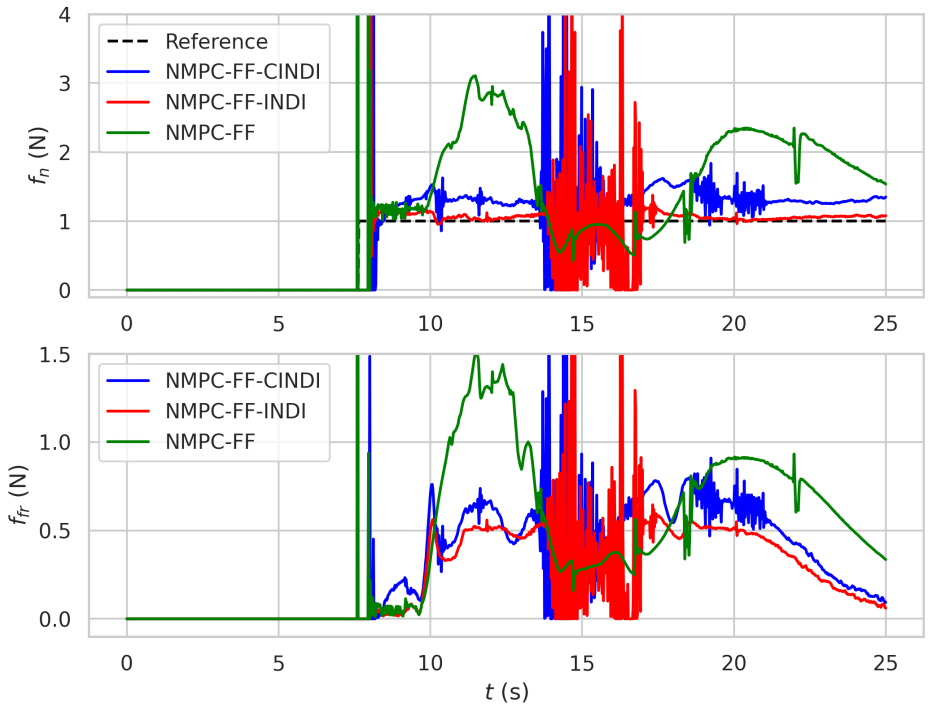


Figure 5-18: The actual contact forces (normal and friction) during the eight writing experiment on a vertical surface in simulation.

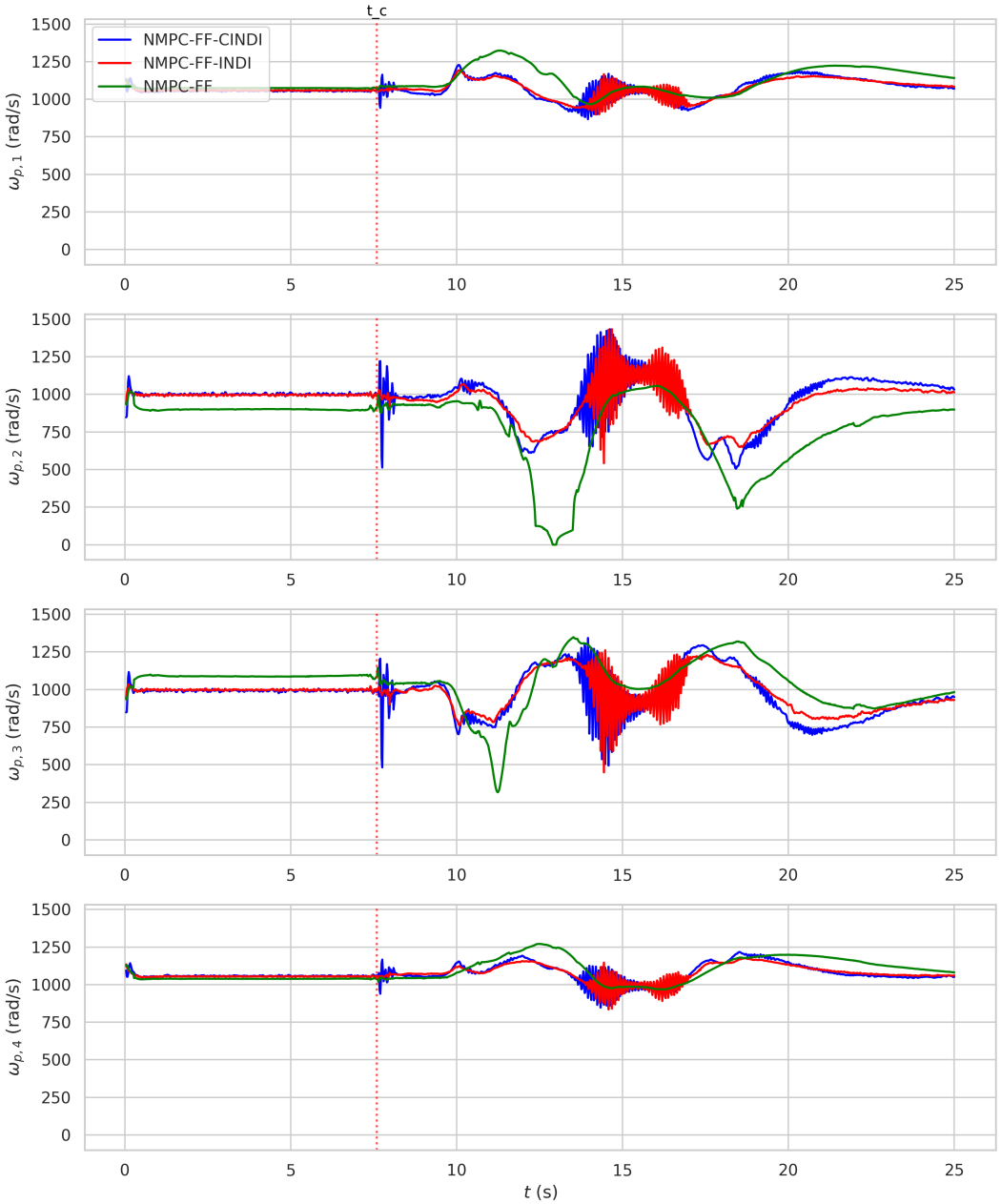


Figure 5-19: The rotor speed commands during the eight writing experiment on a vertical surface in simulation.

The wall is positioned at $z = 0.01$ with the normal vector of the surface being $\hat{\boldsymbol{e}}_n = [0\ 0\ 1]^T$. The contact force being tracked is 1 N. Also the heading and joint angle η_1 is defined beforehand:

$$\begin{aligned} \psi(t) &= \frac{\pi}{2} \\ \eta_1(t) &= \frac{\pi}{2} \end{aligned} \tag{5-8}$$

In Figure 5-20 a visual representation of the end-effector position trajectory during the vertical contact experiment is shown. In Figure 5-22 the absolute end-effector pose tracking errors are plotted over time. In Figure 5-23 the contact forces of the experiment are plotted over time with the desired contact force.

Again all controllers are able to remain stable during this sliding experiment. It seems the baseline NMPC-FF controller has similar absolute pose tracking errors as the NMPC-FF-CINDI control scheme in this experiment, where in Figure 5-22 it can be seen that $||e||_2$ is largest for NMPC-FF-INDI. The force tracking of all controllers do not seem to have large deviations after the initial contact transients. NMPC-FF-CINDI applies an average contact force of around 1.4 N and the other controllers apply slightly lower contact forces while the reference force was set to 1 N.

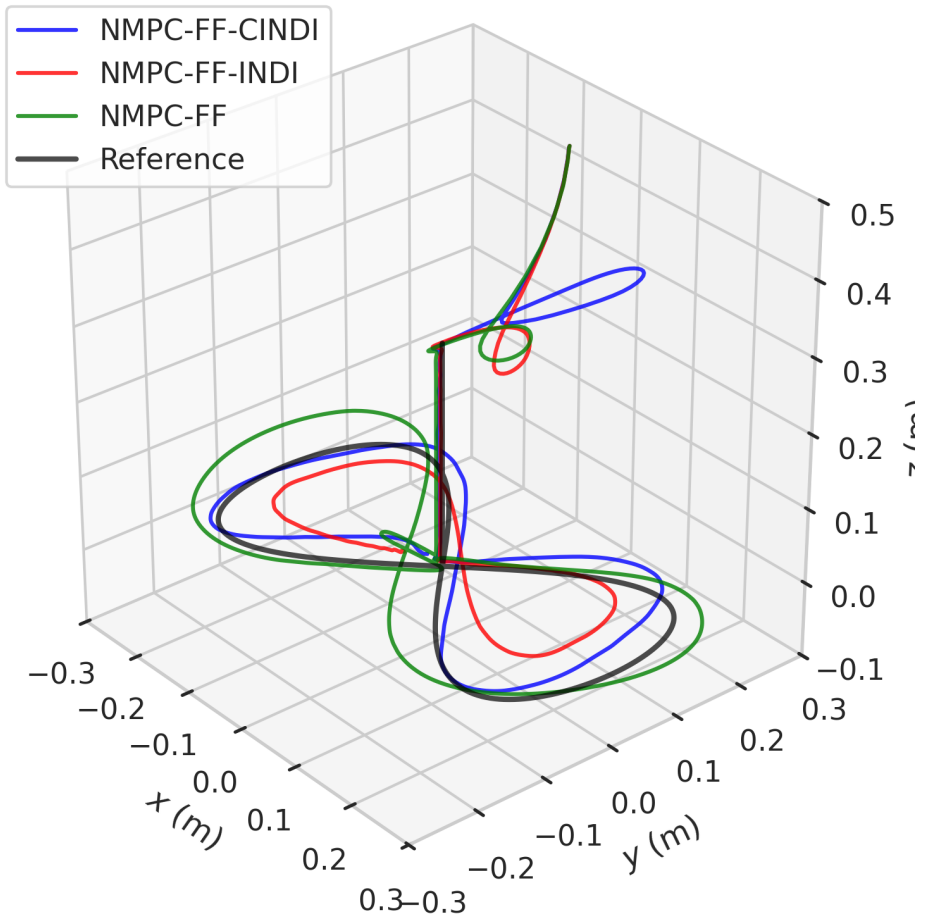


Figure 5-20: 3D representation of the end-effector position trajectory during the vertical contact experiment.

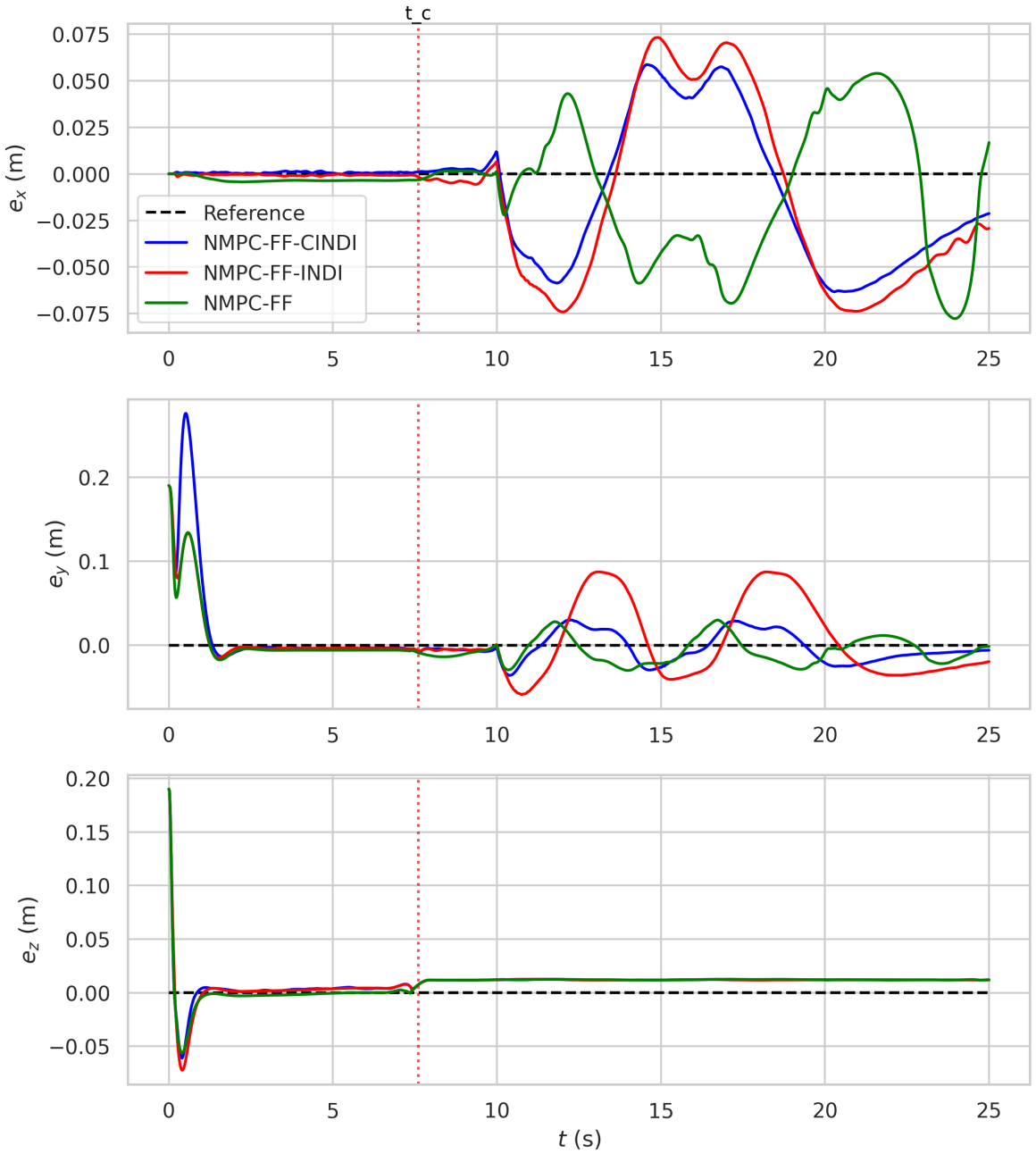


Figure 5-21: The end-effector position tracking error during the vertical contact experiment.

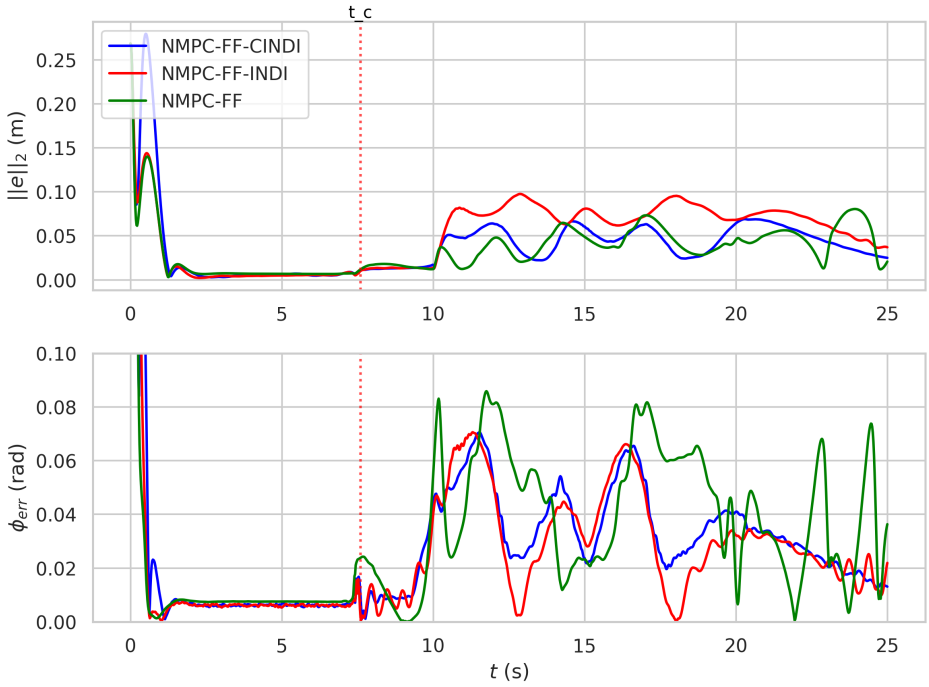


Figure 5-22: The absolute position and attitude error during the end-effector in the vertical contact experiment.

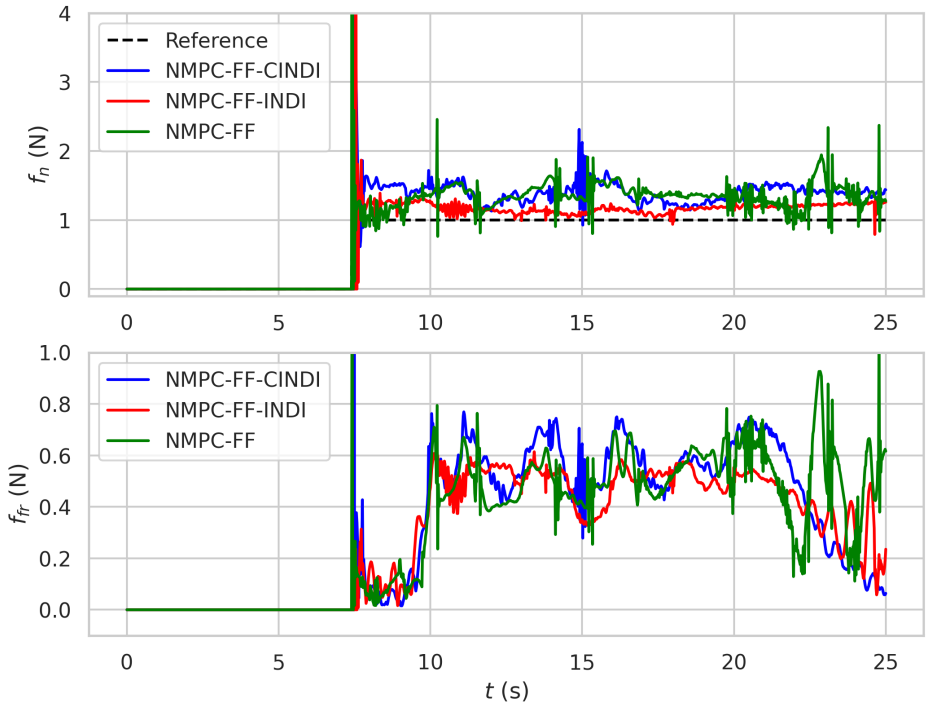


Figure 5-23: The actual contact forces (normal and friction) during the vertical contact experiment.

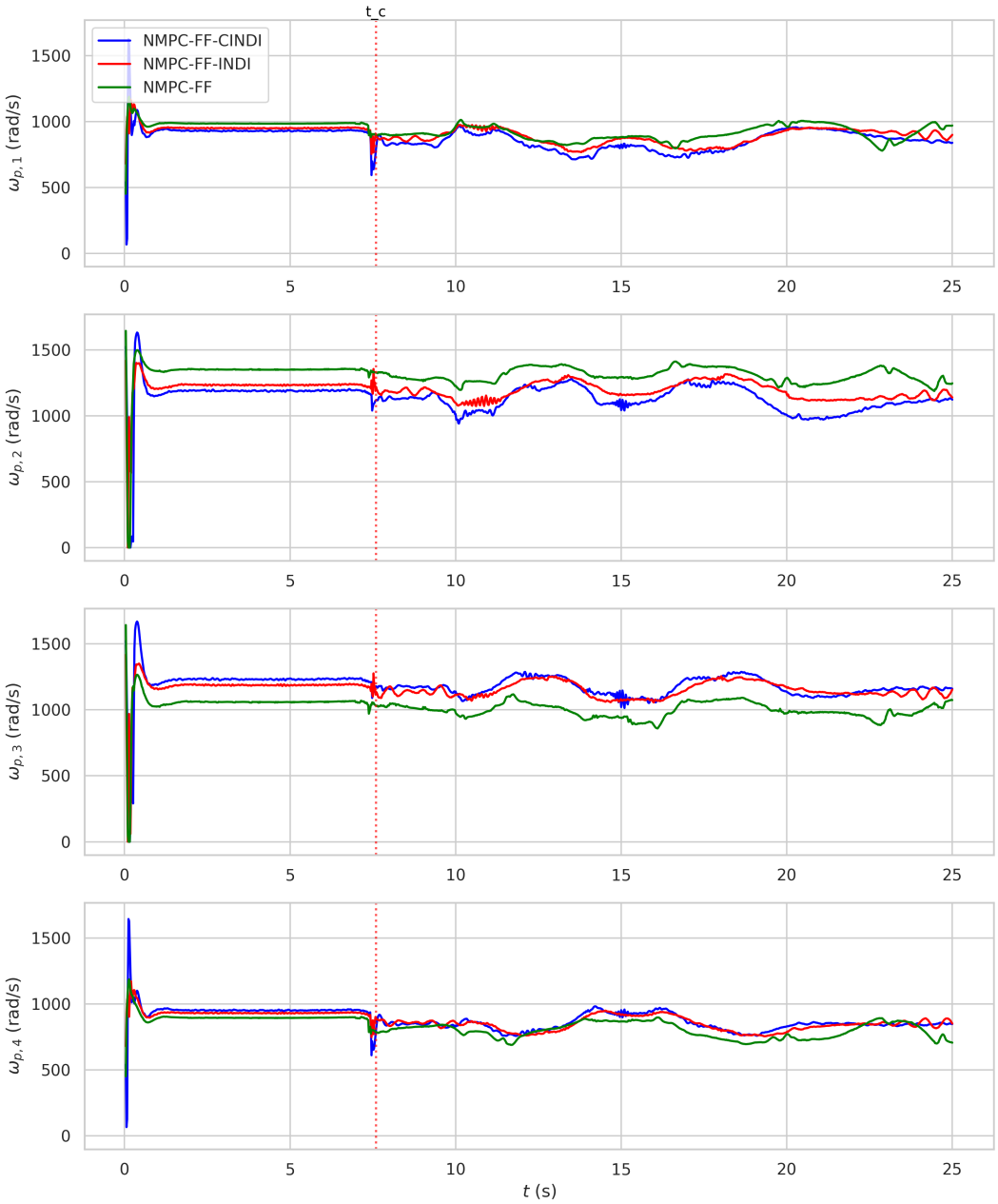


Figure 5-24: The rotor speed commands during the vertical contact experiment.

Inclined Surface

Lastly we will perform a sliding task on a inclined surface. The end-effector translational trajectory is parametrized as follows:

$$
\begin{aligned}
e_x(t) &= \begin{cases} 0 & \text{if } t < t_{\text{write}} \\ 0.3\sin(0.3\pi(t - t_{\text{write}}))\cos(0.3\pi(t - t_{\text{write}})) & \text{else} \end{cases} \\
e_y(t) &= \begin{cases} 0.46\sin\left(\frac{\pi}{2}\frac{t}{t_{\text{move}}}\right) & \text{if } t < t_{\text{move}} \\ 0.46 & \text{if } t_{\text{move}} \leq t < t_{\text{write}} \\ 0.46 + \frac{\sqrt{2}}{2} \cdot 0.3\sin(0.3\pi(t - t_{\text{write}})) & \text{if } t_{\text{write}} \leq t \end{cases} \\
e_z(t) &= \begin{cases} -0.46\sin\left(\frac{\pi}{2}\frac{t}{t_{\text{move}}}\right) & \text{if } t < t_{\text{move}} \\ -0.46 & \text{if } t_{\text{move}} \leq t < t_{\text{write}} \\ -0.46 + \frac{\sqrt{2}}{2} \cdot 0.3\sin(0.3\pi(t - t_{\text{write}})) & \text{if } t_{\text{write}} \leq t \end{cases}
\end{aligned}
\tag{5-9}
$$

where f_r is the loop frequency with $f_r = 0.1$ Hz. This leads to a trajectory with sliding accelerations up to 0.25 ms^{-2} and velocities up to 0.27 ms^{-1}. The timing parameters are chosen to be $t_{\text{move}} = 3$ s and $t_{\text{write}} = 5$ s. The end-effector orientation trajectory is parametrized as:

$$
\begin{aligned}
n_x(t) &= 0 \\
n_y(t) &= \frac{\sqrt{2}}{2} \\
n_z(t) &= -\frac{\sqrt{2}}{2}
\end{aligned}
\tag{5-10}
$$

The wall is positioned at $y = 0.46$ and $z = -0.46$ with the normal vector of the surface being $\hat{\boldsymbol{e}}_n = [0 \ -\frac{\sqrt{2}}{2} \ \frac{\sqrt{2}}{2}]^T$. The end-effector force being tracked is 1 N. Also the heading and joint angle η_1 is defined beforehand:

$$
\begin{aligned}
\psi(t) &= \frac{\pi}{2} \\
\eta_1(t) &= \frac{\pi}{4}
\end{aligned}
\tag{5-11}
$$

In Figure 5-25 a visual representation of the reference trajectory is shown. In Figure 5-26 end-effector tracking errors are shown. In Figure 5-27 the absolute position and attitude tracking error of the end-effector are plotted over time. In Figure 5-28 contact normal and friction forces are plotted.

In this experiment the end-effector attitude tracking performance is better using the INDI based control schemes compared to the baseline NMPC-FF. The end-effector position errors seem similar for all three controllers. The contact forces applied using the INDI control schemes have less peaks and remain relatively consistent compared to the baseline NMPC-FF. Again we see the cascaded INDI controller applies more force than the other controllers. NMPC-FF-CINDI applies an average contact force of 1.5 N compared to NMPC-FF-CINDI which tracks around 1.2 N.

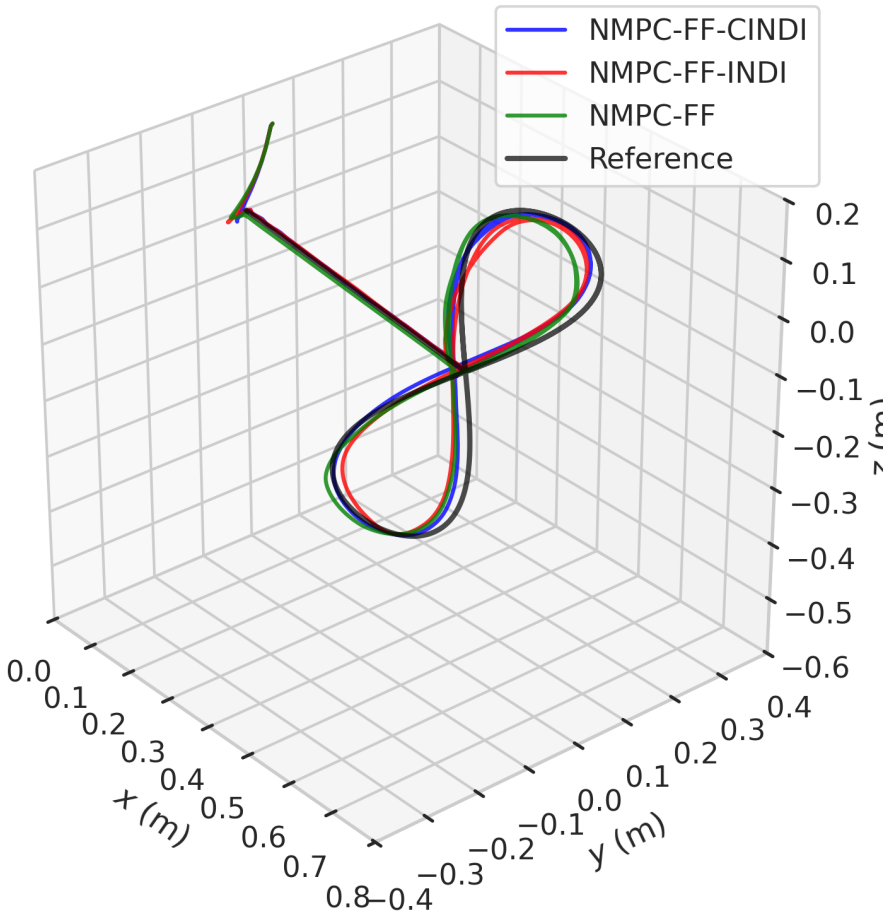


Figure 5-25: 3D representation of the end-effector position trajectory for the inclined contact experiment with nominal model.

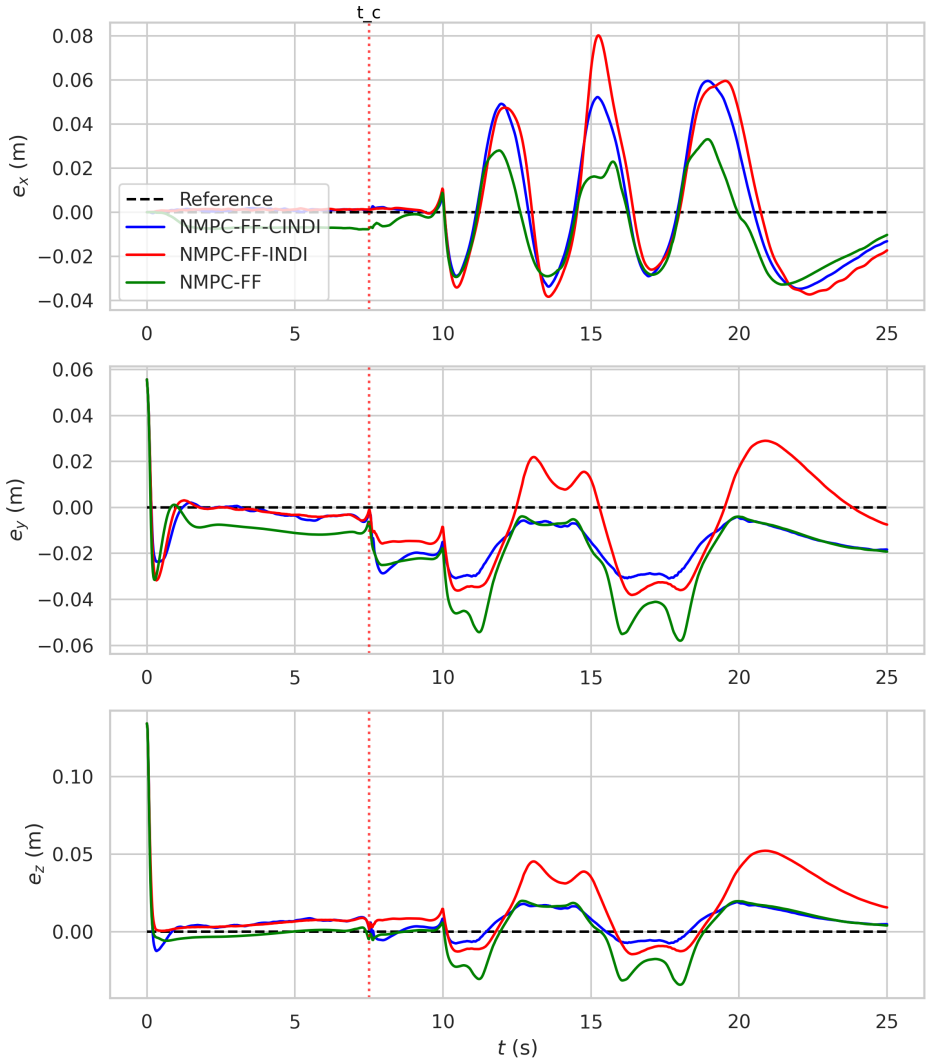


Figure 5-26: The end-effector position tracking error for the inclined contact experiment with nominal model.

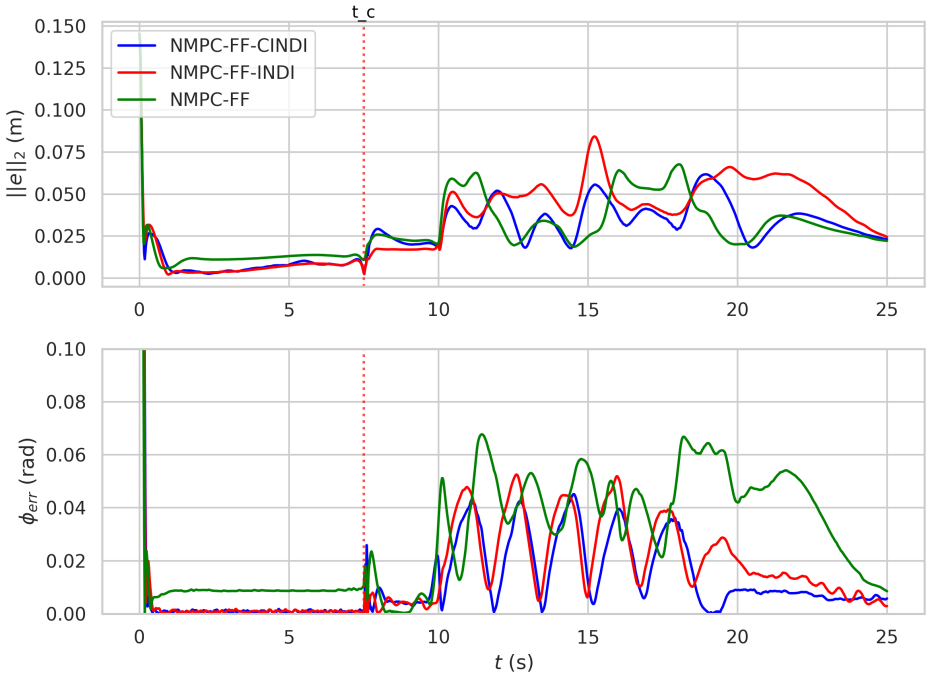


Figure 5-27: The absolute position and attitude error for the end-effector in the inclined contact experiment with nominal model.

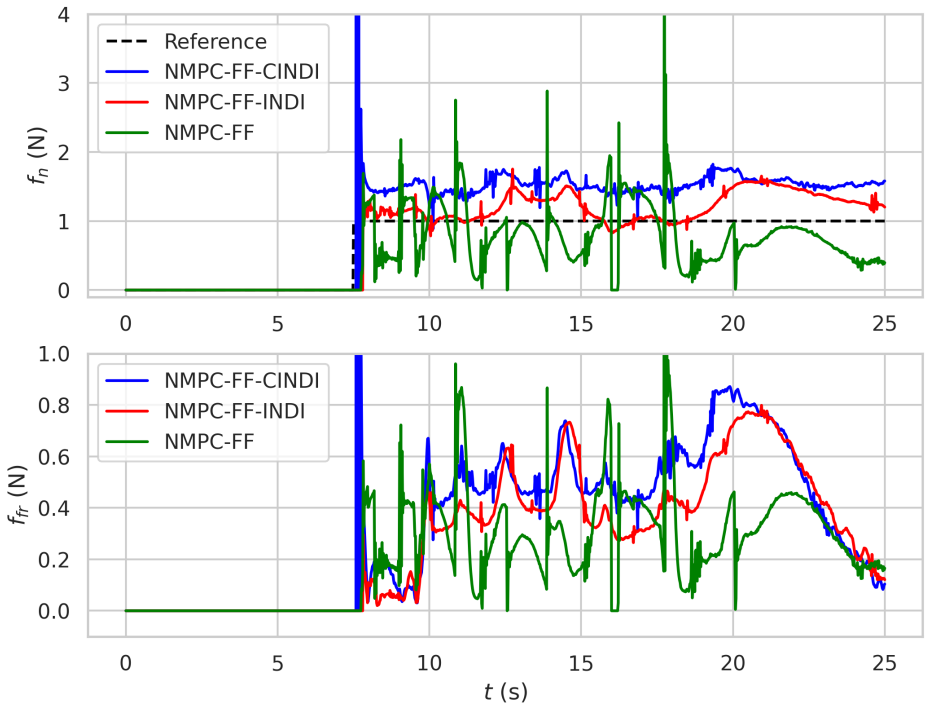


Figure 5-28: The actual contact forces (normal and friction) for the inclined contact experiment with nominal model.

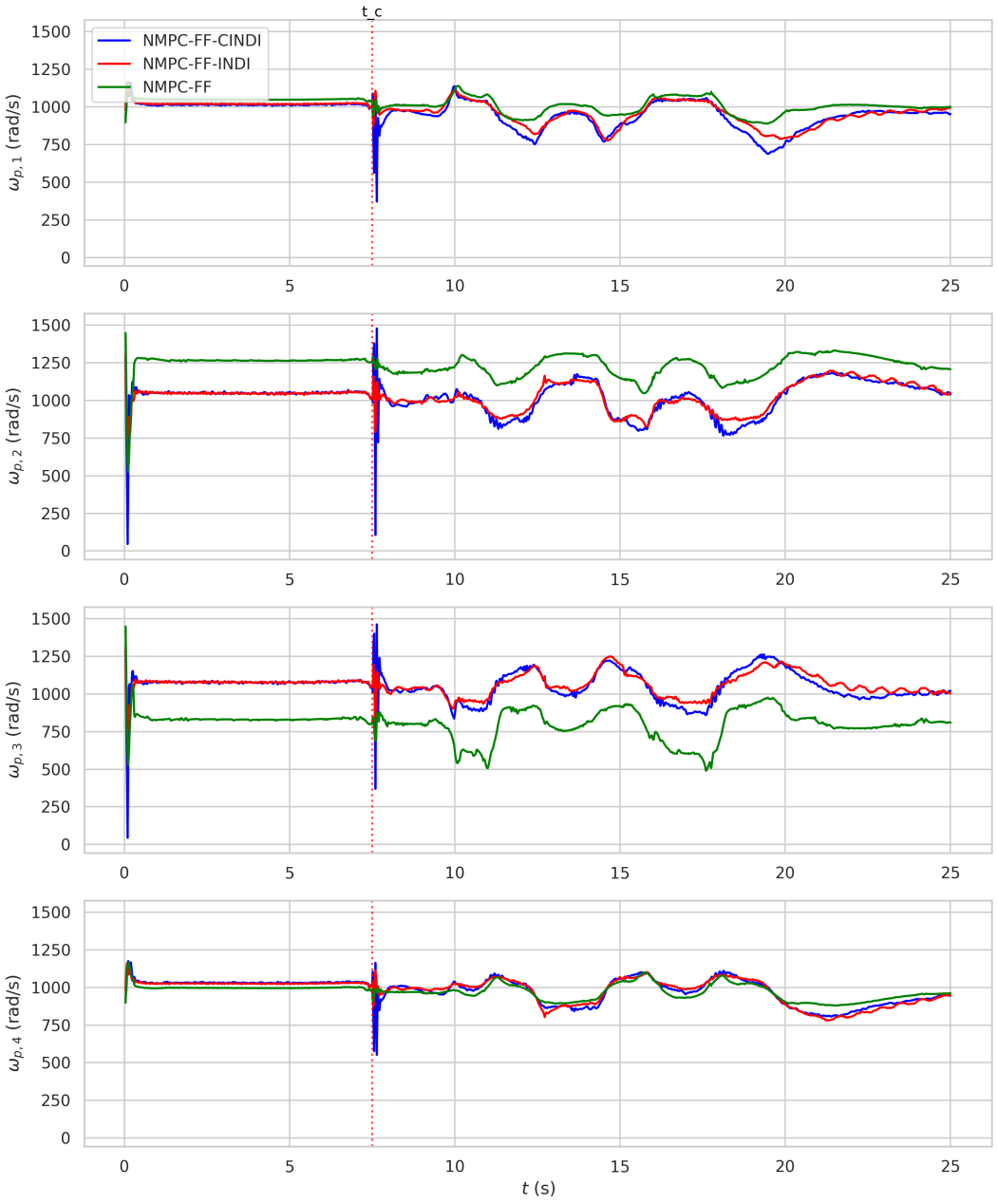


Figure 5-29: The rotor speed commands for the inclined contact experiment.

5-1-4 Ablation Study

In this part we will explore different designs that were also tested in this thesis for the NMPC contact formulation. Also whole-body and decoupled INDI control will be compared in simulation.

NMPC Contact Formulation

We will first study the effect of removing parts of the proposed contact-aware NMPC controller. This includes a feedforward force in the NMPC dynamics when contact is expected and a force tracking term consisting of a external force estimate and a linear spring model. We will first ablate this force tracking term in the cost. The NMPC formulation without the force tracking term in the cost will be referred to as NMPC-noFF.

In Figure 5-30 the position trajectory and in Figure 5-31 the joint trajectory of our proposed NMPC-FF-CINDI and the NMPC-noFF-CINDI is shown for a vertical contact experiment, in which we only wish to achieve force tracking on a point contact experiment. The controller where the force tracking term is ablated becomes unstable after contact. The rotor speeds plot in Figure 5-34 show the NMPC-noFF-CINDI controller quickly diverges, after which the controller becomes unstable and has to reposition itself back it its original position. It managed not to crash but it is definitely undesired behavior.

This instability is expected from any control scheme with the INDI position control layer without feedback on the contact force. The INDI position control increments the thrust based on the NMPC optimal state trajectory. As the aerial manipulator makes contact with a rigid surface, the z-position can never reach the desired position (it can not penetrate the ground). The NMPC however will keep planning its optimal state trajectory under the ground. For this reason the INDI position control keeps incrementing the thrust vector until either the rotors saturate or the system becomes unstable. To counteract this behavior some feedback of the contact force should be included in the contact prediction model, so that the optimal state sequence does not get planned behind a fixed surface when sufficient force is already being applied. For that reason we added a force tracking term in the cost, which includes feedback on the estimate contact force.

Interestingly enough this instability does not happen actually when ablating the force feedback for horizontal contact. In horizontal contact the position offset is in one of the underactuated directions. The NMPC will generate a trajectory which moves inside the wall to minimize end-effector position tracking error. The INDI position controller will increment the thrust toward that direction. As a result the PD attitude controller will tilt the base more towards the wall to apply more force in the direction of offset. At the same time the NMPC is unaware of any contact, so the planned tilt of the base from the NMPC is less tilted than the desired orientation from the INDI position controller. We are taking a linear combination of the NMPC angular rate and the INDI position desired attitude to calculate the angular acceleration command in the INDI attitude controller and this will reach a steady state in this direction. The norm of the thrust vector will also reach a steady state as the thrust increments lead to zero change in the norm of the thrust vector in the INDI position control, only a change in the thrust direction.

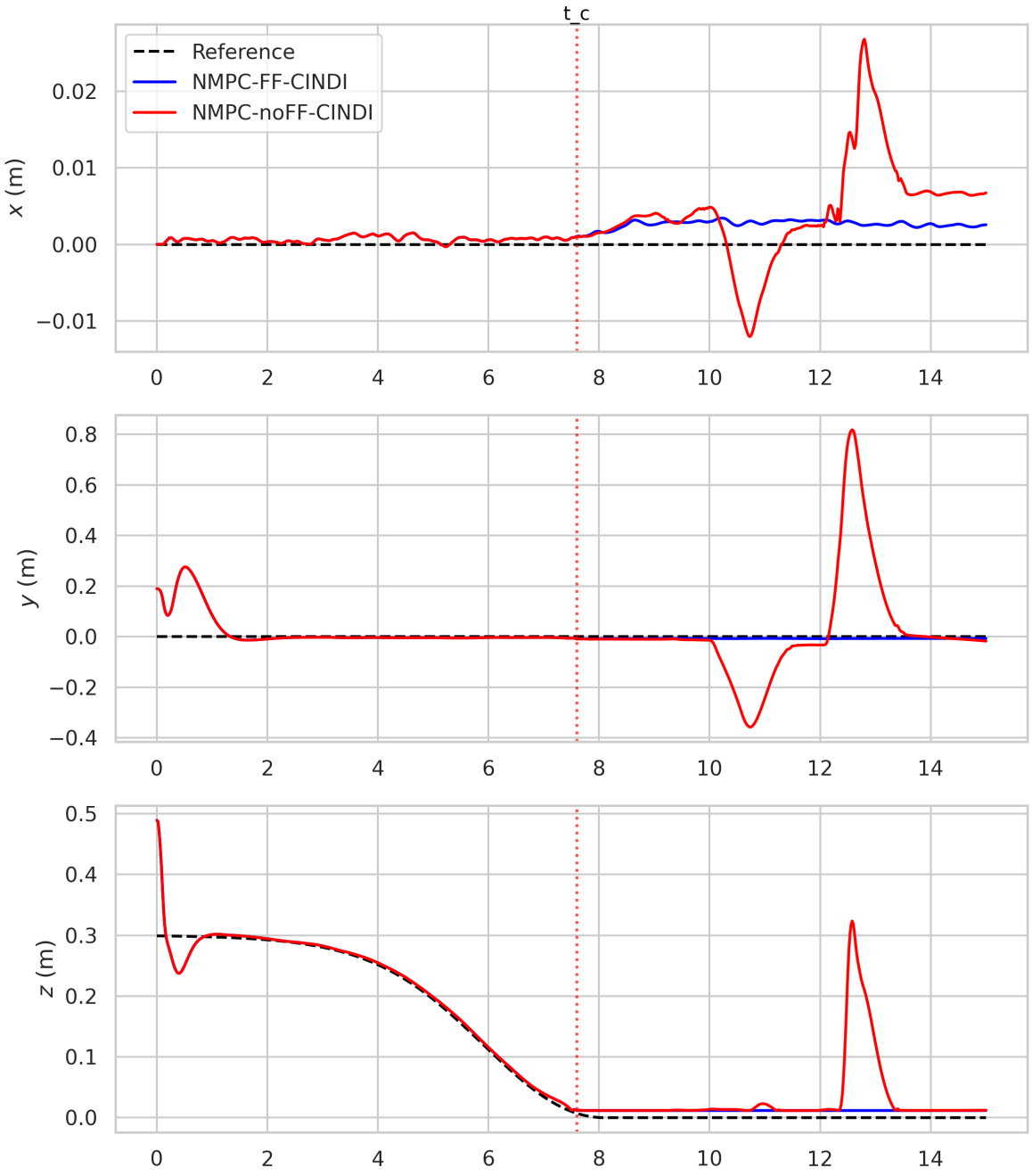


Figure 5-30: The end-effector position trajectory for the ablation study on the force tracking cost term for the vertical contact experiment.

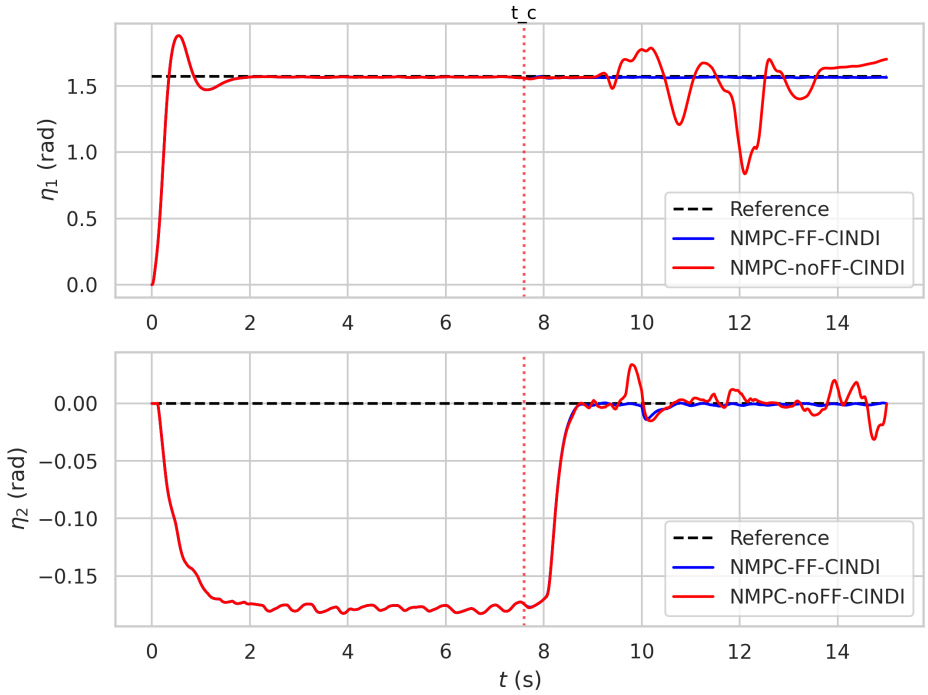


Figure 5-31: The joint position trajectory for the vertical contact ablation experiment.

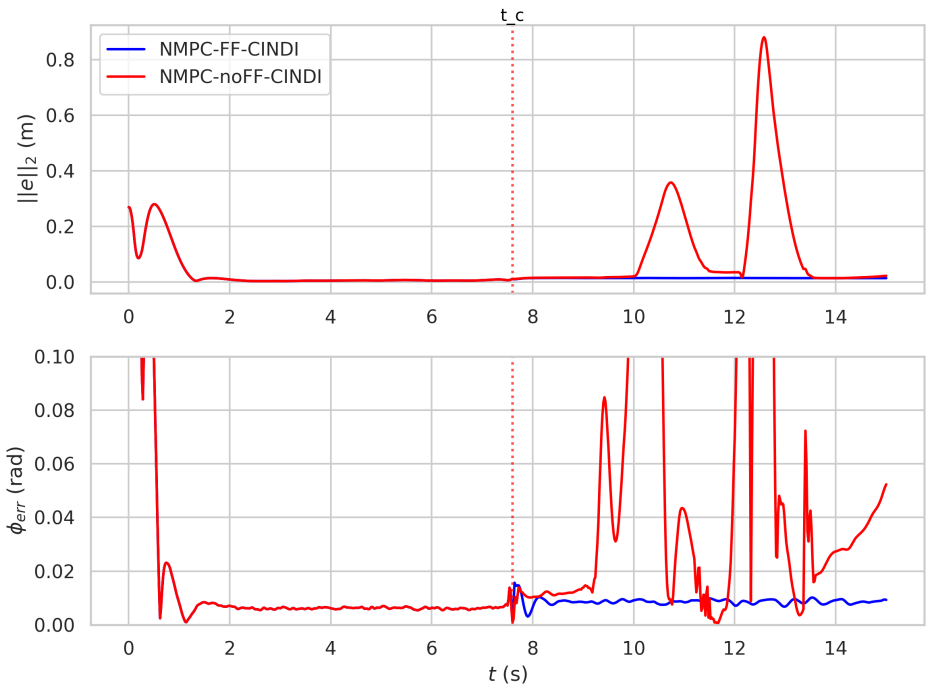


Figure 5-32: The absolute position and attitude error for the end-effector in the vertical contact ablation experiment.

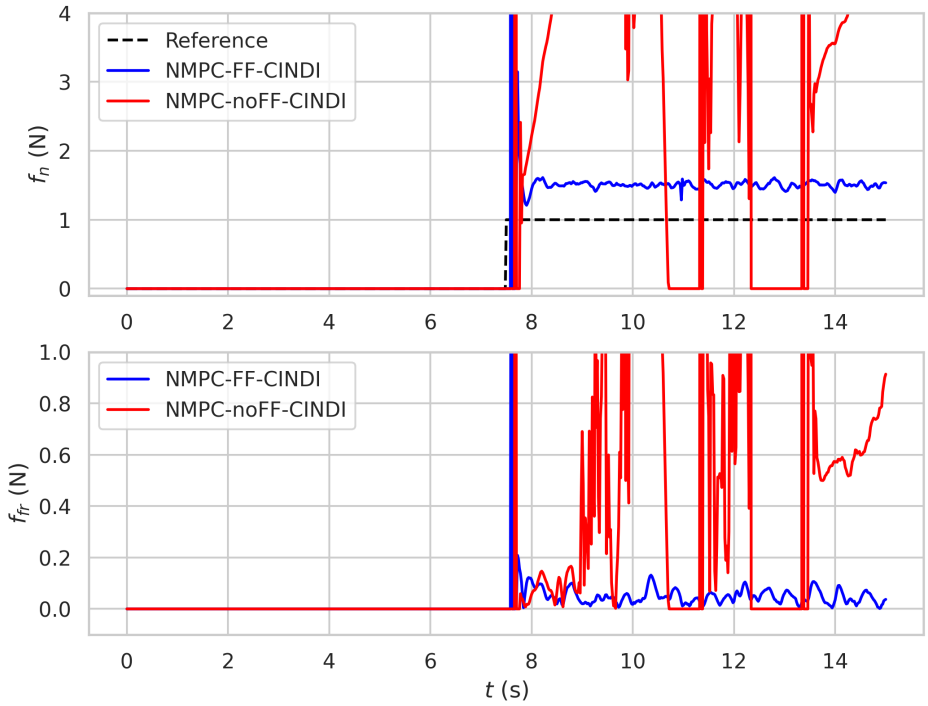


Figure 5-33: The actual contact forces (normal and friction) for the vertical contact ablation experiment.

Next both the added contact force model in the cost and dynamics are studied for force tracking performance on a vertical surface. Without the force term in the NMPC dynamics it is difficult to track an arbitrary reference force for one fixed tuning on the force tracking weight Q_f and stiffness in the linear spring model $k_{p,\text{contact}}$. In Figure 5-35 three different NMPC formulations with a cascaded INDI inner loop are compared for different reference forces and penetration depths for a simple point contact experiment. The NMPC-CINDI scheme is contact-unaware, so it has no force term in the both the cost and dynamics. As can be seen choosing a small penetration depth leads to almost no applied force which is expected. Each contact force will probably have one specific penetration depth that would allow the controller to track the reference force accurately in this experiment, however this would be dependent on the end-effector position weights, base position weights and other parameters. NMPC-noFF-CINDI only has the reference contact force included in the dynamics and no force cost term. The most important observation is that the controllers with the force term in the dynamics formulation track the reference force accurately with low penetration depth. This makes it easier to use these controllers when accurate force tracking is desired without the need of finding a mapping between penetration depths and applied contact force. This mapping would also need to be rediscovered every time when other controller gains are changed.

Whole-body Control

Next the whole-body terms in the INDI inner loops are studied and ablated in an experiment. These coupling terms were introduced to derive an INDI controller for the multibody system

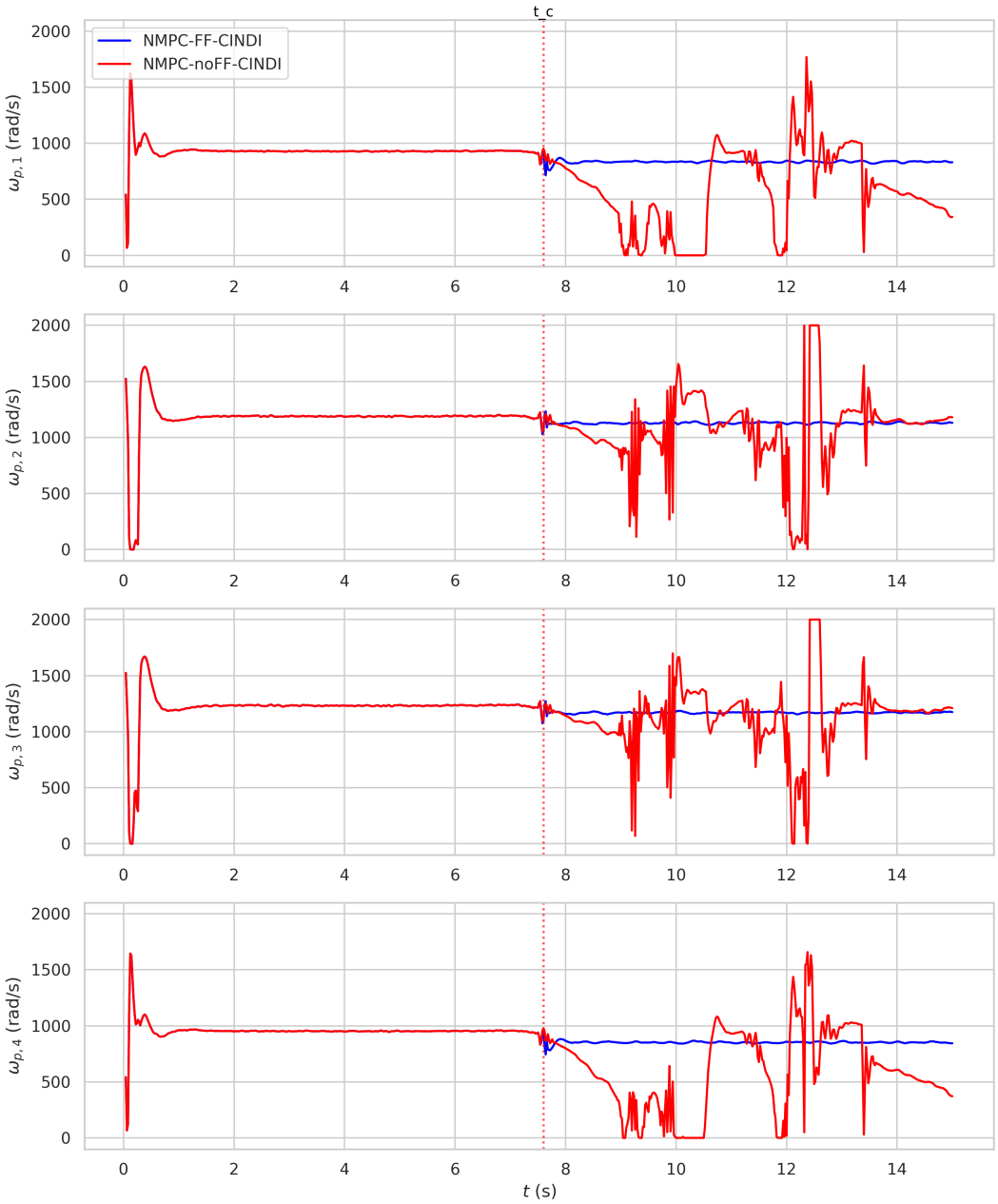


Figure 5-34: The rotor speeds commands in the vertical contact ablation experiment.

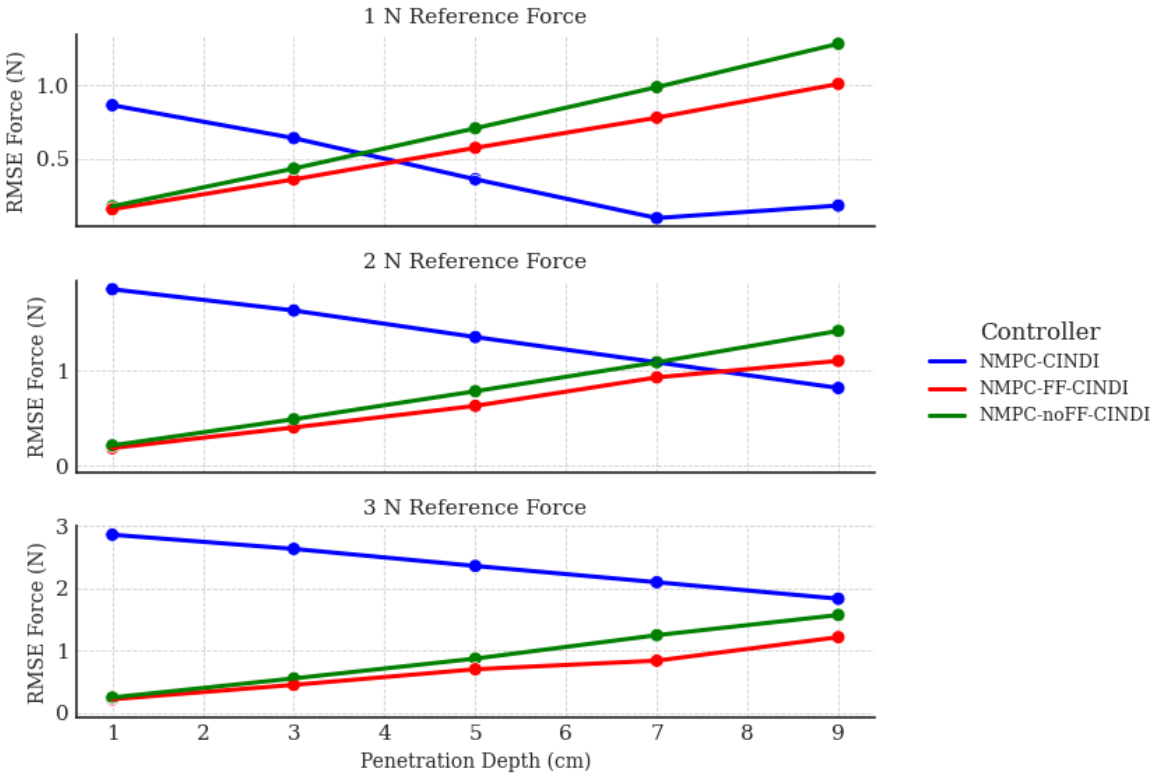


Figure 5-35: Relation between penetration depth and force tracking for different contact formulations in the NMPC and different reference forces. The RMSE is calculated after initial contact transients.

without additional assumptions on the aerial manipulator design. In this experiment we want to see if adding these coupling terms has noticeable effects on the DSAM. Specifically we are interested in an experiments where the coupling terms become significant in the dynamics. The results of this experiment is less relevant to our final demonstration of aerial writing in real world experiments, as the physical DSAM will have a very light arm and does not need to move its arm in sliding experiments.

Whole-body control and decoupled control are key design choices in aerial manipulation literature and our work is a study on applying INDI in aerial manipulators. Therefore it is of interest to study how whole-body and decoupled control compare using INDI. Also it is of interest for future work with the DSAM where the arm may need to carry payload with significant masses. In [47] a decoupled INDI controller is used on the base to reject disturbances due to dynamic arm movements. It is of interest however, to study if INDI might still be perform better when coupling terms of the arm in the base INDI controller are added.

To study the ablation of the coupling terms in the INDI controllers we perform the eight flight trajectory again with nominal model. We keep all the gains on the NMPC-FF-CINDI controller from before and keep the arm mass at 0.2 kg, but we also show the results of the NMPC-FF-CINDI control scheme without coupling terms, which we will refer to as NMPC-FF-CINDI-decoupled. These include all the decoupled control laws derived in Chapter 4 instead of the proposed whole-body variants. The absolute end-effector position and attitude tracking error is shown in Figure 5-36. At the start of the experiment some initial transients to reach steady hover conditions are needed, however the decoupled controller is unable to recover from these transients. The behavior of the decoupled controller is very oscillatory in simulation and this is also visible in the errors plots. If we look at the motor speeds commands in Figure 5-37, it can be seen that the rotor commands are very noisy and also reach the saturation limit during the entire experiment. An arm mass of 0.2 kg is significant for an aerial manipulator of with total mass of 1 kg. From these results we can see that a decoupled approach in the cascaded INDI inner loop controller deteriorates control performance significantly in simulation.

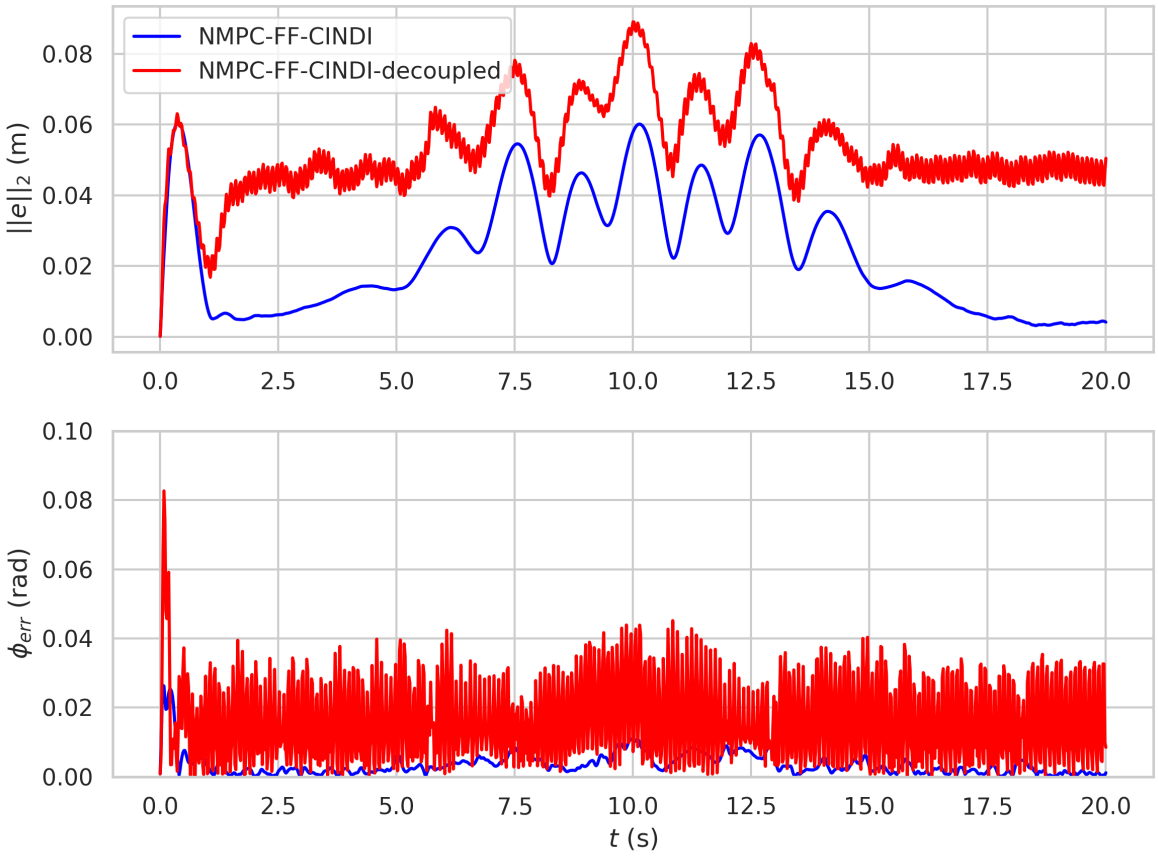


Figure 5-36: The absolute end-effector pose tracking errors in the whole-body control ablation experiment.

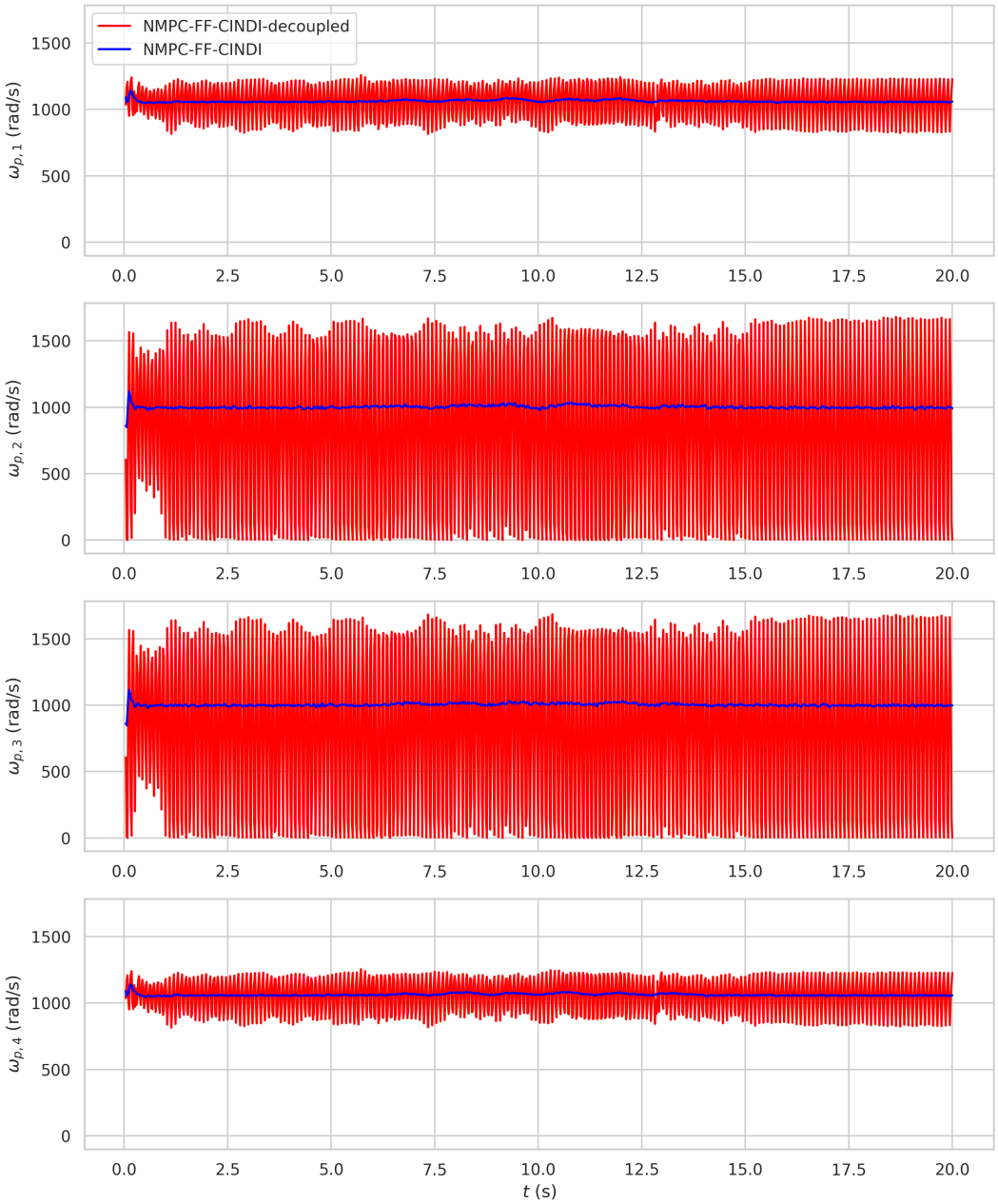


Figure 5-37: The rotor speeds commands in the whole-body control ablation experiment.

Table 5-5: Identified parameters of the physical DSAM.

Parameter	Value	Unit
Quadrotor CoG to root (${}_B\boldsymbol{r}_{br}$)	$[0.045, 0, -0.01]^T$	m
Root to arm CoG (${}_A\boldsymbol{r}_{ra}$)	$[0.12, 0, 0]^T$	m
Root to end-effector (${}_A\boldsymbol{r}_{re}$)	$[0.205, 0, 0]^T$	m
Quadrotor Mass (m_B)	0.787	kg
Quadrotor Moment of Inertia ($\mathcal{I}_b$)	$\text{diag}(0.00254, 0.00220, 0.00515)$	kg·m^2
Arm Moment of Inertia ($\mathcal{I}_a$)	$\text{diag}(0.0001, 0.0001, 0.0001)$	kg·m^2
Arm Mass (m_A)	0.02	kg
Rotor tilt angles ($\boldsymbol{\theta}$)	$[-15, 15, 15, -15]^T$	degrees

5-2 Real World Experiments

5-2-1 Implementation Details

The experiments done in this work are performed at the Mobile Robotics Lab (MRL) at the TU Delft. The MRL has an indoor arena equipped with a VICON motion capture system and has a size of 5 m ×5 m ×5 m. This motion capture system is able to estimate the orientation and position of the DSAM base. Furthermore the DSAM has an IMU located close to the base CoG. The full state estimates of the system are then obtained using an Extended Kalman Filter (EKF) combining the Vicon measurements and the IMU measurements. The arm is controlled using Dynamixel motors, and the joint angles are available as feedback. Accelerometer bias is estimated in flight and subtracted before being used in the INDI control laws and the external force estimates. The flight stack used on the DSAM is based on Agilicious [22] and runs on a Raspberry Pi 5. Communication between the pilot interface, motion capture and the DSAM is done using ROS. In physical experiments the DSAM parameters are found different from the SimScape simulations. These parameters are given in Table 5-5. Also the NMPC is tuned differently. In physical experiments it seemed to be important to track the base position for stability of the flight. The NMPC weights are similar to [55] and are listed in Table 5-6. In [55] the gains were chosen for quadrotor control. In simulation these gains did not show good performance, while the gains given in 5-4 could not remain stable in real world experiments. The reason for this mismatch is unclear, but it could be due to implementation differences in the control schemes on the physical system compared to the simulation or the sim-to-real gap. Furthermore note that in aerial writing task the arm was modified to have a pen attached, which is lighter than the original arm design. It was also attempted to run the simulation with this lower arm mass but this lead to unstable behavior in the SimScape simulation. Finally in Table 5-7 the PD attitude and rate control parameters of the real world experiments are shown. One important note for the the PD attitude and rate control is that in the real world experiment the feedforward angular acceleration from the NMPC ${}_B\dot{\boldsymbol{\omega}}_{b,\text{mpc}}$ is not used. Instead the NMPC predicted orientation enters the angular acceleration command via the the angular rate component.

Table 5-6: NMPC weights and parameters in real world experiments

Parameter	Value
End-Effector Position and Attitude Weights	
$Q_{p,E}$	$\text{diag}([0.1, 0.1, 0.1])$
$Q_{q,E}$	$\text{diag}([200, 200, 200])$
Base Position and Velocity Weights	
$Q_{p,B}$	$\text{diag}([200, 200, 500])$
Q_v	$\text{diag}([1, 1, 1])$
Attitude and Body Rate Weights	
$Q_{q,B}$	$\text{diag}([5, 5, 100])$
Q_ω	$\text{diag}([1, 1, 1])$
Joint Position and Velocity Weights	
Q_η	$\text{diag}([100, 100])$
$Q_{\dot{\eta}}$	$\text{diag}([1.0, 1.0])$
Contact Force and Control Input Weights	
Q_f	3.0
Q_u	$\text{diag}([0.6, 0.6, 0.6, 0.6])$
Joint Acceleration Weights	
$Q_{\ddot{\eta}}$	$\text{diag}([1.0, 1.0])$
Parameters	
$k_{p,\text{arm}}$	400
$k_{d,\text{arm}}$	30
$k_{p,\text{contact}}$	2

Table 5-7: PD attitude and angular rate control parameters in real world experiments

Parameter	Value	Description
$k_{q,\text{red}}$	120	Attitude gain
K_{rate}	$\text{diag}([25, 25, 20])$	Rate gains (roll, pitch, yaw)
$k_{q,\text{yaw}}$	80	Yaw gain

Table 5-8: Filter values for each of the control layers in real world experiments

Filter	Value	Unit
f_{att}	12	Hz
f_{outer}	6	Hz
f_{allo}	12	Hz

5-2-2 Free Flight Experiments

In this section two real world free flight experiments are done for the DSAM to validate the controller end-effector pose tracking performance without contact. But before performing these tasks a short hover experiment is done to analyze some steady-state behavior of the quadrotor and to point out some estimation errors.

In Figure 5-38 the estimated orientation during hover for the three controllers is shown. It can be seen that there seems to be an orientation offset during hover. This offset is likely due to the calibration of the VICON Motion Capture system, which is used to estimate the orientation. In hover the body z-direction should align with the gravity leading to zero tilt. This orientation error could also lead to some positional offsets which we also see in the following experiments due to the coupling between attitude and position tracking. Furthermore as we will see later for the contact force estimation in the aerial writing section, this orientation offset is also needed for analyzing the contact forces more accurately. The average orientation of the DSAM base is estimated in quaternion parametrization as $q_{\text{offset}} \approx [0.019, -0.003, 0.707, 0.707]$ by taking the average of orientation of the hover data over the time span of 1 second. This orientation includes a 90 degree heading rotation, which is not part of the offset. By converting this offset to the Euler angles parametrization, the error in roll estimate seems to be around -1.8 degrees and the error in pitch estimate around 1.3 degrees. For attitude tracking we will not consider this error as the controller is unaware of this offset.

Next up an eight trajectory is tracked in free flight, similar to the simulations. A top view of the base position during the eight flight experiment is shown in Figure 5-39. The position tracking errors of the end-effector are shown in Figure 5-40 and the absolute end-effector position and attitude tracking error are shown in Figure 5-41. Finally the joint angles are shown in Figure 5-42 and the commanded rotor speeds in Figure 5-43.

It can be observed from the motor speeds that in the physical experiment the baseline NMPC is noisier compare to the controllers with INDI inner loops. At first it was thought that INDI would add more noise in the inputs compared to the outer loop due to noisy accelerometer measurements, however this did not seem to be the case. Model mismatch and external disturbances in reality are hard to mimic in simulation. So while in simulation NMPC mostly shows well behaved input and performs on par in free flight without disturbances compared to the INDI-based schemes, in reality this did not seem the case. Also due to the tilting of the rotors, the airflow of each of the rotors may actually interfere with each other leading to more disturbances and unmodeled dynamics. In the absolute position tracking errors in Figure 5-41 it is visible that the NMPC-FF-CINDI significantly out performs the baseline NMPC and NMPC-FF-INDI controllers, where the NMPC-FF-CINDI control scheme reaches maximum end-effector tracking errors of around 7 cm while the NMPC-FF controller reaches maximum end-effector tracking errors of around 14 cm.

We note in the results that all controllers have position offsets, also when close to hover at the start and end of the trajectory. The bias can be due to many reasons, including hardware imperfections, unmodeled dynamics and estimation errors. Especially the NMPC-FF-INDI has significantly more offset close to hover compared to the other two controllers. This is probably mostly due to the INDI attitude tracking inner loop that is unaware of the constant orientation error. In the cascaded INDI version this is partly compensated in the position INDI controller which adjusts the tilt because of this steady state position offset. The pure attitude INDI

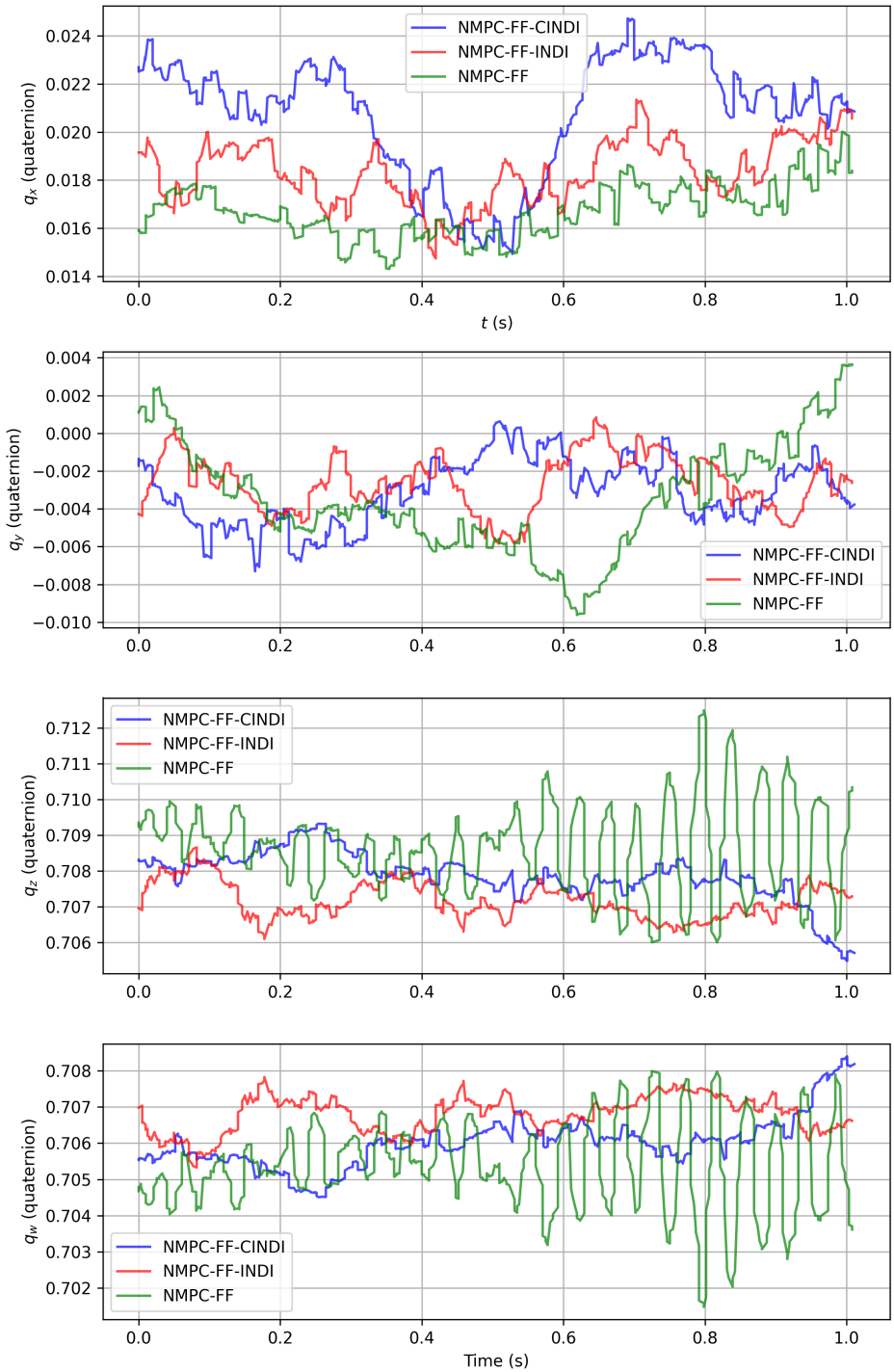


Figure 5-38: Estimated orientation in quaternions of the DSAM base during hover in a real world experiment.

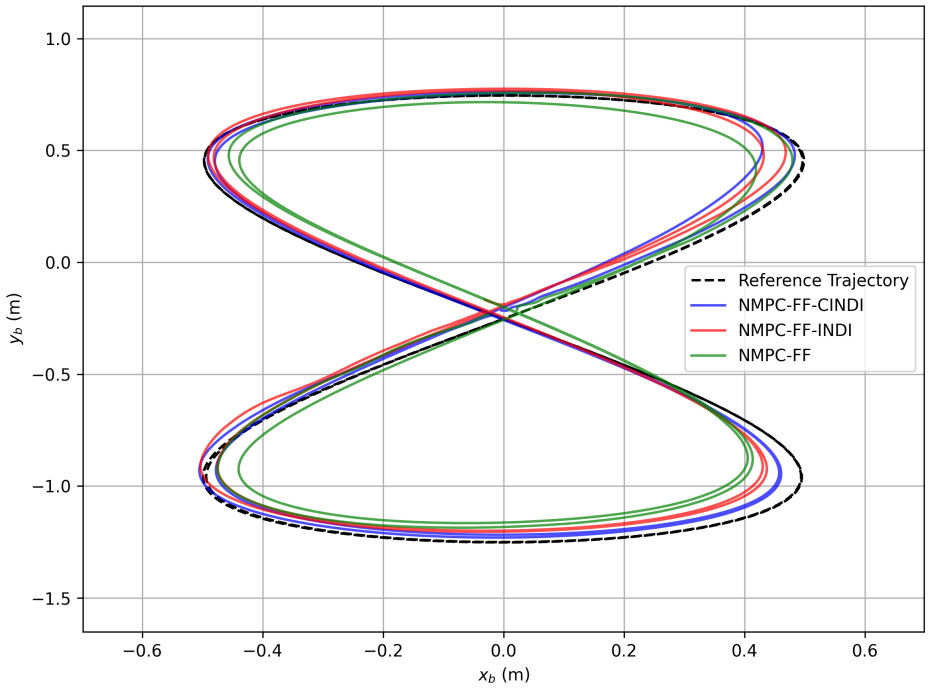


Figure 5-39: Top view of the base position trajectory for the physical eight flight experiment.

controller wants to reject this orientation offset error until the outer loop NMPC outputs zero angular acceleration. This occurs when a balance is found between all the tracking offsets.

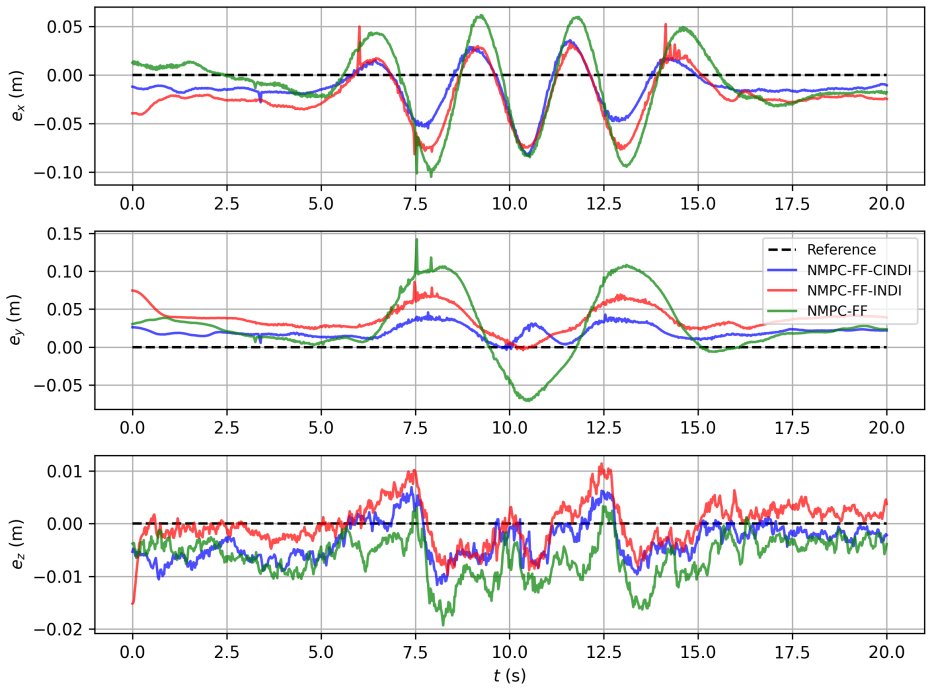


Figure 5-40: The end-effector position trajectory tracking errors in the physical eight flight experiment.

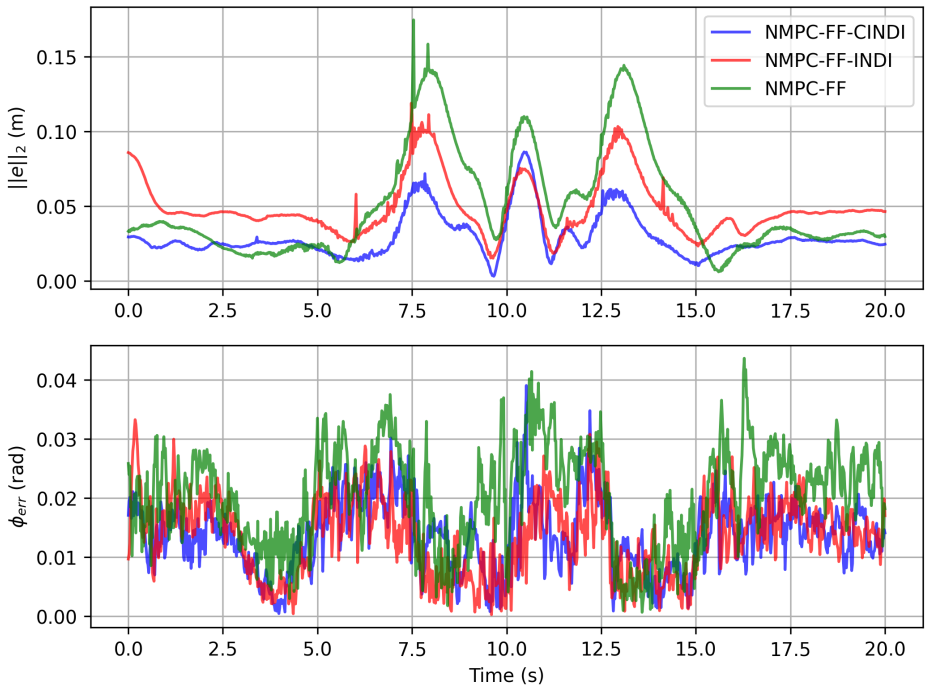


Figure 5-41: The absolute end-effector position and attitude error in the physical eight flight experiment.

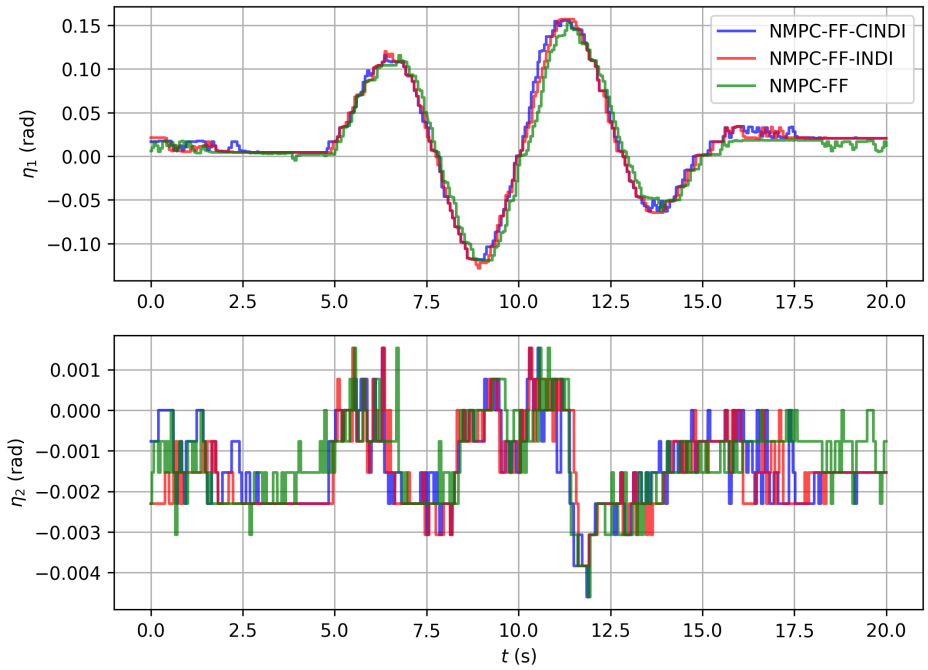


Figure 5-42: The joint positions in the physical eight flight experiment.

Next a fixed end-effector tracking experiment is performed, where the aerial manipulator has to move its joint, base position and yaw while tracking a fixed end-effector position. A visual montage of the end-effector is given in Figure 5-44. A top view of the base position during the fixed end-effector experiment is shown in Figure 5-45 and a side view is shown in Figure 5-46. The absolute end-effector position and attitude tracking error are shown in Figure 5-47. Finally the joint angles are shown in Figure 5-48 and the commanded rotor speeds in Figure 5-49.

Again we can see that commanded rotor speeds are more noisy for the pure NMPC approach compared to the INDI-based control methods. The absolute end-effector position error seems to be similar for NMPC-FF and NMPC-FF-CINDI, while NMPC-FF-INDI has more end-effector position tracking error. The error seems quite constant and related to the steady-state position offset described in the previous results. This trajectory moves the arm, but is not very dynamic for the base compared to the eight flight trajectory. It seems the INDI control layers do not add significant benefit when this is the case compared to the baseline NMPC control scheme.

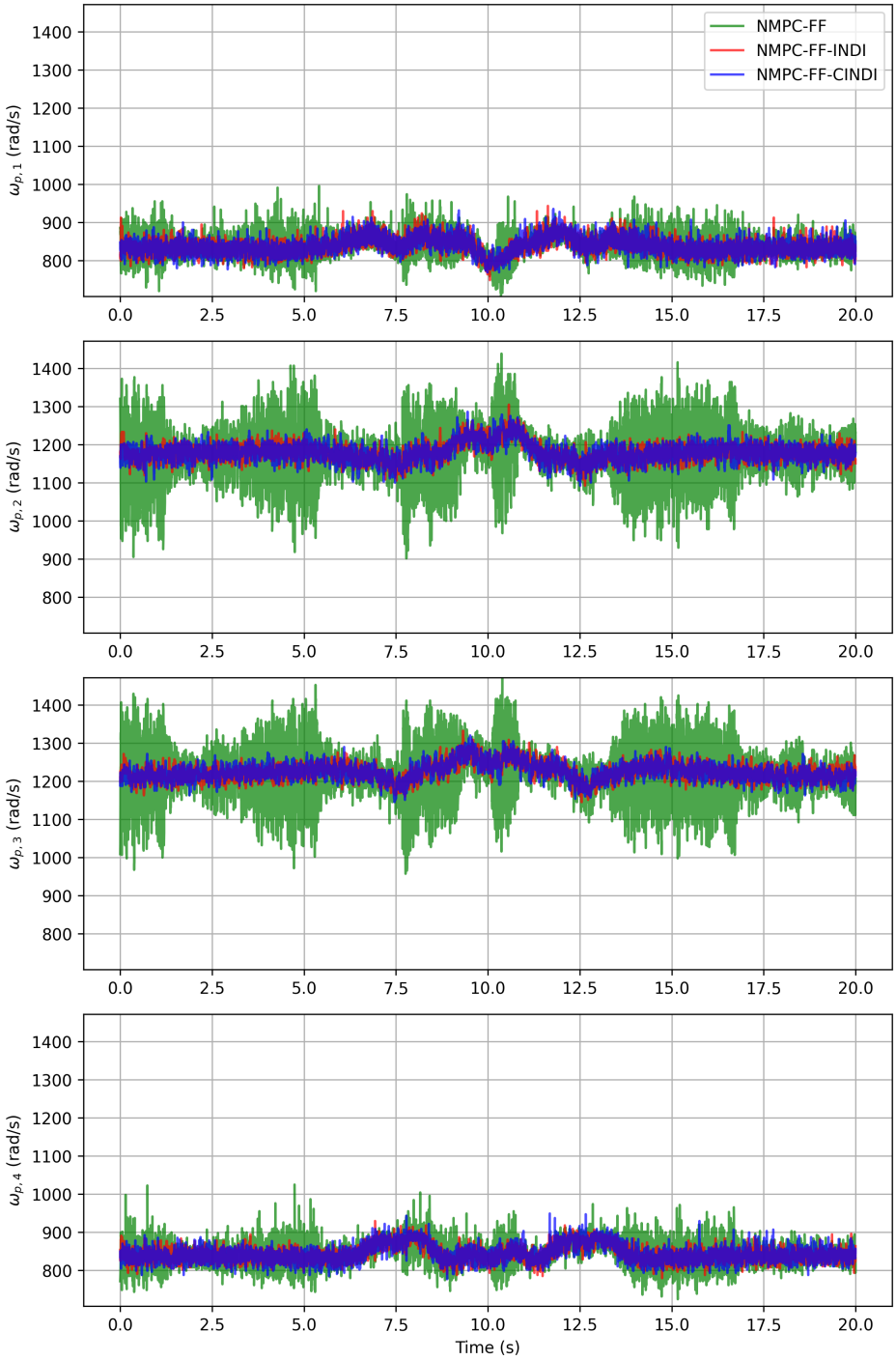


Figure 5-43: The rotor speeds commands for the physical eight flight experiment.

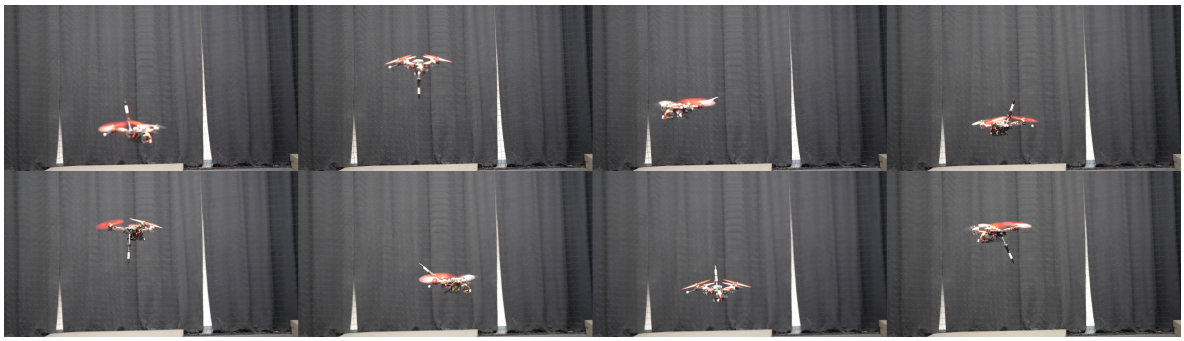


Figure 5-44: A visual montage of the fixed end-effector experiment.

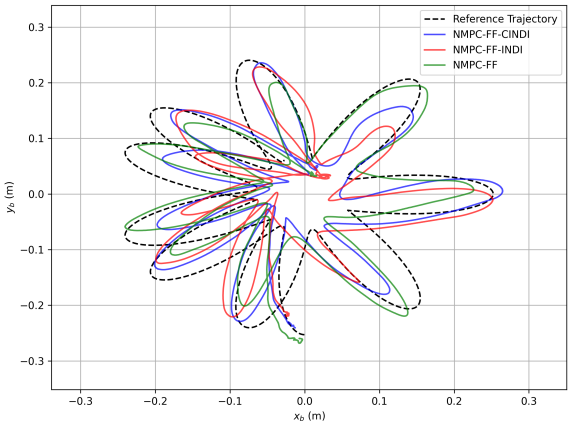


Figure 5-45: Top view of the base position trajectory for the physical fixed end-effector tracking experiment.

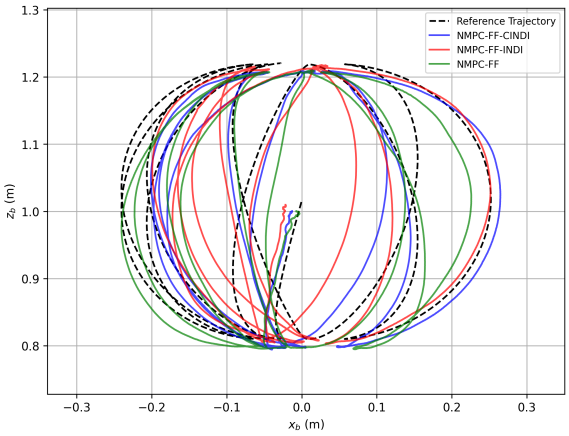


Figure 5-46: Side view of the base position trajectory for the physical fixed end-effector tracking experiment.

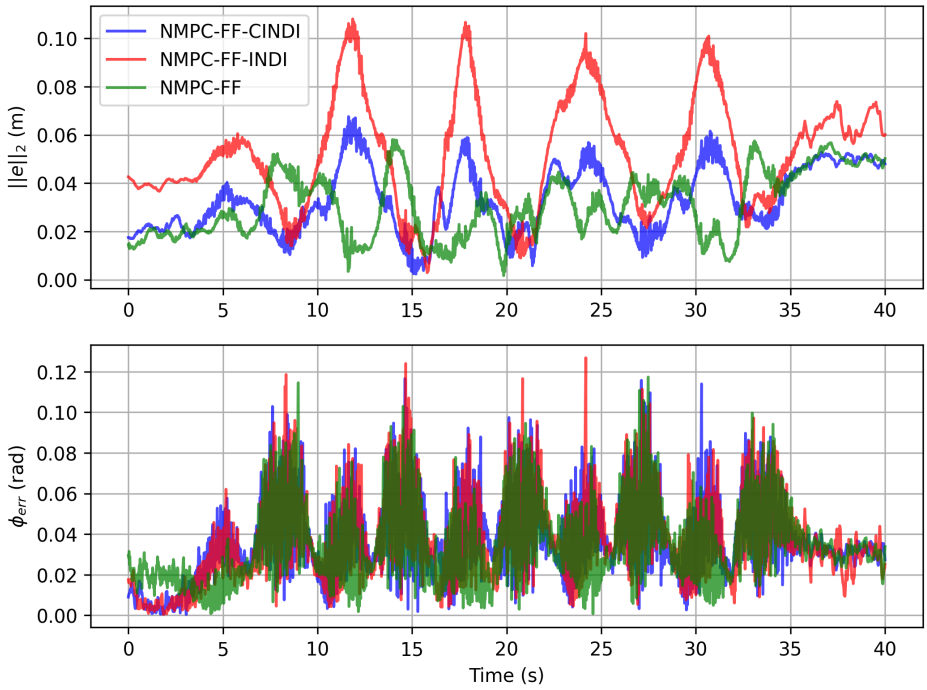


Figure 5-47: The absolute end-effector position and attitude error in the physical fixed end-effector tracking experiment.

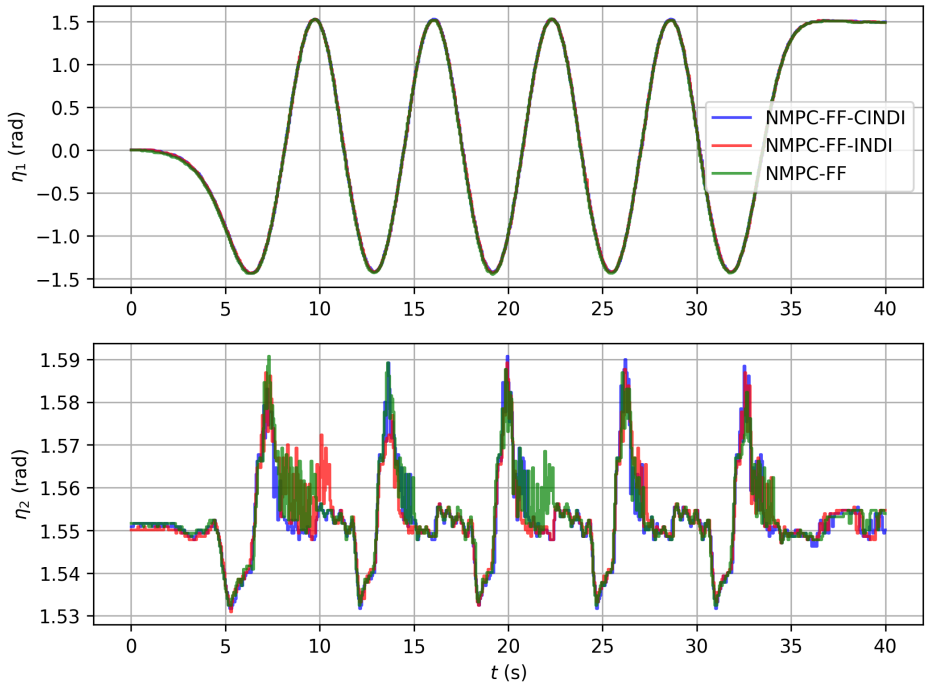


Figure 5-48: The joint positions in the physical fixed end-effector tracking experiment.

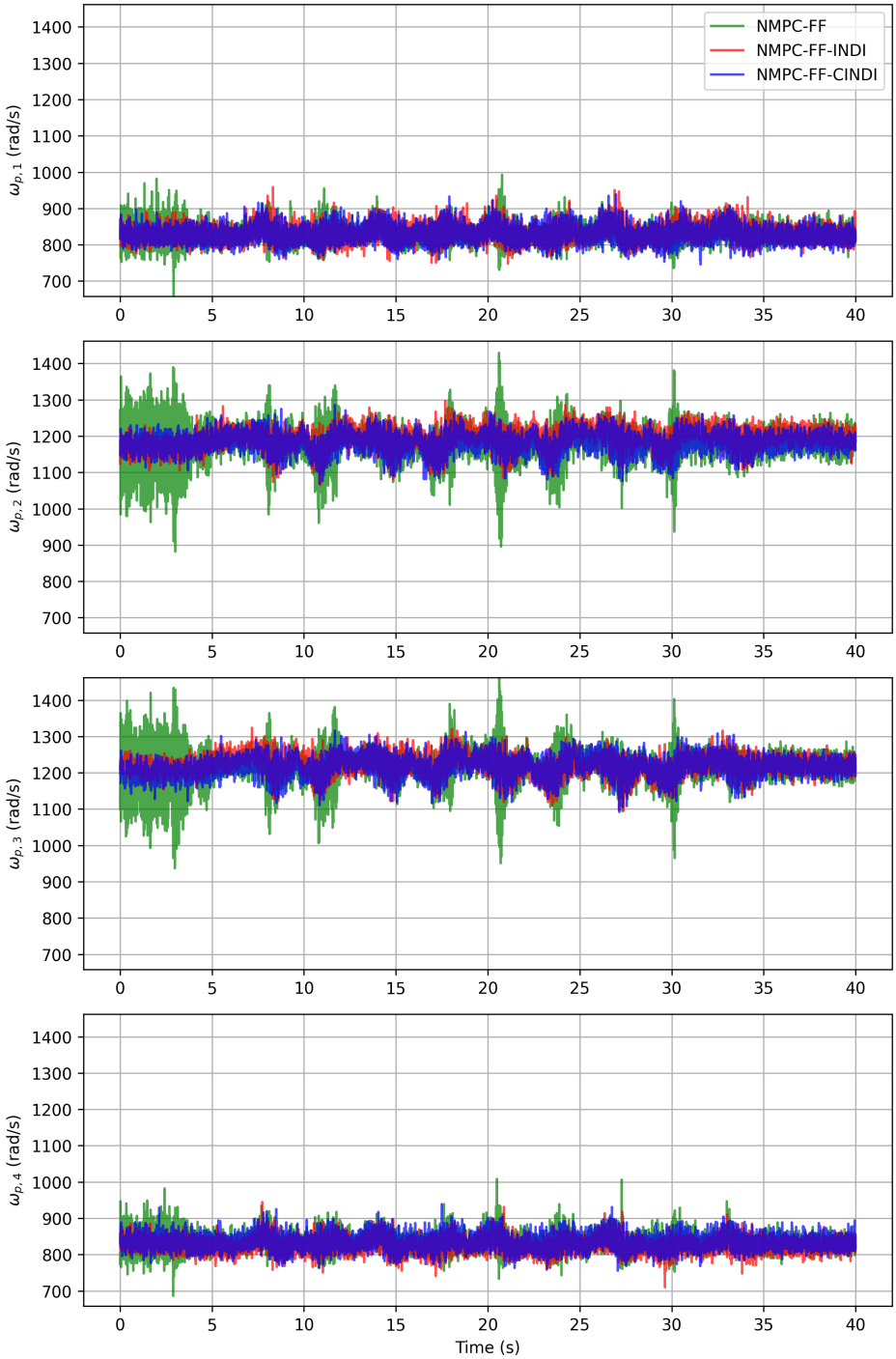


Figure 5-49: The rotor speeds commands for the physical fixed end-effector tracking experiment.

5-2-3 Aerial Writing: Whiteboard

In this section we test the DSAM on APhI sliding tasks. Specifically we use a whiteboard and a marker attached to the DSAM to draw a figure eight while tracking a desired contact normal force, similar to the simulation experiments. We do this horizontally with and without external disturbances. The external disturbances will be due to wind generated by a fan with wind speeds of around 5 ms^{-1}. We will also perform the eight sliding experiment on an 45 degrees inclined surface similar to simulation without external disturbances to show that the DSAM is also able to do the drawing task on non vertical surfaces.

In all the coming experiments the baseline NMPC-FF is not shown. This is because this controller failed to perform the sliding experiment for the same NMPC weights. The controller could remain stable in contact, however we were unable to get sliding to work. As seen in the free flight experiment, the rotor speeds were already very noisy for the pure NMPC-FF controller in the eight-like flight trajectory.

In the following sections, the estimated external forces are decomposed into two components: the estimated force in the direction normal to the contact surface, F_n, and the estimated force in the direction parallel to the surface in the horizontal world plane F_p.

As mentioned, the real system is not equipped with a force/torque sensor, and therefore the contact forces cannot be measured directly as in simulation. Instead, the contact force estimates in the result are derived from the orientation of the drone during the horizontal contact experiments. Specifically, the tilt of the drone with respect to the gravity vector is used to compute the force components in the normal and parallel directions of the whiteboard surface. This approach relies on the fact that, in steady-state contact, the drone must tilt to exert a force on the surface, and the magnitude of this tilt is directly related to the applied contact force. The calculation of the horizontal force is based on the mass of the aerial manipulator. This requires an additional assumption, that the aerial manipulator is not accelerating in the z-direction.

However, basing the force estimate solely on the drone's orientation and mass means that inertial forces due to linear accelerations are included in both F_n and F_p. These forces are removed using filtered accelerometer measurements when processing the results. The calculated force estimates still include other external forces besides contact forces and this should be considered when interpreting the results. This is important for the experiments with wind disturbances.

Another important note is the orientation estimation offset mentioned before. This would directly influence the force estimates and therefore this also needs to be taken into account when calculating the estimated forces. This is done by rotating the measured orientation by the inverse of the estimated orientation offset q_{offset} in postprocessing of the data.

Figure 5-50: A visual montage of the eight drawing experiment on a whiteboard

The same eight drawing experiment as in simulation on a vertical surface is performed in a real world experiment with some initial position offset of the reference trajectory. The end-effector is equipped with a marker and the surface will be a whiteboard. The normal vector of the whiteboard is $\hat{\boldsymbol{e}}_n = [0 \ -1 \ 0]^T$. A visual montage of the experiment is shown in Figure 5-50. A side view of the horizontal eight experiment without disturbances is shown in Figure 5-51. The end-effector position tracking errors are shown in Figure 5-52 and the absolute end-effector position and attitude tracking error are shown in Figure 5-53. The joint positions during the experiment are shown in Figure 5-55. The estimated external forces are shown in Figure 5-54. The commanded rotor speeds are shown in Figure 5-56. Two time instants are highlighted in the figures with a dashed vertical line, which correspond to the time at which contact occurs t_c and the time at which the sliding starts t_s.

First we note that force tracking seems to be quite consistent, and tracked force is around 0.5 N too much for NMPC-FF-INDI and around 0.4 N for NMPC-FF-CINDI. If we compare it to the results in simulation, the tracked force was also too large for NMPC-FF-CINDI but consistent throughout the experiment. The extra contact force may be partly explained by the orientation offset. During hover the DSAM is estimated to have a pitch offset which would also leads to additional contact normal force during this experiment. The cascaded INDI inner loop applies slightly more force than the controllers without the position INDI control layer. This was also observed during simulation. The controllers track a consistent force reference during the experiment which is desired. In Figure 5-51 and in Figure 5-53 we see that cascaded controller is able to obtain visibly less tracking error, leading to end-effector position offsets of at most 5 cm compared to around 7 cm without the INDI position control layer. In Figure 5-52 we see the difference in position tracking error before contact is mostly due to a steady-state offset from the orientation estimate error. After contact the end-effector position offset is increased. This is also visible in Figure 5-51, where the drawing seems to be shifted to the bottom-left.

In Figure 5-55 we see that $\eta_1^r = 0$ during contact. However during contact it should be non-zero, if the end-effector direction should be perpendicular to the wall. The NMPC layer does calculate a desired joint angle to account for this. However due to the weight on the joint position tracking in the NMPC this will lead to end-effector orientations which are not perpendicular to the surface because of the reference joint position. This should be improved in the reference trajectory generation method. This could explain the significant offset in the end-effector z-direction after contact.

During sliding there are friction forces. This may lead to additional tracking error. It can be seen that NMPC-FF-CINDI is less affected by the sliding friction forces than NMPC-FF-INDI. This can be seen in the deviations of the end-effector position errors e_x and e_z during the drawing experiment after t_s.

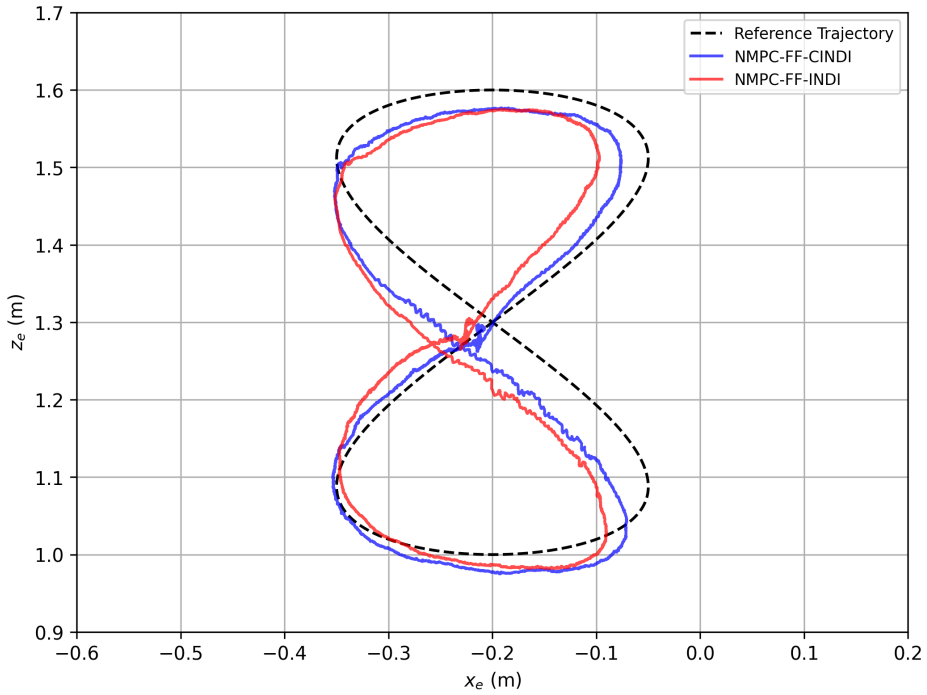


Figure 5-51: Side view of the end-effector position trajectory during the eight drawing experiment on a vertical surface with no wind.

Besides the steady-state end-effector position offset during contact, the obtained results are consistent with the simulation results. A probable reason why this offset did not occur in simulation is because the NMPC weights were mostly put on the end-effector pose tracking and less on the base tracking, while in real world experiments the NMPC weights were largest on the base position tracking instead of the end-effector. Lastly, in the real world experiment the importance of the attitude INDI is really shown as the pure NMPC-FF controller became unstable during the sliding experiment. Friction and normal forces on the aerial manipulator end-effector introduce large torques on the base due to the length of the arm. Ignoring these terms in the pure NMPC-FF control scheme does not seem to work in reality and a more robust controller is needed to reject these disturbances. For our work this is done using an INDI attitude control layer.

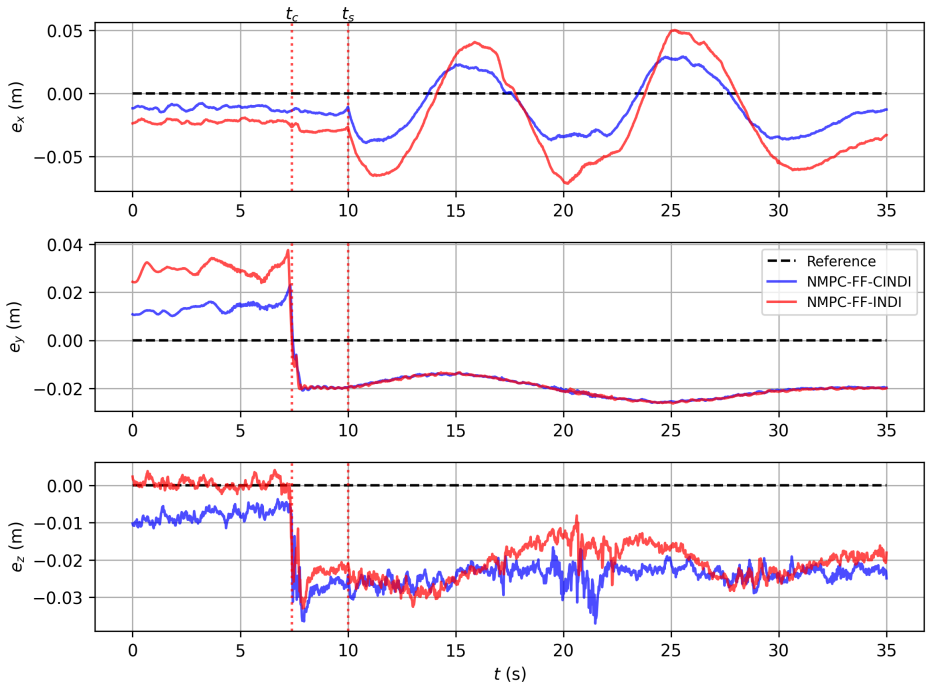


Figure 5-52: The end-effector position trajectory tracking errors in the real world eight drawing experiment on a vertical surface with no wind.

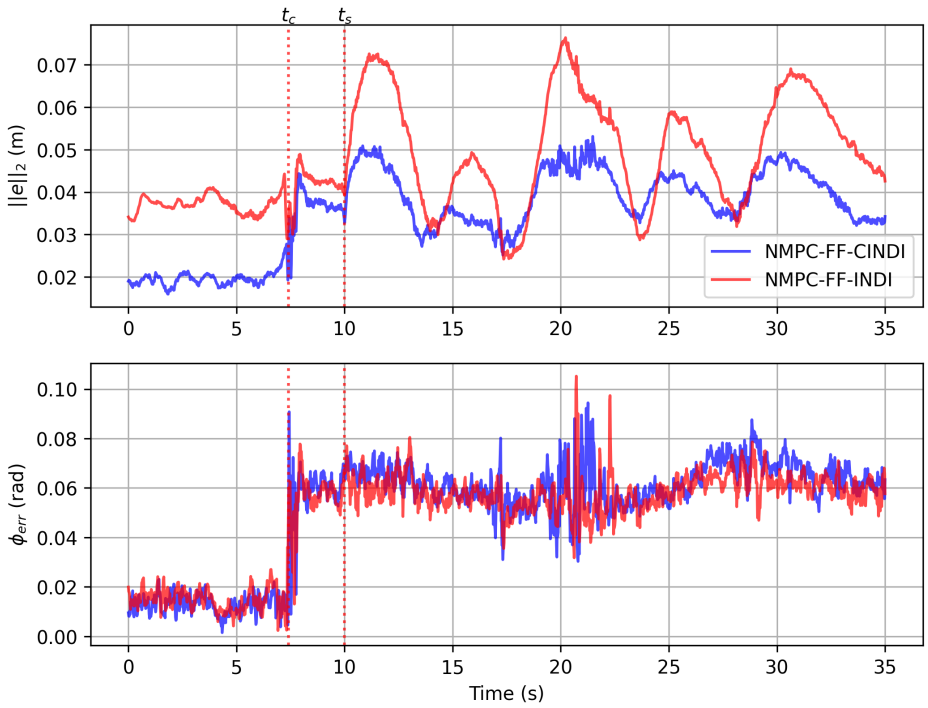


Figure 5-53: The absolute end-effector position and attitude error in the real world eight drawing experiment on a vertical surface with no wind.

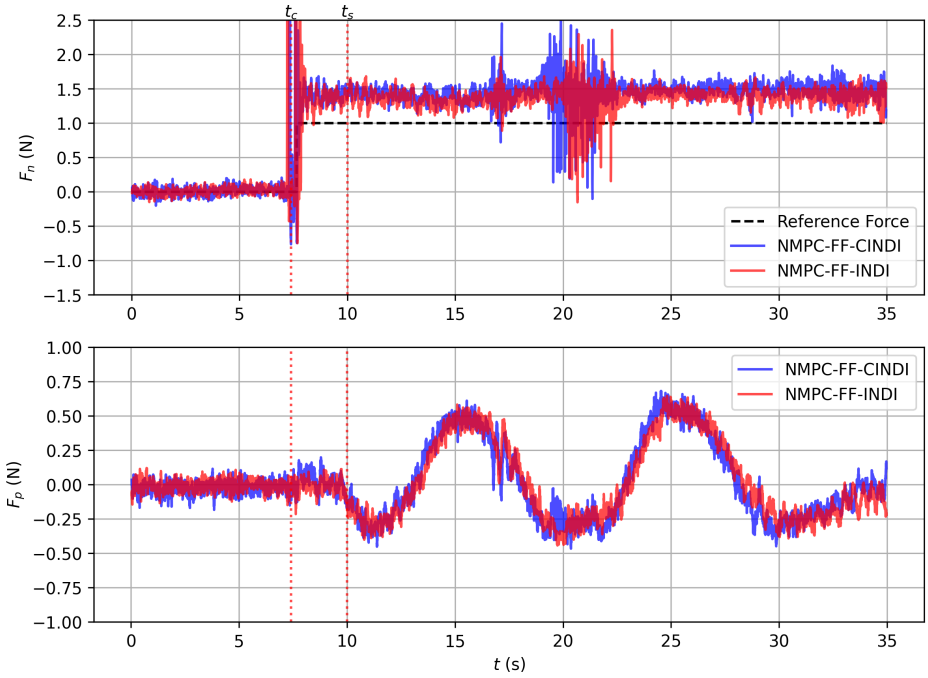


Figure 5-54: The estimated external forces in the real world eight drawing experiment on a vertical surface with no wind.

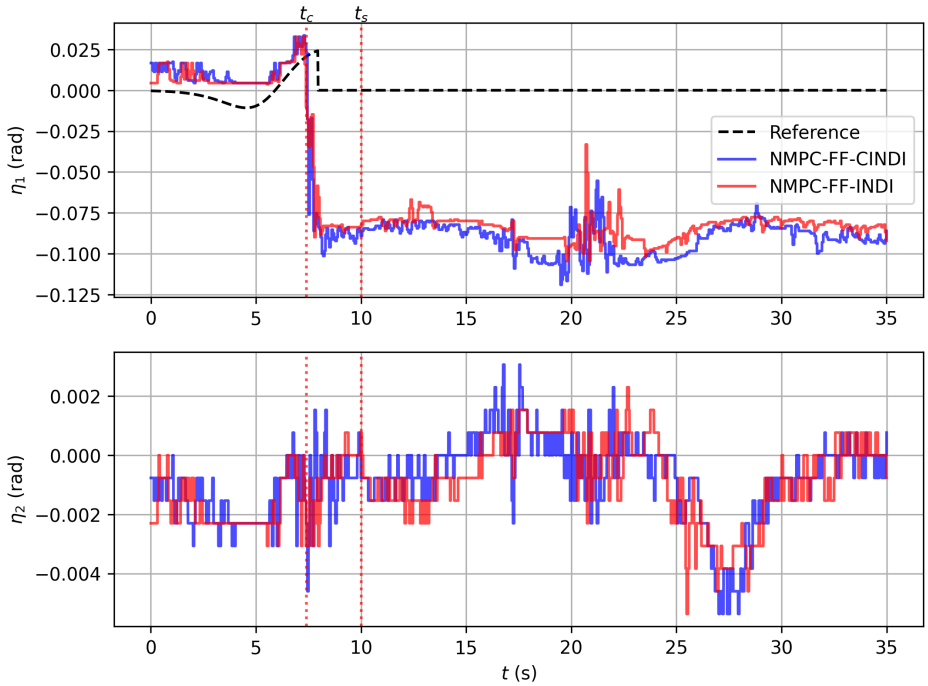


Figure 5-55: The joint angles in the real world eight drawing experiment on a vertical surface with no wind.

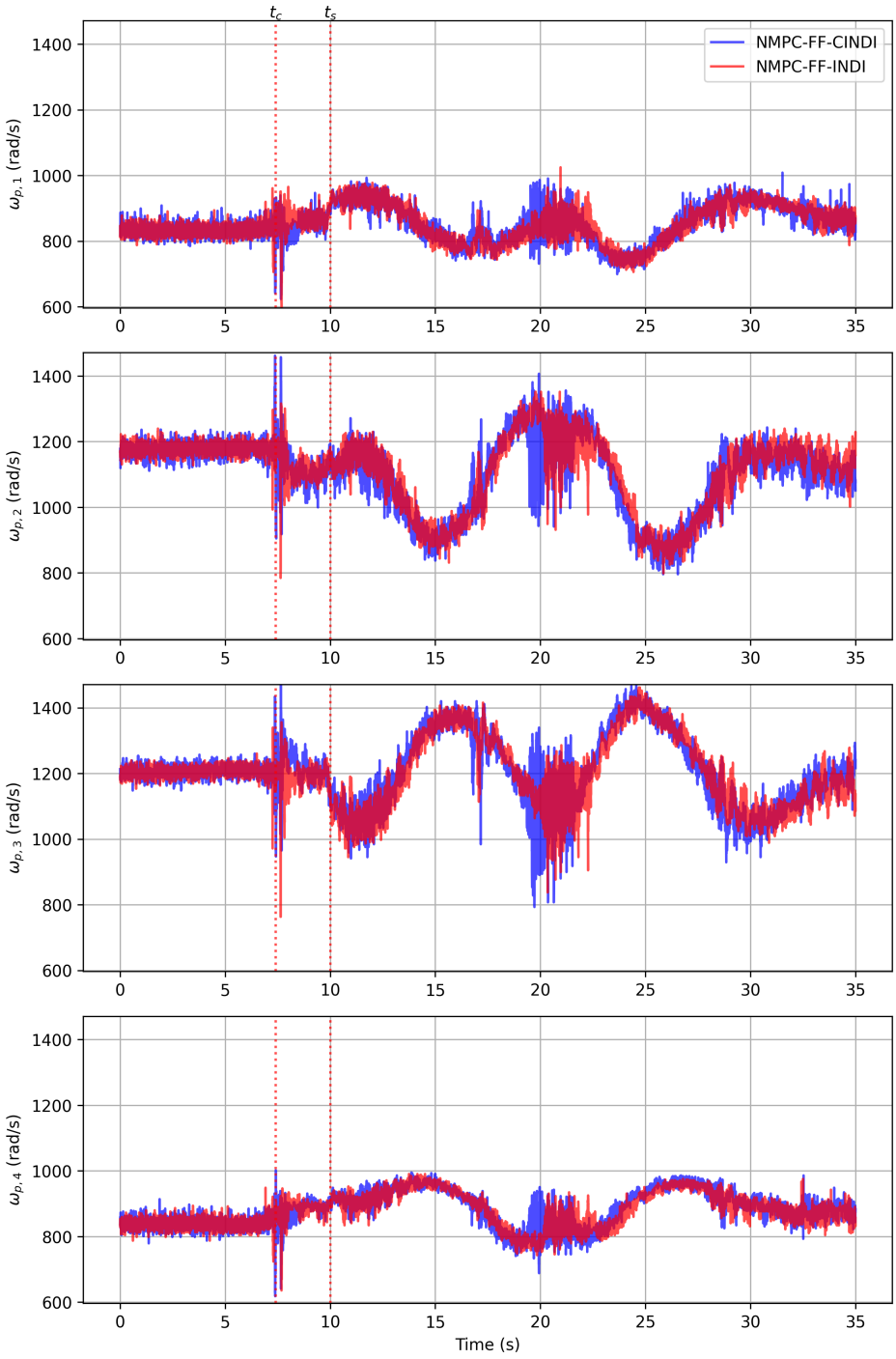


Figure 5-56: The rotor speeds commands for the real world eight drawing experiment on a vertical surface with no wind.

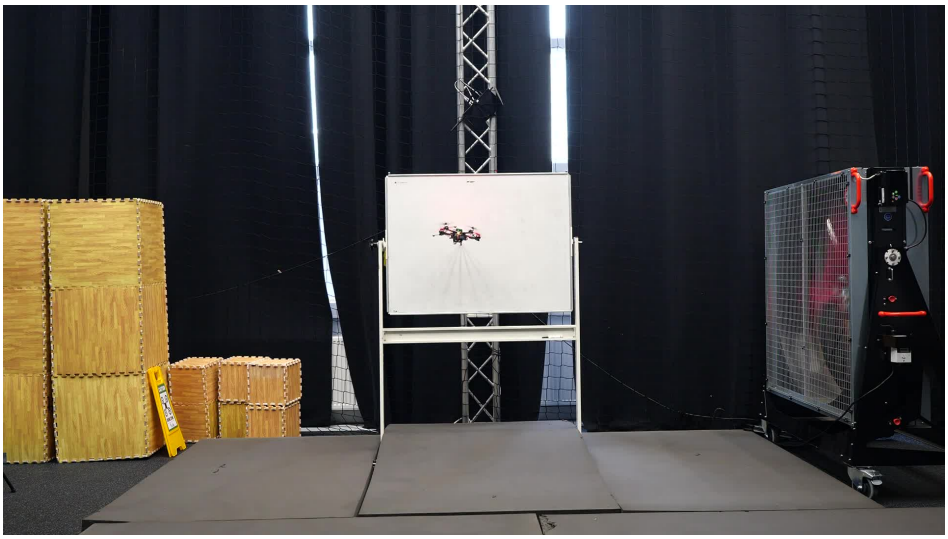


Figure 5-57: A snapshot of the eight drawing experiment on a whiteboard with external wind parallel to the whiteboard. The fan is located on the right of the whiteboard blowing wind speeds up to 5 m/s

Next we perform the same experiment but with wind blowing parallel to the whiteboard to introduce additional disturbances besides friction. The wind speeds were set to around 5 m/s. In Figure 5-57 a visual of the experimental setup is given. A side view of the end-effector position during the eight writing experiment with wind is shown in Figure 5-58. The position tracking errors of the end-effector are shown in Figure 5-59 and the absolute end-effector pose tracking errors are shown in Figure 5-60. The estimated contact forces are shown in Figure 5-61. The commanded rotor speeds are shown in Figure 5-62. Two times instants are highlighted in the figures with a dashed vertical line which correspond to the time at which contact occurs t_c and the time at which the sliding starts t_s.

The wind direction was orientated to be parallel to the wall and the center of the fan was placed near the same y-position as the whiteboard. This leads to large estimates F_p forces as can be seen in Figure 5-61. It seems to be around 1 N extra external estimated horizontal disturbance if we compare the estimated external forces in Figure 5-61 and in Figure 5-54. Furthermore it adds a lot more noise into the system as can be seen in all the state trajectories and inputs. The force tracking ability of both controllers do not differ much from the eight drawing experiment without external wind.

In this experiment the benefit of adding the position INDI layer is really shown by looking at the absolute end-effector position error in Figure 5-60, where NMPC-FF-CINDI reaches absolute position errors of around 12 cm while the NMPC-FF-INDI scheme reaches up to 22 cm tracking error. This is a reduction of around 45% maximum tracking error for this experiment. In Figure 5-58 it is clearly visible that both trajectories still have a significant steady-state offset due to the external disturbance in the world x-direction. INDI should in theory reject external disturbances asymptotically for the dynamics it is applied on. This does not happen however, due to the trajectory generation of the NMPC-FF layer. This layer is unaware of any external disturbance and will keep generating optimal trajectories given no external disturbances. Both NMPC-FF-INDI and NMPC-FF-CINDI are thus unable to

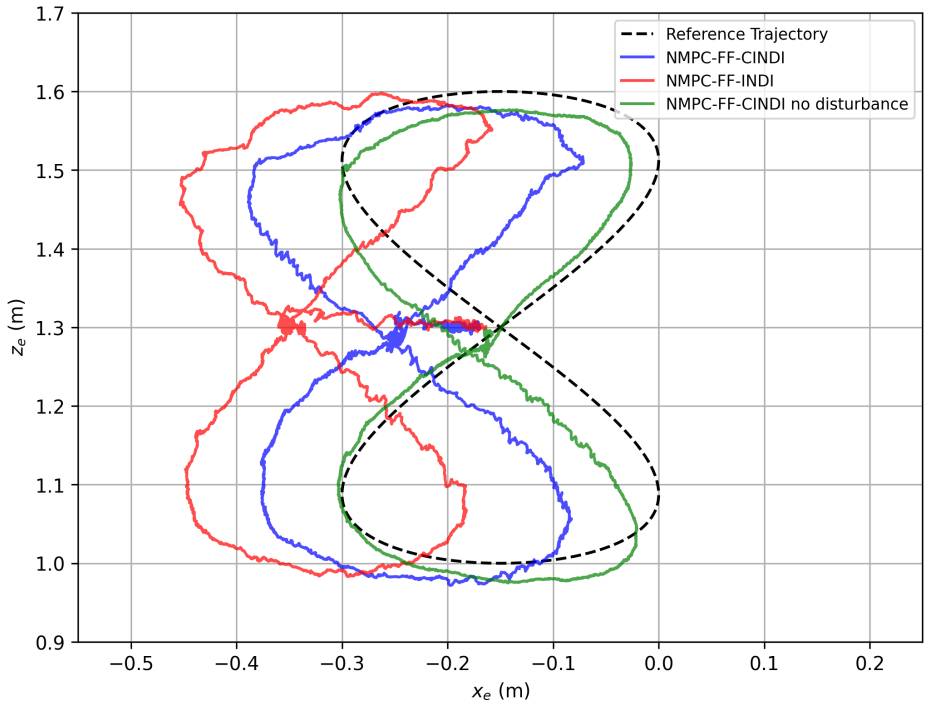


Figure 5-58: Side view of the end-effector position trajectory for the physical eight drawing experiment with wind blown parallel to the wall.

asymptotically reject constant disturbances which shows the limitations of the control schemes.

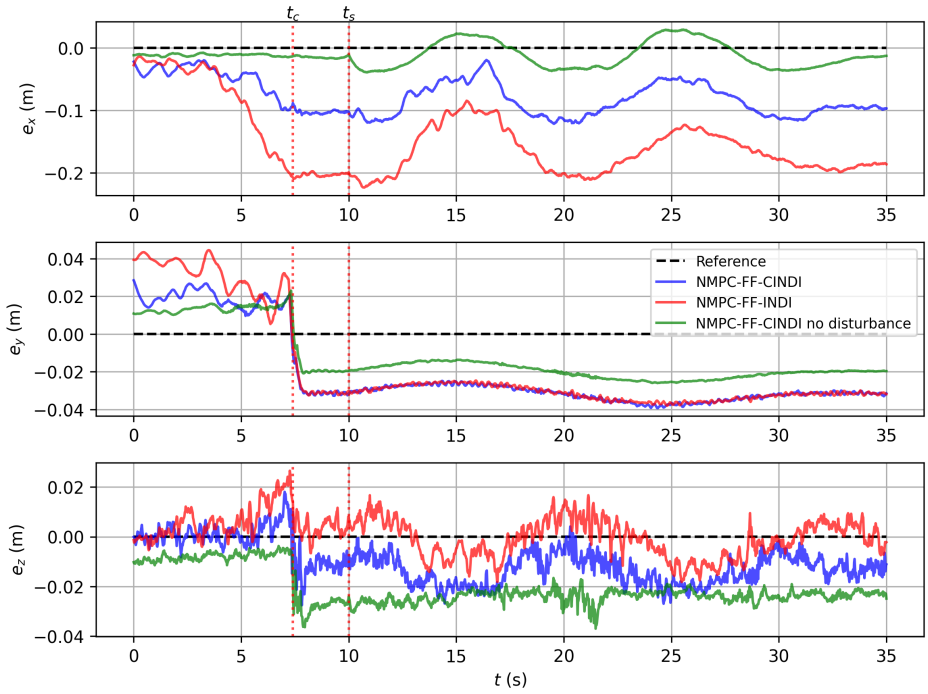


Figure 5-59: The end-effector position trajectory tracking errors in the physical eight drawing experiment with wind blown parallel to the whiteboard.

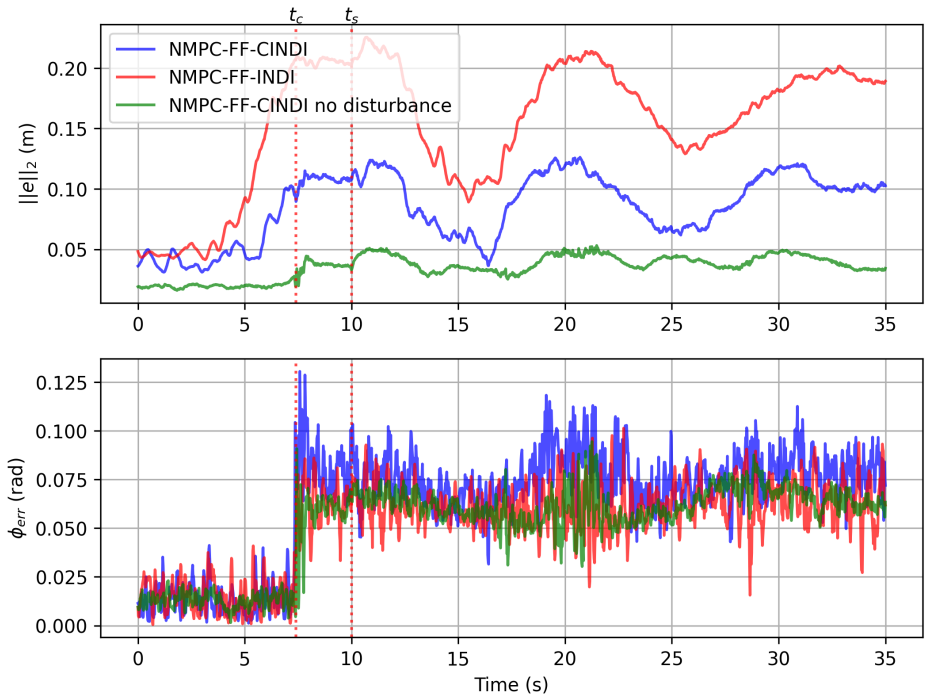


Figure 5-60: The absolute end-effector position and attitude tracking error in the physical eight drawing experiment with wind blown parallel to the whiteboard.

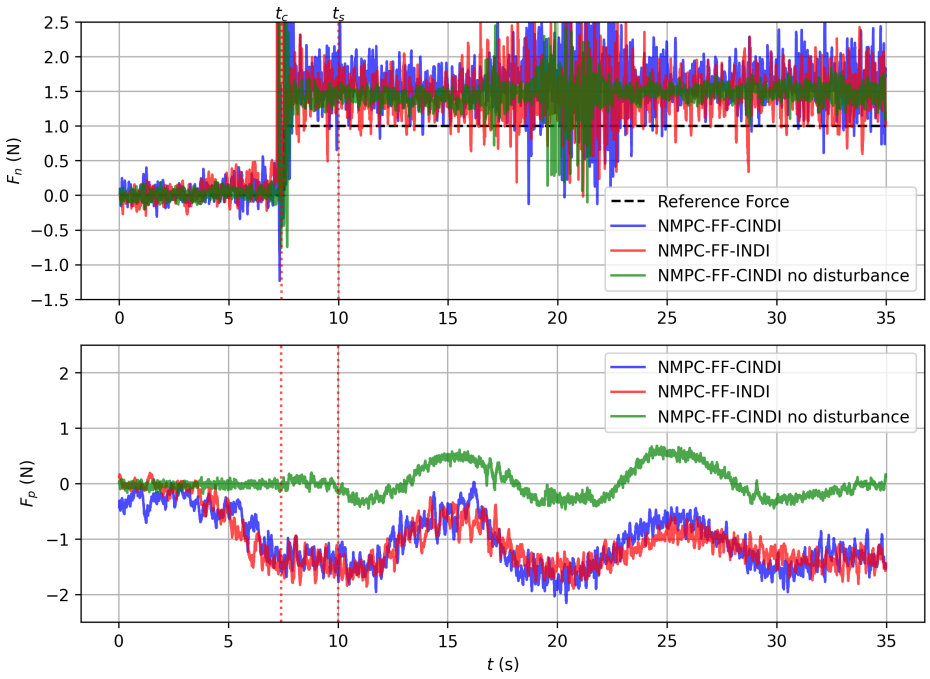


Figure 5-61: The estimated external forces in the physical eight drawing experiment with wind blown parallel to the whiteboard.

We also perform the same eight trajectory but with wind blowing perpendicular on the whiteboard. This will influence the estimation of F_n, but also the estimated contact force for the NMPC as it is in the direction of the surface normal. A side view of the end-effector position during the eight wall experiment with wind is shown in Figure 5-63. The position tracking errors of the end-effector are shown in Figure 5-64 and the absolute end-effector position and attitude tracking error are shown in Figure 5-65. The estimated contact forces are shown in Figure 5-66. The commanded rotor speeds are shown in Figure 5-67. Two times instants are highlighted in the figures with a dashed vertical line which correspond to the time at which contact occurs t_c and the time at which the sliding starts t_s.

We can see in Figure 5-66 that F_n is estimated around -0.8 N from the start. This makes sense as the wind was blowing from behind the aerial manipulator towards the whiteboard. During contact, the normal force is estimated to be around 1 N for both cascaded as non cascaded INDI control schemes. As the system is unable to distinguish between contact forces and other disturbances it is difficult to estimate the real contact force applied. But if we assume that the wind was applying an additional force of around 0.8 N on the aerial manipulator, the contact force would have been around 1.8 N while the reference force was 0.5 N. The system did however remain stable even though the real contact normal force was probably significantly larger than the reference and estimated force.

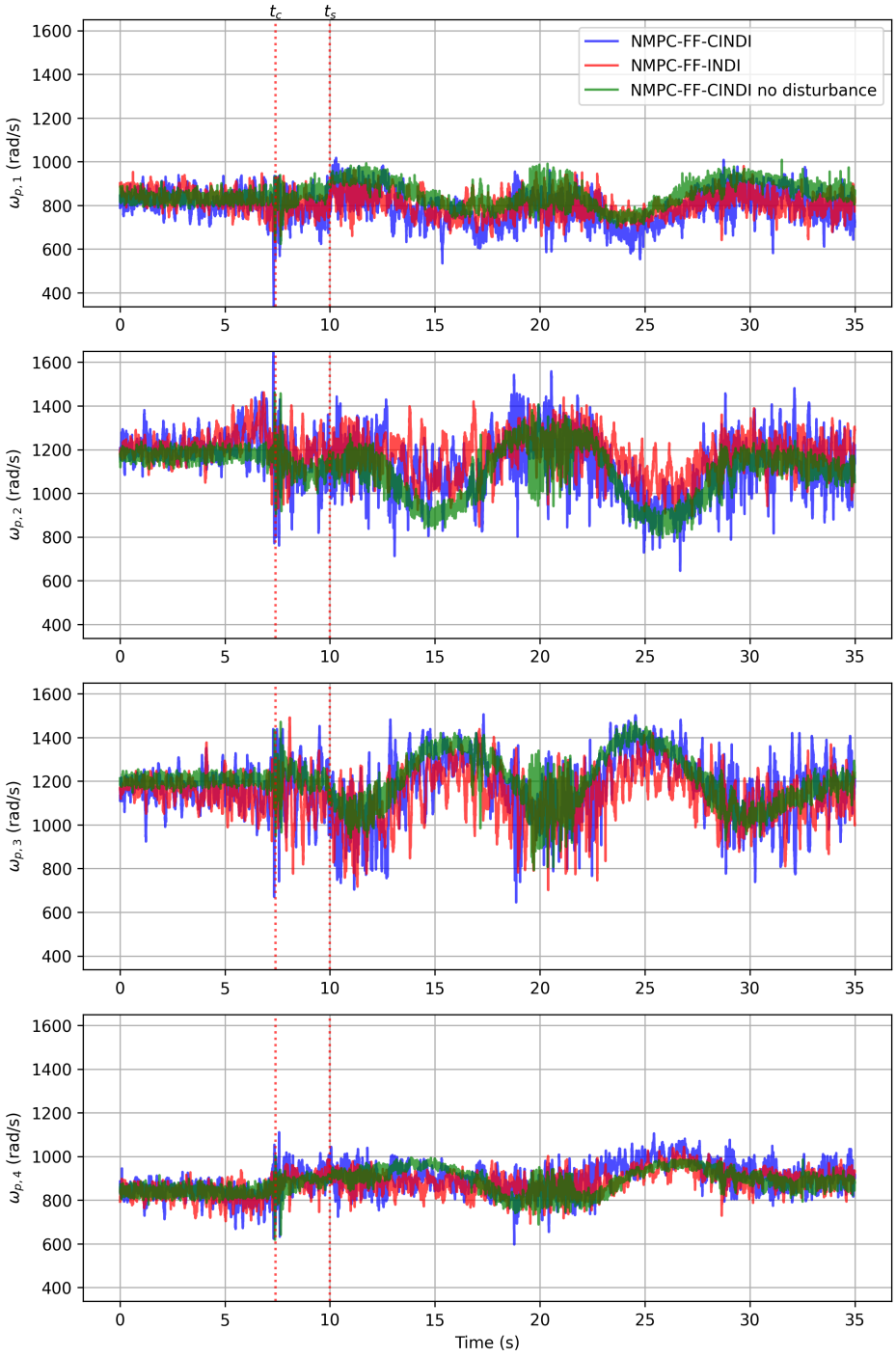


Figure 5-62: The rotor speeds commands for the physical eight drawing experiment with wind blown parallel to the whiteboard.

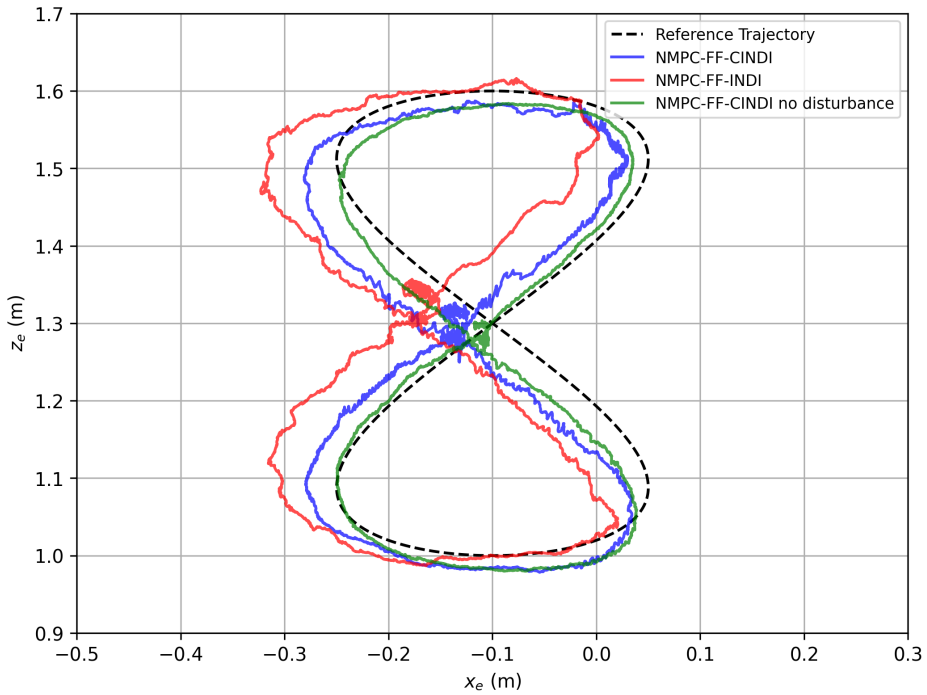


Figure 5-63: Side view of the end-effector position trajectory for the physical eight drawing experiment with wind blown perpendicular on the whiteboard.

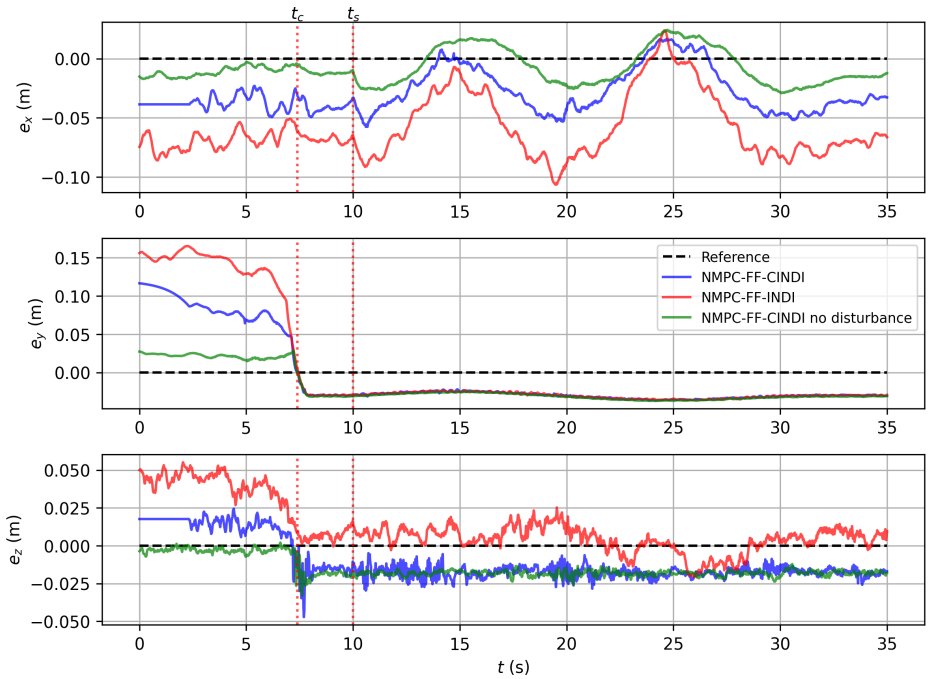


Figure 5-64: The end-effector position trajectory tracking errors in the physical eight drawing experiment with wind blown perpendicular on the whiteboard.

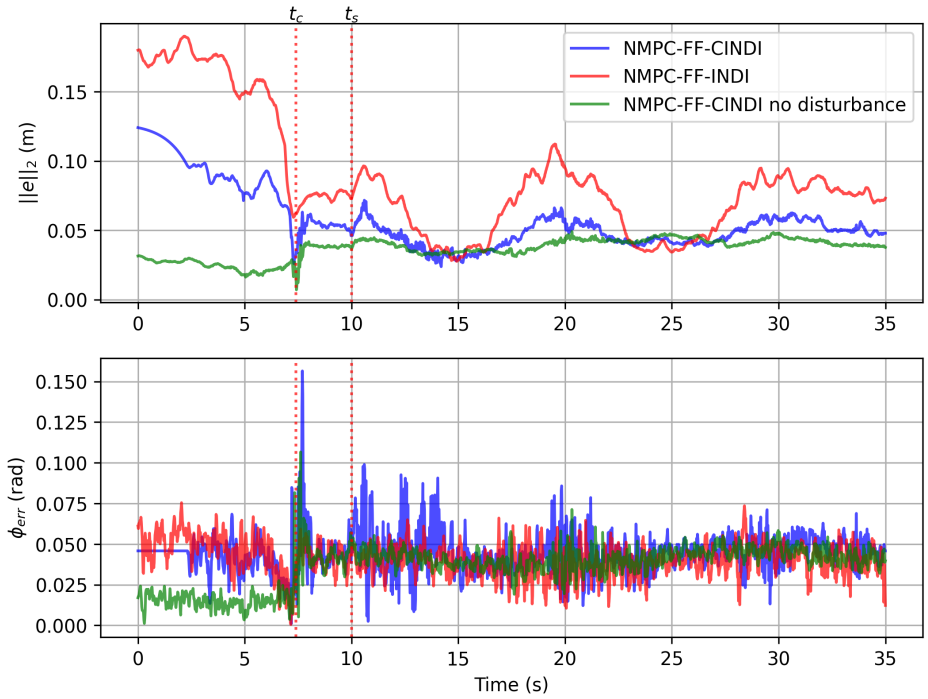


Figure 5-65: The absolute end-effector position and attitude error in the physical eight drawing experiment with wind blown perpendicular on the whiteboard.

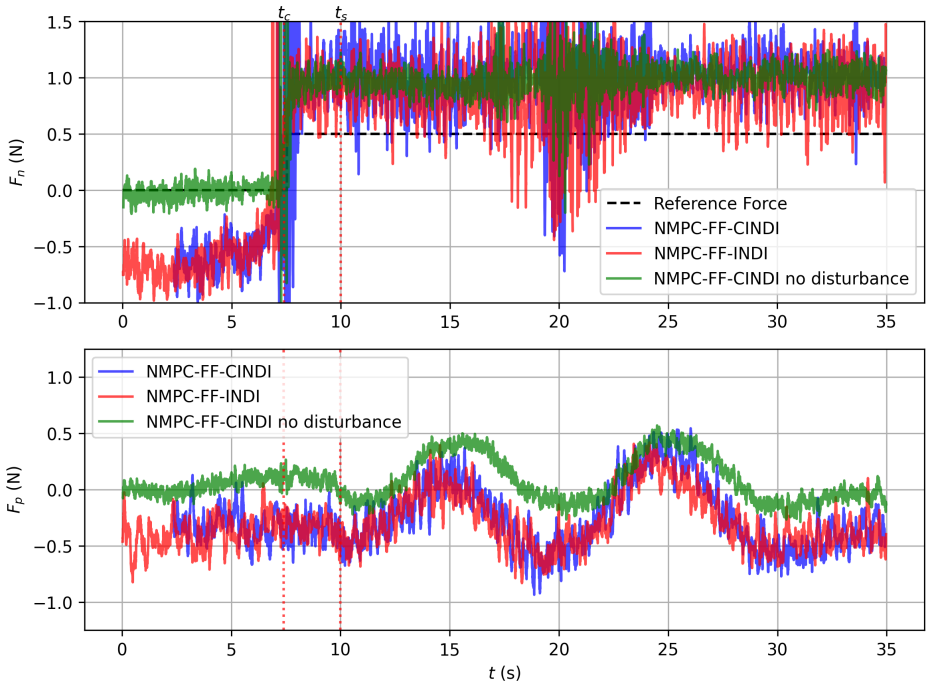


Figure 5-66: The estimated external forces in the physical eight drawing experiment with wind blown perpendicular on the whiteboard.

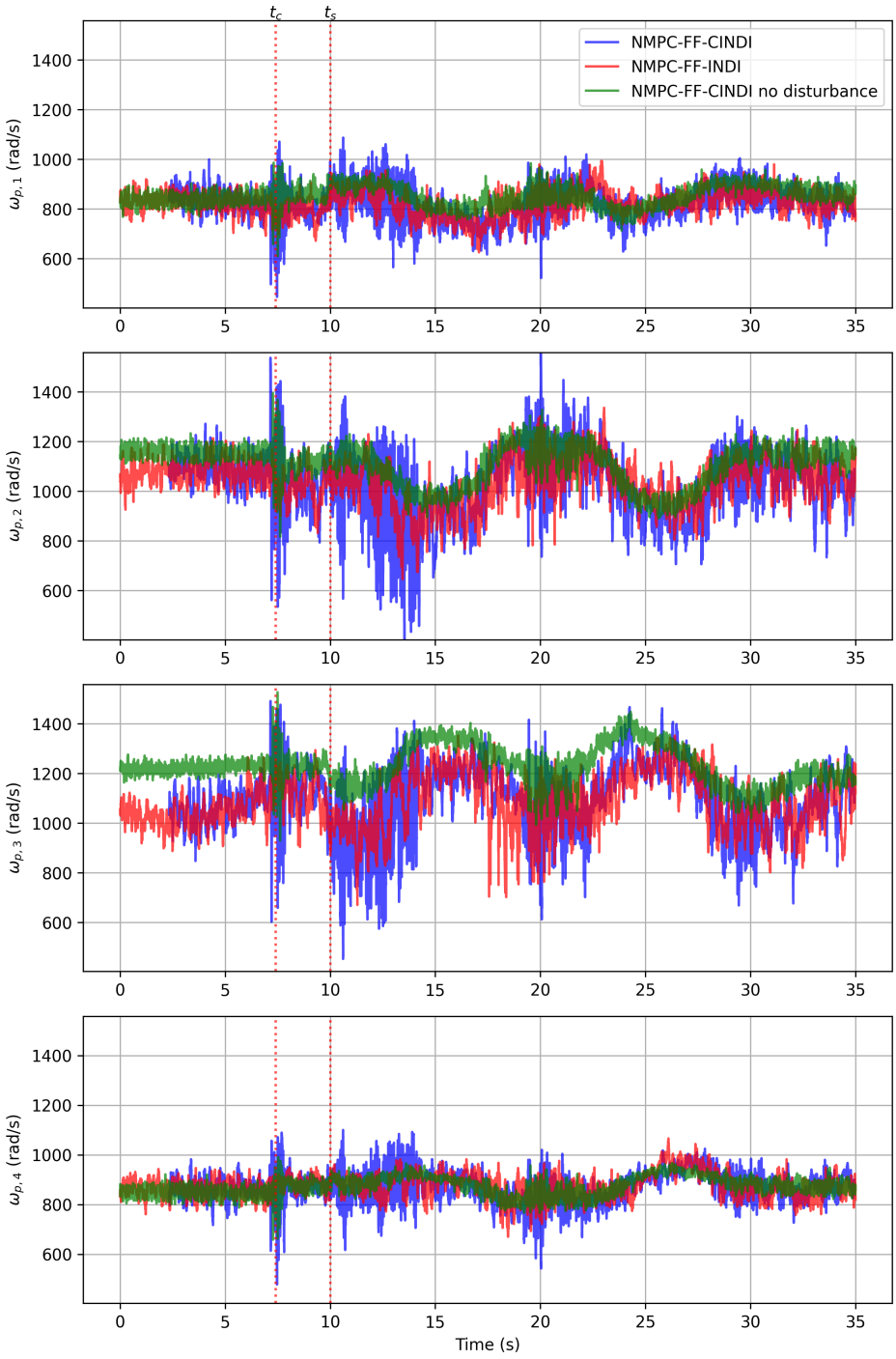


Figure 5-67: The rotor speeds commands for the physical eight drawing experiment with wind blown perpendicular on the whiteboard.

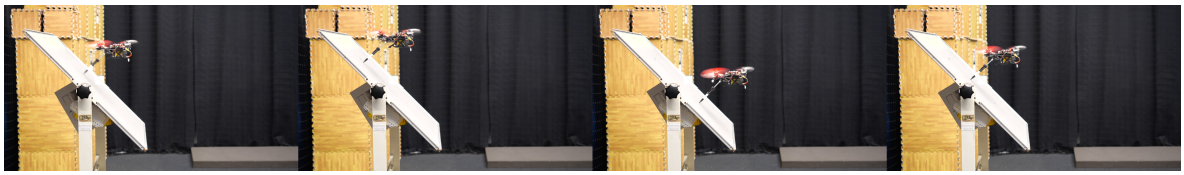


Figure 5-68: A visual montage of the eight drawing experiment on an inclined whiteboard

Also an eight drawing experiment on an inclined surface is performed similar to the one in simulation to show ability to perform sliding tasks on inclined surfaces also in real world experiments. The surface normal vector is $\hat{\boldsymbol{e}}_n = [0, -0.7071, 0.7071]^T$. A visual montage of the experiment is shown in Figure 5-68. A side view of the end-effector position during the inclined eight drawing experiment is shown in Figure 5-69 and in Figure 5-70. The position tracking errors of the end-effector are shown in Figure 5-71 and the absolute end-effector position and attitude tracking error are shown in Figure 5-73. The estimated contact forces are shown in Figure 5-74. The commanded rotor speeds are shown in Figure 5-75.

The controllers were able to perform the inclined eight drawing experiment in real world. The NMPC-FF-CINDI scheme had large high frequency oscillations after contact with the surface was made. It did manage to stabilize after around 4 seconds as can be seen in Figure 5-74 and in Figure 5-75. Reason for these oscillations may be due to increased aerodynamic interference due to the surface on the rotor thrusts. The cascaded INDI may be more affected by this due to the position INDI control layer. The absolute position tracking error of the end-effector seems to be lower for the cascaded controller by around 2 cm. This lower tracking error however seems related to the reduced steady-state offset of the NMPC-FF-CINDI controller compared to the NMPC-FF-INDI controller.

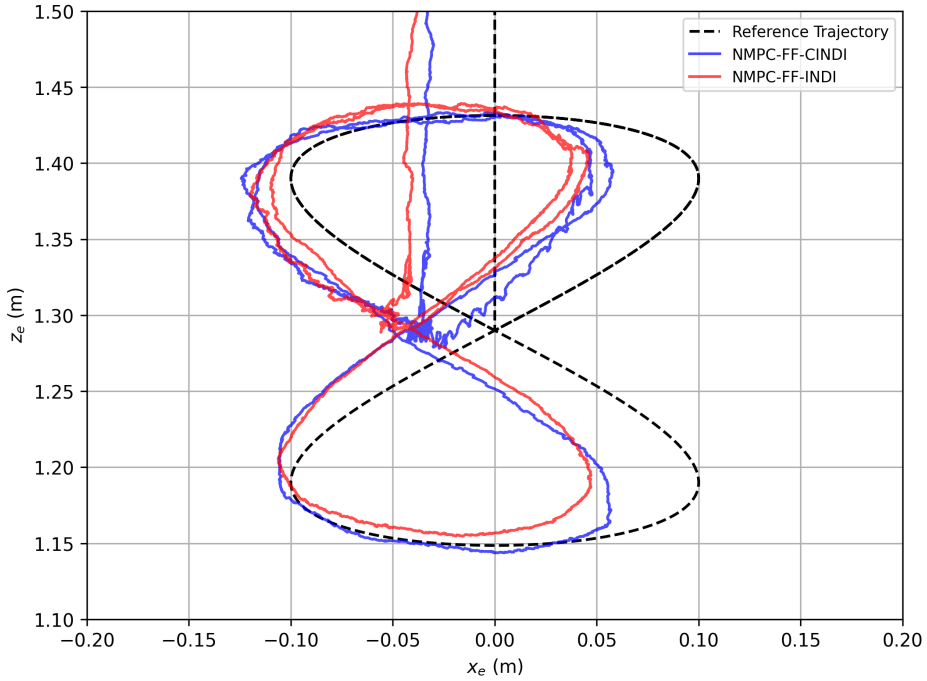


Figure 5-69: Side view of the end-effector position trajectory for the physical inclined eight drawing experiment with no wind.

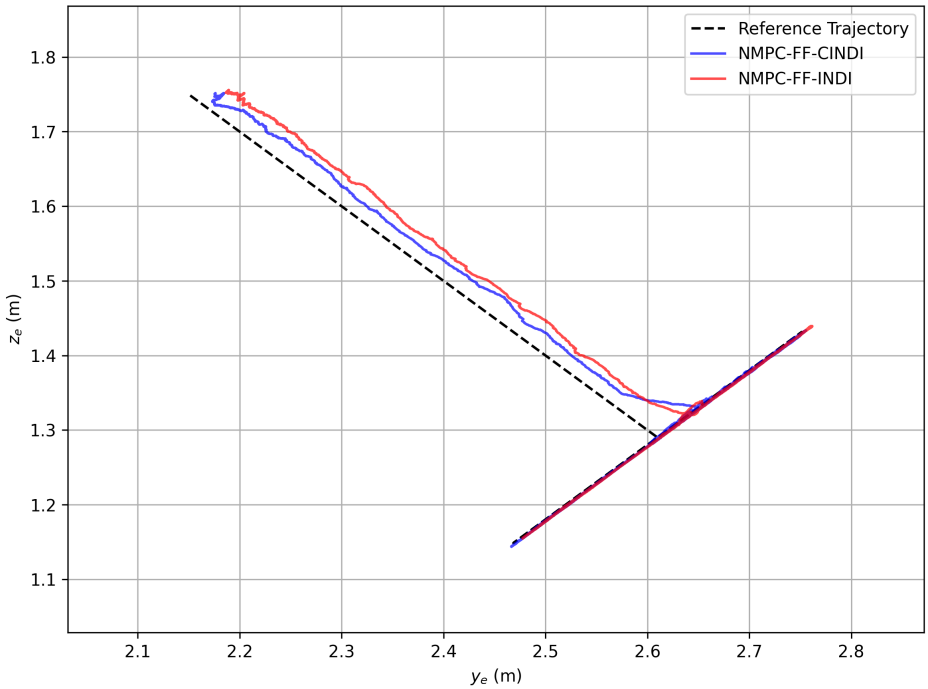


Figure 5-70: Side view of the end-effector position trajectory for the physical inclined eight drawing experiment with no wind.

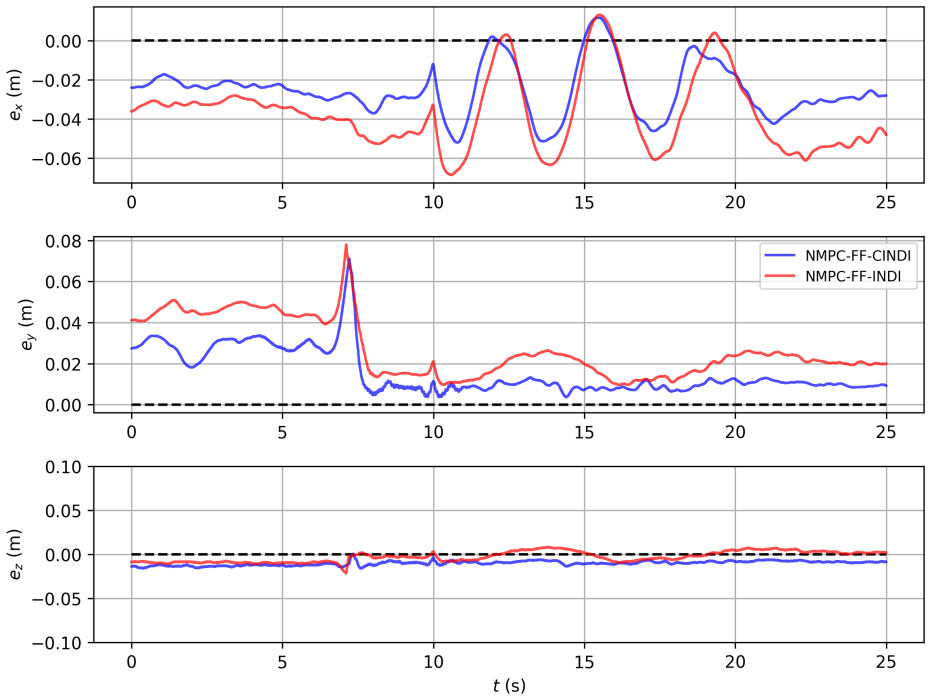


Figure 5-71: The end-effector position errors in the physical inclined eight drawing experiment.

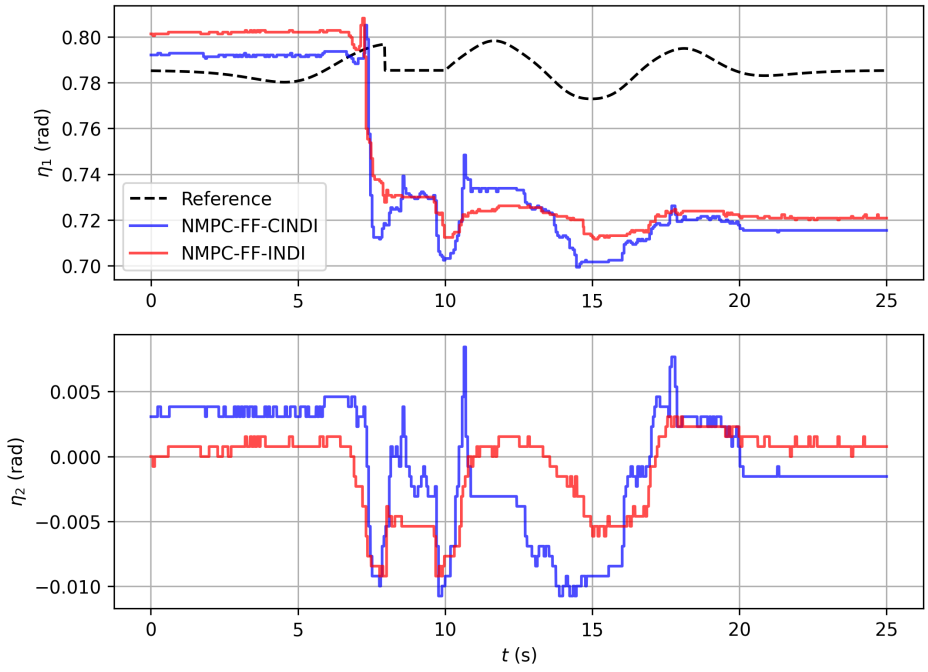


Figure 5-72: The joint positions in the physical inclined eight drawing experiment.

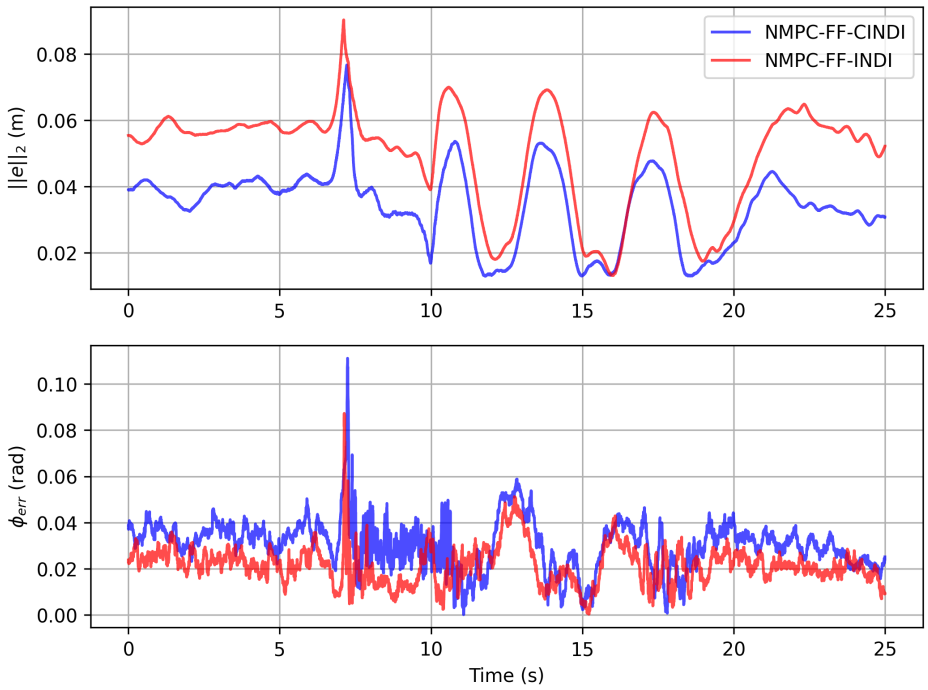


Figure 5-73: The absolute end-effector position and attitude error in the physical inclined eight drawing experiment.

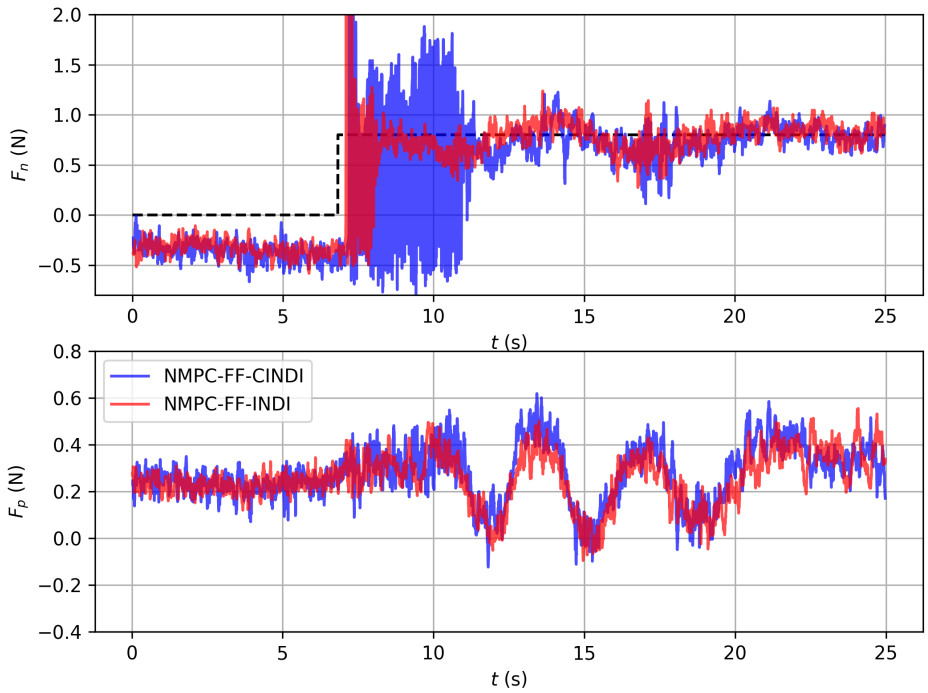


Figure 5-74: The estimated external forces in the physical inclined eight drawing experiment.

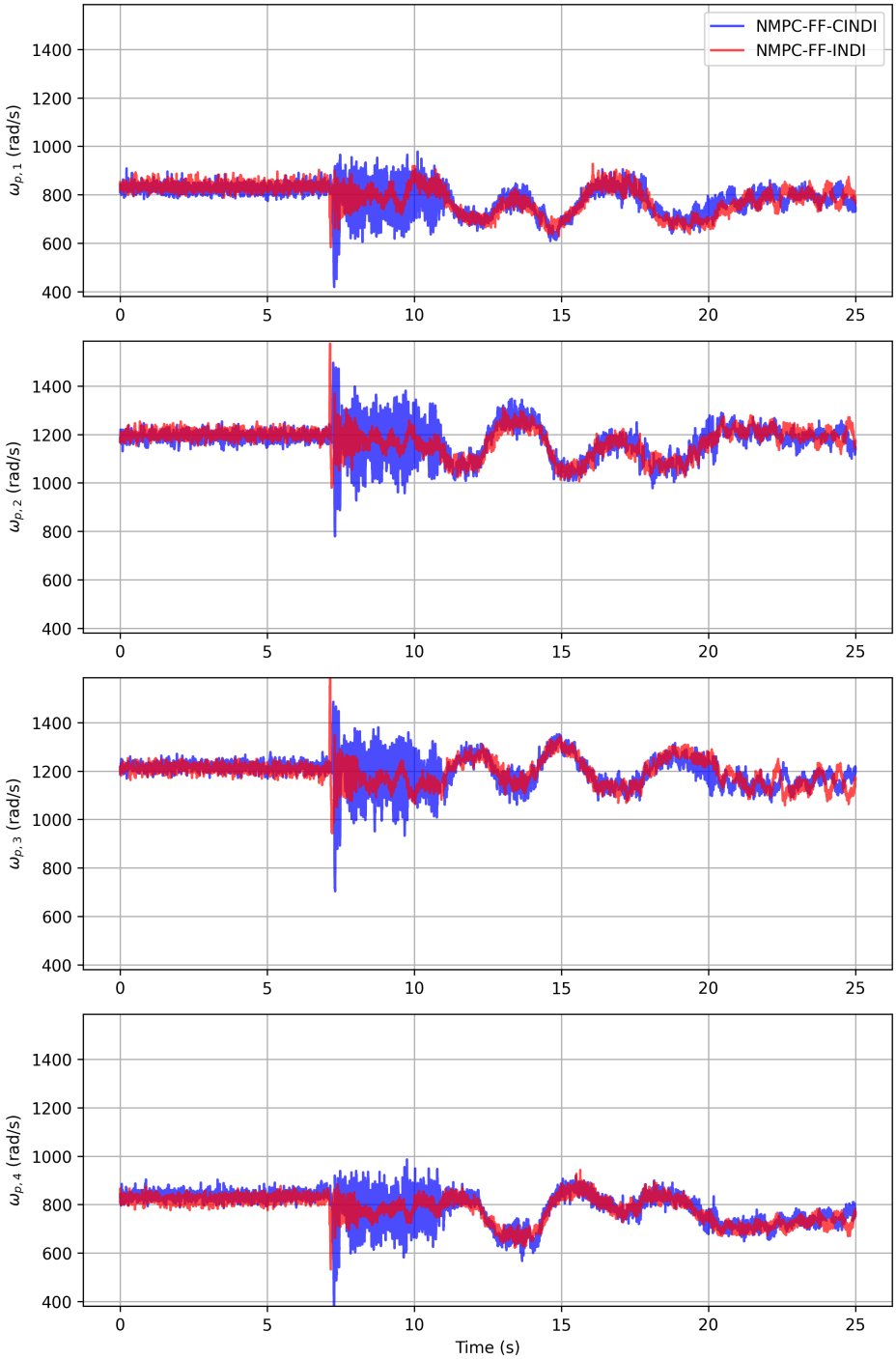


Figure 5-75: The rotor speeds commands for the physical inclined eight drawing experiment.

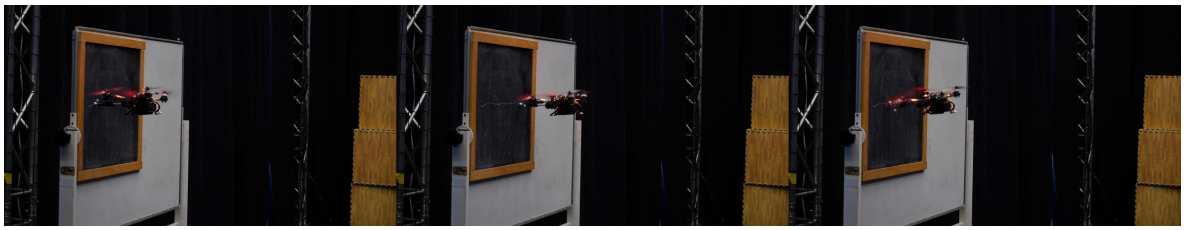


Figure 5-76: A montage of the drawing experiment using a crayon on a blackboard.

5-2-4 Aerial Writing: Blackboard

In this final section also an aerial sliding tasks using a crayon on a blackboard is performed. A visual montage of the experiment is shown in Figure 5-76. The relative arm-root end-effector position was changed in this experiment to ${}_B\boldsymbol{r}_{re} = [0.17, 0, 0]^T$, as the crayon was shorter than the marker. The end-effector trajectory follows a horizontal line on a blackboard which is oriented vertically. This sliding experiment is significantly more difficult than the whiteboard experiment, as the friction forces are higher for this task. This is partly because the static friction coefficient of the crayon is very high. Also the tip of a crayon is brittle, less compliant and larger compared to a marker. All these factors makes the sliding more difficult and this has clear effect on the stability of the DSAM during the experiments. In the end lowering the attitude cut-off frequency from 12 to 6 Hz made the system remain stable during the blackboard experiment described in this section.

An explanation of this instability may be the switching in dynamic and static friction forces in some end-effector velocity regimes can be viewed as high frequency disturbances due to the contact imperfections. The INDI attitude controller tries to compensate for this high frequency disturbance with some delay, which results in instability. By lowering the attitude low-pass filter cut-off frequency f_{att}, we are smoothing out the indirectly observed disturbance torques on the base in the angular acceleration measurements, which allows the controller to remain stable.

In Figure 5-77 the end-effector position is plotted over time, in Figure 5-78 the end-effector velocity is plotted during the experiment, in Figure 5-79 the absolute end-effector pose tracking errors are plotted over time. In the absolute end-effector pose tracking error we will be ignoring the error in the direction of the surface to remove the penetration depth offset. In Figure 5-80 the estimated external forces are plotted over time and in Figure 5-81 the motor speeds are plotted. We see that the controllers are both able to track the reference force of 1 N. The estimated friction forces reach up to 1 N. The ratio normal and friction contact force is significantly larger than for the whiteboard experiment. We see that NMPC-FF-CINDI is able to follow the end-effector reference faster and also reaches less steady-state error in tracking the reference compared to NMPC-FF-INDI. This is especially visible during the 10 seconds where $x_E = -0.2$. The maximum absolute tracking error of the cascaded INDI inner loop is around 8 cm during sliding while the attitude INDI inner loop has a maximum absolute tracking error of around 13 cm.

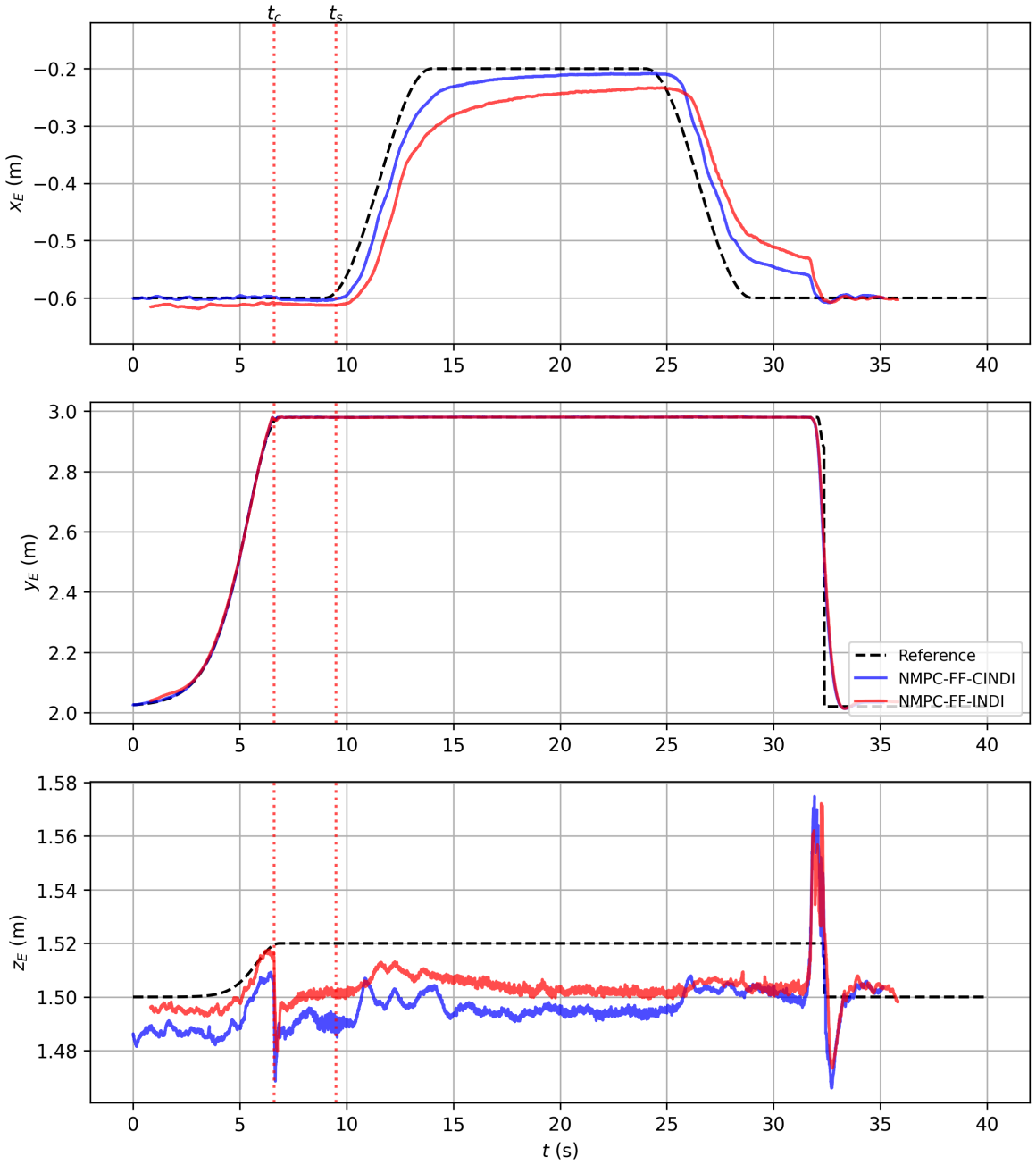


Figure 5-77: The end-effector position trajectory in the blackboard drawing experiment.

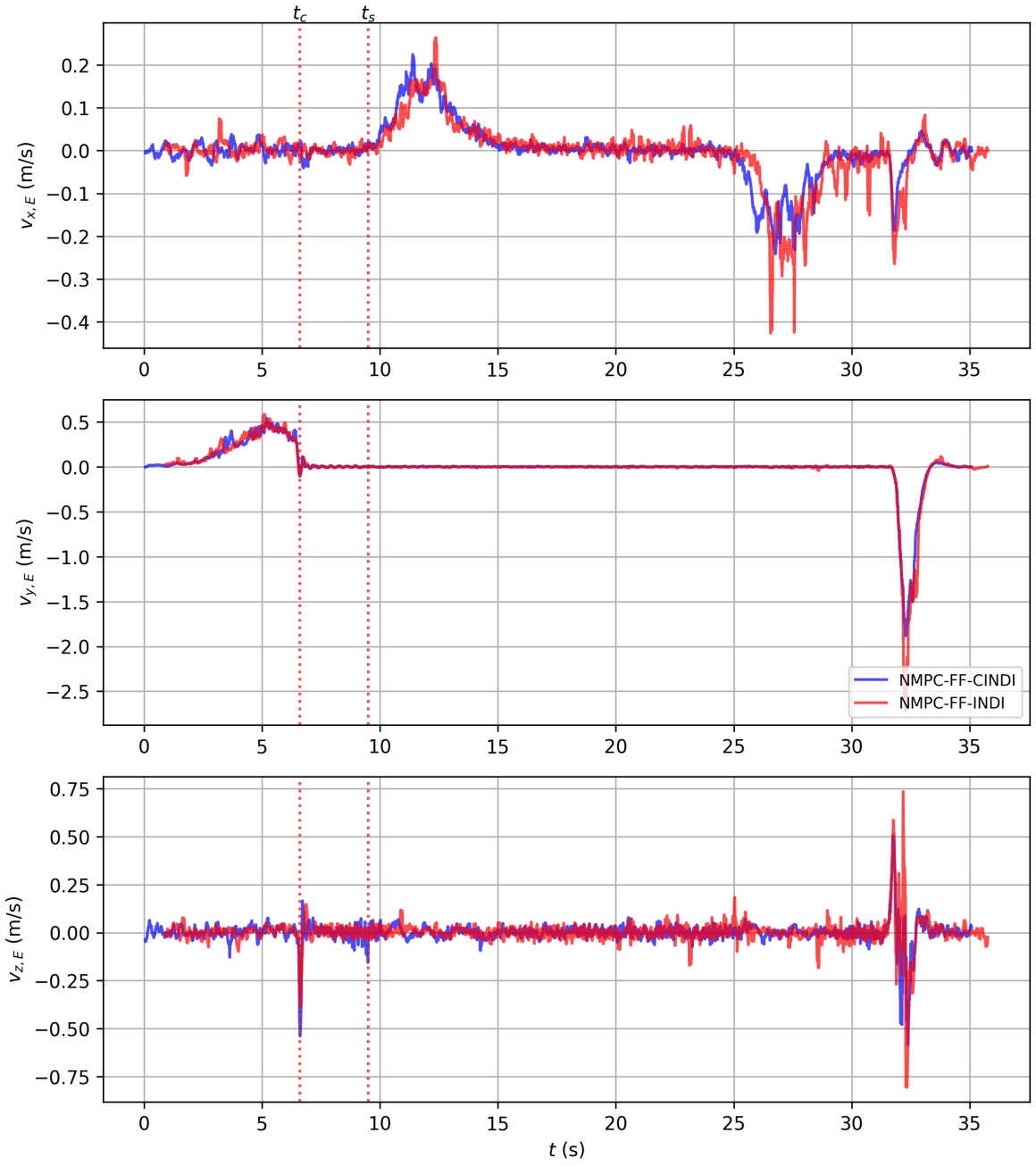


Figure 5-78: The end-effector velocity during the blackboard drawing experiment.

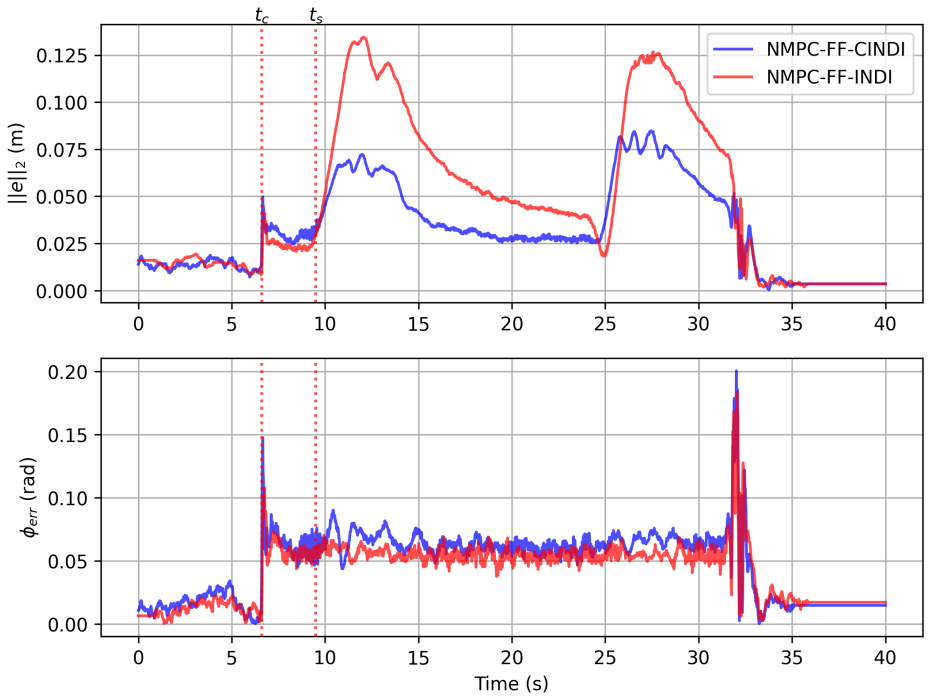


Figure 5-79: The absolute end-effector position and attitude error in the blackboard drawing experiment. The absolute end-effector position error $||e||_2$ is only considered in the xz-plane for this experiment.

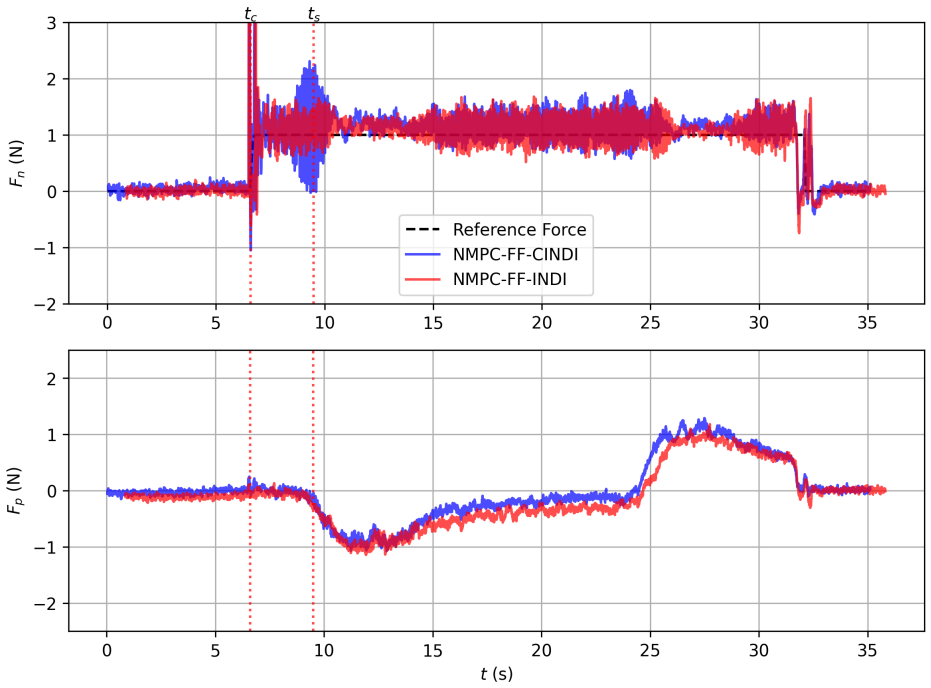


Figure 5-80: The estimated external forces in the blackboard drawing experiment.

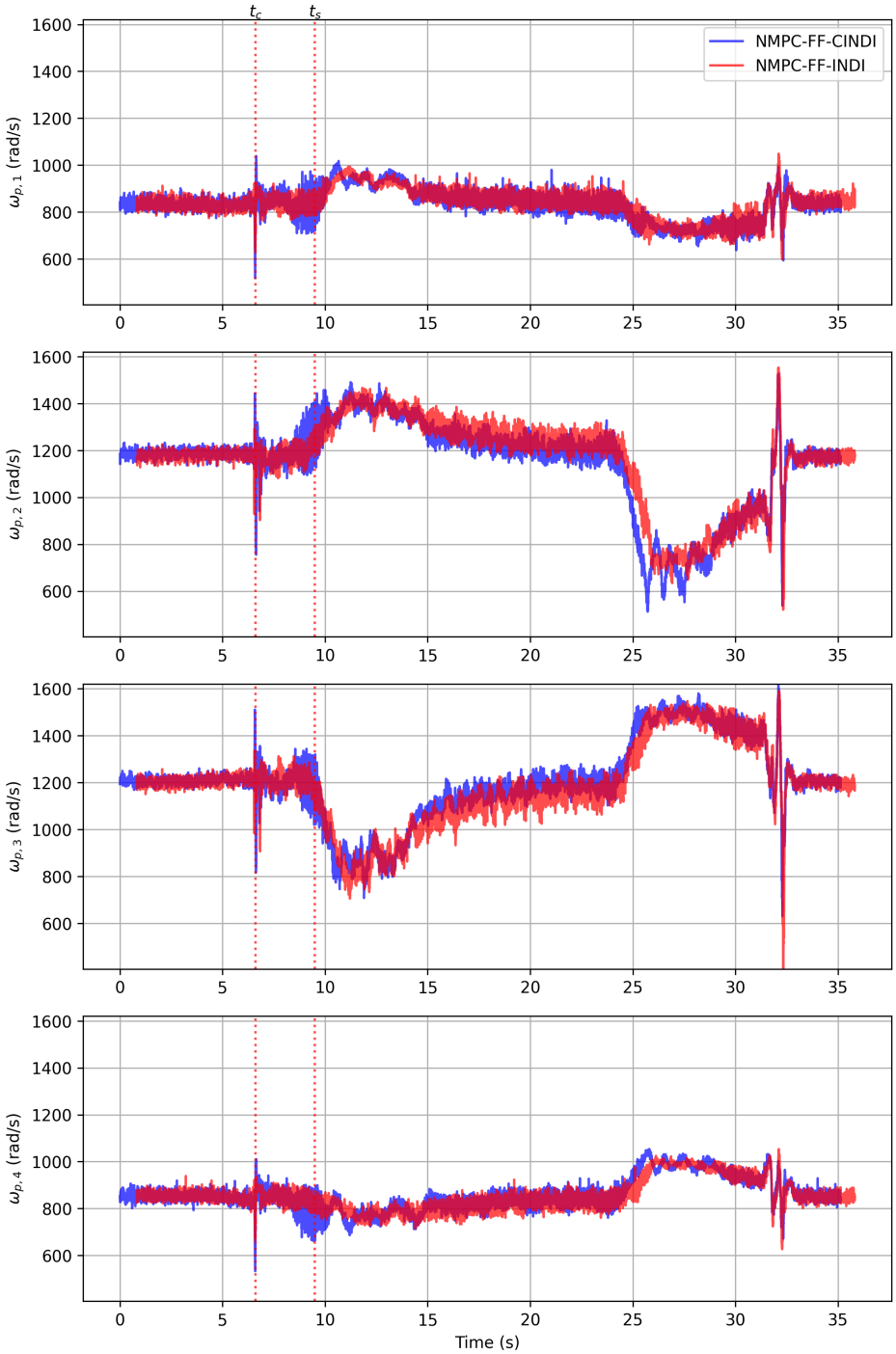


Figure 5-81: The rotor speed commands during the blackboard drawing experiment.

Chapter 6

Conclusions

In this work a modified Nonlinear Model Predictive Control (NMPC) control scheme is proposed for aerial manipulation, building upon a baseline NMPC formulation proposed in [55] for agile quadrotor flight. The NMPC is contact-aware and incorporates external force feedback, with a spring model introduced in the cost function in a manner similar to [59]. Alongside this, two whole-body inner-loop controllers were developed: an attitude Incremental Nonlinear Dynamic Inversion (INDI) controller and a cascaded INDI controller. Through an ablation study, it was demonstrated that adding the reference force in the dynamics as contact force improves force tracking performance, while incorporating external force feedback and a linear spring model in the cost function enables the NMPC to be combined with the cascaded INDI controller to track reference forces in arbitrary directions. Also the design choice of whole-body and decoupled control is shortly explored for the INDI method in simulation and whole-body control has shown benefit for stability of the controller when the arm mass is significant compared to the total aerial manipulator mass. Real-world aerial writing experiments validated the robustness of the proposed control scheme under real-world uncertainties, where the baseline NMPC failed to maintain stability. The attitude INDI controller allows the aerial manipulator to handle the unmodeled external torques on the base due to the sliding on different surfaces. Furthermore the INDI position layer seems to increase the friction force rejection visibly in the end-effector position tracking error. Also in presence of other external disturbances such as wind, the INDI position control layer increases disturbance rejection performance. The cascaded INDI inner loop achieved up to 45% less maximum end-effector tracking error compared to the attitude INDI inner loop during an eight-drawing experiment conducted under external wind disturbances with wind speeds of around 5 ms^{-1}. The force tracking of the controllers are consistent during sliding, but they all seem to overshoot the reference force. This is likely related to the penetration depth and orientation offset which were found during real world experiments. Lastly the drawing experiment is also shown to work on an inclined surface in real world experiments.

One practical improvement that can be made to the results obtained in this work is a better orientation calibration of the system. This may reduce the force and position tracking offsets currently found in the results. Also the joint reference from the reference trajectory generator

should account for contact forces. This change could make sure that the end-effector orientation is perpendicular to the contact surface during sliding.

The proposed control architecture consists of many control layers. This complexity is not desired and it may be beneficial to question the necessity of certain control layers in the NMPC-FF-CINDI scheme. Starting from the NMPC control scheme, each added layer does appear to offer some benefit over the baseline. However, it may be worth to study if the NMPC outer loop is really necessary, given that the reference trajectory fed into the NMPC is itself the output of an offline planning algorithm. While the predicted states from the NMPC appear to play a stabilizing role for the inner loop controllers, it also leads to a tracking offset in the presence of external disturbances. This is a notable limitation, as the use of INDI is theoretically expected to completely reject such disturbances.

In this work whole-body INDI is proposed and chosen over a decoupled approach to account for larger arm masses. In the ablation study in simulation it seems to indeed add benefit to account for the dynamic forces due to the presence of an arm. However in real world experiments both the whole-body and decoupled INDI controllers showed high frequency oscillations and even instability for large arm masses. Choosing a significantly lower arm mass than reality in the control model made the system able to hover with large arm mass. The reason why a nominal model combined with INDI was unable achieve satisfactory performance for large arm masses is unclear and additionally raises questions why there is such a difference between simulation and reality. Using the baseline NMPC the Differential Shoulder Aerial Manipulator (DSAM) was able to fly when using the nominal mass in the controller in real world experiments even for large payload masses on the end-effector. Understanding and finding a solution to this observation is crucial before INDI can be adopted for other general aerial manipulation tasks.

Several promising directions exist for future research. An alternative to the NMPC outer loop could be explored in combination with INDI. The addition of a disturbance observer to the outer loop could solve the asymptotic offset currently observed with the cascaded INDI scheme. On the theoretical side, stability guarantees for the NMPC formulation on the DSAM platform can be insightful and useful and lack in this work. Also as the NMPC is unaware of the INDI inner loops, formal stability guarantees on the total proposed formulation will probably not exist. A study on a unified NMPC formulation with for example an INDI-based tube controller could therefore be interesting and allow for some strong stability and robustness properties for nonlinear systems.

Applying INDI directly to a fully actuated platform would also be of interest, as it could simplify the proposed cascaded control architecture to a single INDI control layer. The effect of acceleration measurement filters on INDI performance is also interesting to explore further, as in this work only a simple second order low pass filter was used. The choice of cut-off frequency f_{att} in the INDI attitude loop lead to significantly different behavior during sliding, especially for the blackboard drawing experiment.

For the hardware side of the DSAM, transitioning from position to torque control on the manipulator arm could improve the whole-body control performance.

Finally, on a broader level, working towards greater autonomy of the DSAM by adding vision for example will be a useful next step. To add more tasks within the capabilities of the DSAM, learning algorithm can be explored such as reinforcement learning building further

upon the work in [17] or GP-NMPC to be able to adapt to new environments or configurations. Even though the point of this work was to ignore friction forces, it may also be of interest to study the proposed framework with an additional module to adapt to different surfaces and contact properties that may enable consistent performance during sliding on even more difficult and uneven surfaces. Also deploying the DSAM in real-world maintenance tasks such as non-destructive testing outside of a laboratory environment would also show some useful applications of the novel aerial manipulator system and controller design.

Appendix A

DSAM Kinematics and Dynamics

A-1 Kinematics

A Jacobian is defined such that it maps the generalized velocity vector $\boldsymbol{v}$ to the inertial twist (linear velocity $\dot{\boldsymbol{p}}$ and angular velocity $\boldsymbol{\omega}$) of the respective body:

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} J_P \\ J_R \end{bmatrix} \boldsymbol{v} \tag{A-1}$$

where $\boldsymbol{v} = [\boldsymbol{v}_b^T,\ {}_B\boldsymbol{\omega}_b^T,\ \dot{\boldsymbol{\eta}}^T]^T \in \mathbb{R}^8$. In this work we are going to use the Projected Newton-Euler Method (PNEM), which allows to simplify the dynamics by using mixed frames for the rotational and translational dynamics. As long as the Jacobian maps to the inertial velocities, it can be chosen in which frame these inertial velocities are expressed. For the Differential Shoulder Aerial Manipulator (DSAM) we choose to express the translational Jacobian in the inertial frame, and to express the rotational Jacobian in body frame B, so we obtain:

$$\begin{bmatrix} \dot{\boldsymbol{p}}_i \\ {}_B\boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} J_{P,i} \\ {}_BJ_{R,i} \end{bmatrix} \boldsymbol{v} \tag{A-2}$$

for each body $i \in \{B, A\}$.

To simplify the expressions, we define the skew-symmetric matrix operator $[\cdot]_\times$ for a vector $\boldsymbol{x} = [x_1, x_2, x_3]^T$:

$$[\boldsymbol{x}]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{A-3}$$

The base position is given by $\boldsymbol{p}_b$ and its orientation by R_{IB}. The velocities are directly available in the state vector $\boldsymbol{v}$. This gives the translational Jacobian ($J_{P,B} \in \mathbb{R}^{3\times 8}$):

$$J_{P,B} = \left[\ \boldsymbol{I}_3 \ \middle|\ \boldsymbol{0}_{3\times 3} \ \middle|\ \boldsymbol{0}_{3\times 2}\ \right] \tag{A-4}$$

The rotational Jacobian of the base (${}_B J_{R,B} \in \mathbb{R}^{3\times 8}$) is given as:

$$ {}_B J_{R,B} = \left[\begin{array}{c|c|c} \mathbf{0}_{3\times 3} & \boldsymbol{I}_3 & \mathbf{0}_{3\times 2} \end{array} \right] \tag{A-5} $$

as ${}_B\boldsymbol{\omega}_b$ is directly available from the generalized velocity vector.

The inertial position of A is $\boldsymbol{p}_A = \boldsymbol{p}_b + R_{IB}\, {}_B\boldsymbol{p}_{ba}$. Differentiating this with respect to time yields the translational Jacobian. The angular velocity is the sum of the base angular velocity and the arm angular velocity all expressed in the body frame.

We define the relative position of the arm CoG (A) with respect to the base (B), expressed in the base frame Σ_B, as ${}_B\boldsymbol{p}_{ba}$:

$$ {}_B\boldsymbol{p}_{ba} = {}_B\boldsymbol{p}_{br} + R_{BA}(\boldsymbol{\eta})\, {}_A\boldsymbol{p}_{ra} \tag{A-6} $$

Define the manipulator Jacobian $J_{\text{man}} \in \mathbb{R}^{6\times 2}$ which relates the joint velocities $\dot{\boldsymbol{\eta}}$ to the velocity of the arm CoG relative to the base, expressed in the base frame:

$$ J_{\text{man}}(\boldsymbol{\eta}) = \begin{bmatrix} J_{P,\text{man}} \\ J_{R,\text{man}} \end{bmatrix} \quad \text{such that} \quad \begin{array}{l} {}_B\boldsymbol{v}_{A/B} = J_{P,\text{man}}\dot{\boldsymbol{\eta}} \\ {}_B\boldsymbol{\omega}_{A/B} = J_{R,\text{man}}\dot{\boldsymbol{\eta}} \end{array} \tag{A-7} $$

The translational Jacobian of the arm ($J_{P,A} \in \mathbb{R}^{3\times 8}$) is given as:

$$ J_{P,A} = \left[\begin{array}{c|c|c} \boldsymbol{I}_3 & -R_{IB}[{}_B\boldsymbol{p}_{ba}]_\times & R_{IB} J_{P,\text{man}} \end{array} \right] \tag{A-8} $$

by applying the chain rule on the differentiation of inertial position vector of A.

The rotational Jacobian (${}_B J_{R,A} \in \mathbb{R}^{3\times 8}$) is given as:

$$ {}_B J_{R,A} = \left[\begin{array}{c|c|c} \mathbf{0}_{3\times 3} & \boldsymbol{I}_3 & J_{R,\text{man}} \end{array} \right] \tag{A-9} $$

Now we will derive the specific manipulator Jacobian in case of the DSAM. We start with the translational Jacobian which is derived by differentiating the relative position vector:

$$ {}_B\boldsymbol{p}_{ba} = {}_B\boldsymbol{p}_{br} + R_{BA}(\boldsymbol{\eta})\, {}_A\boldsymbol{p}_{ra} \tag{A-10} $$

The rotation matrix $R_{BA}(\boldsymbol{\eta})$ consists of two successive rotations, first around the body y-axis and then the x-axis: $R_{BA}(\boldsymbol{\eta}) = R_y(\eta_1)R_x(\eta_2)$.

Since ${}_B\boldsymbol{p}_{br}$ and ${}_A\boldsymbol{p}_{ra}$ are constant geometric offsets, the time derivative is:

$$ {}_B\dot{\boldsymbol{p}}_{ba} = \dot{R}_{BA}(\boldsymbol{\eta})\, {}_A\boldsymbol{p}_{ra} \tag{A-11} $$

Using the product rule on $R_{BA} = R_y(\eta_1)R_x(\eta_2)$:

$$ \dot{R}_{BA} = \dot{R}_y(\eta_1)R_x(\eta_2) + R_y(\eta_1)\dot{R}_x(\eta_2) \tag{A-12} $$

We use the standard identities for rotation matrix derivatives. For the first rotation η_1 about the fixed base y-axis ($\boldsymbol{e}_2$), and the second rotation η_2 about the local x-axis ($\boldsymbol{e}_1$):

$$ \dot{R}_y(\eta_1) = [\dot{\eta}_1\boldsymbol{e}_2]_\times R_y(\eta_1) \tag{A-13} $$

$$ \dot{R}_x(\eta_2) = R_x(\eta_2)[\dot{\eta}_2\boldsymbol{e}_1]_\times \tag{A-14} $$

Substituting these back into the velocity equation:

$$\begin{aligned} {}_B\dot{\boldsymbol{p}}_{ba} &= \left([\dot{\eta}_1 \boldsymbol{e}_2]_\times R_y(\eta_1)\right) R_x(\eta_2)\, {}_A\boldsymbol{p}_{ra} + R_y(\eta_1)\left(R_x(\eta_2)[\dot{\eta}_2 \boldsymbol{e}_1]_\times\right) {}_A\boldsymbol{p}_{ra} \\ &= [\dot{\eta}_1 \boldsymbol{e}_2]_\times \left(R_{BA}\, {}_A\boldsymbol{p}_{ra}\right) + R_{BA}\left([\dot{\eta}_2 \boldsymbol{e}_1]_\times\, {}_A\boldsymbol{p}_{ra}\right) \end{aligned} \tag{A-15}$$

Using the cross product property $[\boldsymbol{u}]_\times \boldsymbol{v} = -[\boldsymbol{v}]_\times \boldsymbol{u}$, we can rearrange to isolate $\dot{\eta}_1$ and $\dot{\eta}_2$:

$${}_B\dot{\boldsymbol{p}}_{ba} = -[R_{BA}\, {}_A\boldsymbol{p}_{ra}]_\times \boldsymbol{e}_2 \dot{\eta}_1 - R_{BA}[{}_A\boldsymbol{p}_{ra}]_\times \boldsymbol{e}_1 \dot{\eta}_2 \tag{A-16}$$

From this expression, the translational manipulator Jacobian $J_{P,\text{man}}$ can be extracted as:

$$J_{P,\text{man}} = \left[-[R_{BA}\, {}_A\boldsymbol{p}_{ra}]_\times \boldsymbol{e}_2, \quad -R_{BA}[{}_A\boldsymbol{p}_{ra}]_\times \boldsymbol{e}_1\right] \tag{A-17}$$

The rotational manipulator Jacobian $J_{R,\text{man}}$ is easier to derive as it is possible to add the angular velocities if expressed in the same frame. The angular velocity of the arm Center of Gravity (CoG) or frame A relative to base frame B is:

$${}_B\boldsymbol{\omega}_{A/B} = bse_2 \dot{\eta}_1 + R_{BA}\boldsymbol{e}_1 \dot{\eta}_2 \tag{A-18}$$

which yields the following rotational manipulator Jacobian:

$$J_{R,\text{man}} = \begin{bmatrix} \boldsymbol{e}_2 & R_{BA}\boldsymbol{e}_1 \end{bmatrix} \tag{A-19}$$

A-2 Generalized Mass Matrix

Now we have derived all the Jacobians relevant for the PNEM. To obtain the Equations of Motion (EoM) is now simply applying the PNEM which is described in Section 2-2. Here we will write out the mass matrix in block matrices, which is used for analyzing the different inner loop controllers in Chapter 4.

Let m_b, m_a be the masses of the base and arm, respectively. Let ${}_B\mathcal{I}_b$ be the inertia tensor of the base expressed in the base frame Σ_B, which is a known parameter. Let ${}_A\mathcal{I}_a$ be the inertia tensor of the arm expressed in the arm frame Σ_A, which is a known parameter.

We first define the arm inertia tensor rotated into the base frame:

$${}_B\mathcal{I}_a(\boldsymbol{\eta}) = R_{BA}(\boldsymbol{\eta})\, \mathcal{I}_a\, R_{BA}^T(\boldsymbol{\eta}) \tag{A-20}$$

The mass matrix $M(\boldsymbol{q}) \in \mathbb{R}^{8\times 8}$ is constructed by summing the contributions:

$$M(\boldsymbol{q}) = \sum_{i\in\{A,B\}} \left(J_{P,i}^T m_i J_{P,i} + {}_B J_{R,i}^T\, {}_B\mathcal{I}_i\, {}_B J_{R,i}\right) \tag{A-21}$$

We partition the mass matrix according to the generalized velocities $\boldsymbol{v} = [\boldsymbol{v}_b^T,\ {}_B\boldsymbol{\omega}_b^T,\ \dot{\boldsymbol{\eta}}^T]^T$:

$$M(\boldsymbol{q}) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{12}^T & M_{22} & M_{23} \\ M_{13}^T & M_{23}^T & M_{33} \end{bmatrix} \tag{A-22}$$

The symbolic expression for each block is given below. First we look at the diagonal entries, which are all square matrices. The first submatrix ($M_{11} \in \mathbb{R}^{3\times3}$) represents the total mass of the system and is purely diagonal:

$$M_{11} = (m_b + m_a)\boldsymbol{I}_3 \tag{A-23}$$

The second diagonal entry in the mass matrix is ($M_{22} \in \mathbb{R}^{3\times3}$), which represents the total inertia seen at the base frame. It includes the base inertia, the arm inertia (rotated), and the Steiner term (parallel axis theorem) accounting for the arm's offset position:

$$M_{22} = \mathcal{I}_b + {}_B\mathcal{I}_a - m_a[{}_B\boldsymbol{p}_{ba}]_\times[{}_B\boldsymbol{p}_{ba}]_\times \tag{A-24}$$

The third diagonal entry ($M_{33} \in \mathbb{R}^{2\times2}$) is the mass matrix of the manipulator arm itself, as if the base were fixed. The expression of this block is:

$$M_{33} = m_a J_{P,\text{man}}^T J_{P,\text{man}} + J_{R,\text{man}}^T {}_B\mathcal{I}_a J_{R,\text{man}} \tag{A-25}$$

As can be seen the term is completely independent of the base Jacobians and generalized coordinates related to the base.

Lastly there are three coupling blocks, which define how the translational dynamics, the rotational dynamics of the base and arm dynamics are related.

The coupling block ($M_{12} \in \mathbb{R}^{3\times3}$) defines the relation between the translational and angular dynamics of the DSAM. As the arm shifts the CoG of the aerial manipulator system from the quadrotor base, this also induces additional forces on the base with rotation. The expression of this block is given as:

$$M_{12} = -m_a R_{IB}[{}_B\boldsymbol{p}_{ba}]_\times \tag{A-26}$$

The next block ($M_{13} \in \mathbb{R}^{3\times2}$) couples inertial linear velocity with joint velocities. The expression is given as:

$$M_{13} = m_a R_{IB} J_{P,\text{man}} \tag{A-27}$$

The block ($M_{23} \in \mathbb{R}^{3\times2}$) couples base rotation with joint motion. It includes coupling from the arm's linear motion (via the lever arm) and the arm's rotational inertia. The expression is given as:

$$M_{23} = m_a[{}_B\boldsymbol{p}_{ba}]_\times J_{P,\text{man}} + {}_B\mathcal{I}_a J_{R,\text{man}} \tag{A-28}$$

Appendix B

NMPC Tunings

In this appendix the different tunings used in the disturbance rejection experiments in simulation are given. The weights are given in Table B-1.

Table B-1: NMPC weight changes for the disturbance rejection experiments in simulation

Experiment	$Q_{p,E}$	$Q_{q,E}$	$Q_{p,B}$
1	$\text{diag}([100, 100, 100])$	$\text{diag}([200, 200, 200])$	$\text{diag}([10, 10, 10])$
2	$\text{diag}([200, 200, 200])$	$\text{diag}([200, 200, 200])$	$\text{diag}([20, 20, 20])$
3	$\text{diag}([10, 10, 10])$	$\text{diag}([200, 200, 200])$	$\text{diag}([200, 200, 200])$
4	$\text{diag}([50, 50, 50])$	$\text{diag}([100, 100, 100])$	$\text{diag}([50, 50, 50])$
5	$\text{diag}([50, 50, 50])$	$\text{diag}([50, 50, 50])$	$\text{diag}([50, 50, 50])$

Bibliography

[1] Karim Ahmadi, Davood Asadi, Seyed-Yaser Nabavi-Chashmi, and Onder Tutsoy. Modified adaptive discrete-time incremental nonlinear dynamic inversion control for quad-rotors in the presence of motor faults. *Mechanical Systems and Signal Processing*, 188:109989, April 2023.

[2] A. R. P. Andriën, E. Lefeber, D. J. Antunes, and W. P. M. H. Heemels. Model Predictive Control for Quadcopters With Almost Global Trajectory Tracking Guarantees. *IEEE Transactions on Automatic Control*, 69(8):5216–5230, August 2024.

[3] Oualid Araar and Nabil Aouf. Visual servoing of a Quadrotor UAV for autonomous power lines inspection. In *22nd Mediterranean Conference on Control and Automation*, pages 1418–1424, June 2014.

[4] M Bernard, K Kondak, and G Hommel. Load transportation system based on autonomous small size helicopters. *The aeronautical journal*, 114(1153):191–198, 2010.

[5] Markus Bernard, Konstantin Kondak, Ivan Maza, and Anibal Ollero. Autonomous transportation and deployment with aerial robots for search and rescue missions. *Journal of Field Robotics*, 28(6):914–931, 2011.

[6] Karen Bodie, Maximilian Brunner, Michael Pantic, Stefan Walser, Patrick Pfändler, Ueli Angst, Roland Siegwart, and Juan Nieto. An Omnidirectional Aerial Manipulation Platform for Contact-Based Inspection. In *Robotics: Science and Systems XV*, June 2019.

[7] Karen Bodie, Maximilian Brunner, Michael Pantic, Stefan Walser, Patrick Pfändler, Ueli Angst, Roland Siegwart, and Juan Nieto. Active interaction force control for contact-based inspection with a fully actuated aerial vehicle. *IEEE Transactions on Robotics*, 37(3):709–722, 2021.

[8] Karen Bodie, Marco Tognon, and Roland Siegwart. Dynamic End Effector Tracking With an Omnidirectional Parallel Aerial Manipulator. *IEEE Robotics and Automation Letters*, 6(4):8165–8172, October 2021.

[9] Hossein Bonyan Khamseh, Farrokh Janabi-Sharifi, and Abdelkader Abdessameud. Aerial manipulation—A literature survey. *Robotics and Autonomous Systems*, 107:221–235, September 2018.

[10] Dario Brescianini and Raffaello D'Andrea. Tilt-prioritized quadrocopter attitude control. *IEEE Transactions on Control Systems Technology*, 28(2):376–387, 2018.

[11] Can Cakiroglu, Erik-Jan Van Kampen, and Q Ping Chu. Robust incremental nonlinear dynamic inversion control using angular accelerometer feedback. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 1128, 2018.

[12] Huazi Cao, Yongqi Li, Cunjia Liu, and Shiyu Zhao. ESO-Based Robust and High-Precision Tracking Control for Aerial Manipulation. *IEEE Transactions on Automation Science and Engineering*, 21(2):2139–2155, April 2024.

[13] E. Cataldi, G. Muscio, M. A. Trujillo, Y. Rodriguez, F. Pierri, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero. Impedance Control of an aerial-manipulator: Preliminary results. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3848–3853, October 2016.

[14] Wen-Hua Chen, D.J. Ballance, P.J. Gawthrop, and J. O'Reilly. A nonlinear disturbance observer for robotic manipulators. *IEEE Transactions on Industrial Electronics*, 47(4):932–938, August 2000.

[15] Eugenio Cuniato, Ismail Geles, Weixuan Zhang, Olov Andersson, Marco Tognon, and Roland Siegwart. Learning to Open Doors with an Aerial Manipulator. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6942–6948, October 2023.

[16] A. Das, K. Subbarao, and F. Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET Control Theory & Applications*, 3(3):303–314, March 2009.

[17] Shlok Deshmukh, Javier Alonso-Mora, and Sihao Sun. Global end-effector pose control of an underactuated aerial manipulator via reinforcement learning. *arXiv preprint arXiv:2512.21085*, 2025.

[18] Wei Dong, Zhao Ma, Xinjun Sheng, and Xiangyang Zhu. Centimeter-Level Aerial Assembly Achieved With Manipulating Condition Inference and Compliance. *IEEE/ASME Transactions on Mechatronics*, 27(3):1660–1671, June 2022.

[19] Dale Enns, Dan Bugajski, Russ Hendrick, and Gunter Stein. Dynamic inversion: an evolving methodology for flight control design. *International Journal of control*, 59(1):71–91, 1994.

[20] Matthias Faessler, Antonio Franchi, and Davide Scaramuzza. Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories. *IEEE Robotics and Automation Letters*, 3(2):620–626, April 2018.

[21] Paulo Flores and Hamid M Lankarani. *Contact force models for multibody dynamics*, volume 226. Springer, 2016.

[22] Philipp Foehn, Elia Kaufmann, Angel Romero, Robert Penicka, Sihao Sun, Leonard Bauersfeld, Thomas Laengle, Giovanni Cioffi, Yunlong Song, Antonio Loquercio, et al. Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight. *Science robotics*, 7(67):eabl6259, 2022.

[23] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi. Modeling and control of a flying robot for contact inspection. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3532–3537, October 2012.

[24] Gowtham Garimella and Marin Kobilarov. Towards model-predictive control for aerial pick-and-place. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4692–4697, May 2015.

[25] Xiaofeng Guo, Guanqi He, Jiahe Xu, Mohammadreza Mousaei, Junyi Geng, Sebastian Scherer, and Guanya Shi. Flying Calligrapher: Contact-Aware Motion and Force Planning and Control for Aerial Manipulation. *IEEE Robotics and Automation Letters*, 9(12):11194–11201, December 2024.

[26] Neville Hogan. Impedance Control: An Approach to Manipulation. In *1984 American Control Conference*, pages 304–313, June 1984.

[27] Yingzhi Huang, Ye Zhang, Daan M. Pool, Olaf Stroosma, and Qiping Chu. Time-Delay Margin and Robustness of Incremental Nonlinear Dynamic Inversion Control. *Journal of Guidance, Control, and Dynamics*, 45(2):394–404, 2022.

[28] Felix Huber, Konstantin Kondak, Kai Krieger, Dominik Sommer, Marc Schwarzbach, Maximilian Laiacker, Ingo Kossyk, Sven Parusel, Sami Haddadin, and Alin Albu-Schäffer. First analysis and experiments in aerial manipulation using fully actuated redundant robot arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3452–3457, November 2013.

[29] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

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

[31] Christopher Korpela, Matko Orsag, and Paul Oh. Towards valve turning using a dual-arm aerial manipulator. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3411–3416, September 2014.

[32] Dongjae Lee, Sunwoo Hwang, Jeonghyun Byun, Seung Jae Lee, and H. Jin Kim. Autonomous aerial perching and unperching using omnidirectional tiltrotor and switching controller. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1590–1596, May 2024.

[33] Dongjae Lee, Hoseong Seo, Dabin Kim, and H Jin Kim. Aerial manipulation using model predictive control for opening a hinged door. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1237–1242. IEEE, 2020.

[34] Hakjin Lee and Duck-Joo Lee. Rotor interactional effects on aerodynamic and noise characteristics of a small multirotor unmanned aerial vehicle. *Physics of Fluids*, 32(4):047107, April 2020.

[35] Vincenzo Lippiello and Fabio Ruggiero. Cartesian Impedance Control of a UAV with a Robotic Arm. *IFAC Proceedings Volumes*, 45(22):704–709, 2012.

[36] Zhenchang Liu, Yufei Zhang, JianJian Liang, and Haixin Chen. Application of the Improved Incremental Nonlinear Dynamic Inversion in Fixed-Wing UAV Flight Tests. *Journal of Aerospace Engineering*, 35(6):04022091, November 2022.

[37] Peng Lu, Erik-Jan van Kampen, Cornelis de Visser, and Qiping Chu. Aircraft fault-tolerant trajectory control using Incremental Nonlinear Dynamic Inversion. *Control Engineering Practice*, 57:126–141, December 2016.

[38] Tito J. Ludeña Cervantes, Seong H. Choi, and Byoung S. Kim. Flight Control Design using Incremental Nonlinear Dynamic Inversion with Fixed-lag Smoothing Estimation. *International Journal of Aeronautical and Space Sciences*, 21(4):1047–1058, December 2020.

[39] Dario Lunni, Angel Santamaria-Navarro, Roberto Rossi, Paolo Rocco, Luca Bascetta, and Juan Andrade-Cetto. Nonlinear model predictive control for aerial manipulation. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 87–93, June 2017.

[40] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Robotics & Automation Magazine*, 19(3):20–32, September 2012.

[41] Grzegorz Malczyk, Maximilian Brunner, Eugenio Cuniato, Marco Tognon, and Roland Siegwart. Multi-directional Interaction Force Control with an Aerial Manipulator Under External Disturbances. *Autonomous Robots*, 47(8):1325–1343, December 2023.

[42] Josep Martí-Saumell, Joan Sola, Angel Santamaria-Navarro, and Juan Andrade-Cetto. Full-body torque-level non-linear model predictive control for aerial manipulation, 2021.

[43] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, 2011.

[44] Abeje Y. Mersha, Stefano Stramigioli, and Raffaella Carloni. Variable impedance control for aerial interaction. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3435–3440, September 2014.

[45] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[46] Gabriele Nava, Quentin Sablé, Marco Tognon, Daniele Pucci, and Antonio Franchi. Direct Force Feedback Control and Online Multi-Task Optimization for Aerial Manipulators. *IEEE Robotics and Automation Letters*, 5(2):331–338, April 2020.

[47] Chanhong Park, Alex Ramirez-Serrano, and Mahdis Bisheban. Adaptive Incremental Nonlinear Dynamic Inversion Control for Aerial Manipulators With Swift Arm Motions. *AIAA SCITECH 2025 Forum*, January 2025.

[48] Markus Ryll, Giuseppe Muscio, Francesco Pierri, Elisabetta Cataldi, Gianluca Antonelli, Fabrizio Caccavale, Davide Bicego, and Antonio Franchi. 6D interaction control with aerial robots: The flying end-effector paradigm. *The International Journal of Robotics Research*, 38(9):1045–1062, August 2019.

[49] Pedro Sanchez-Cuevas, Guillermo Heredia, and Anibal Ollero. Characterization of the aerodynamic ground effect and its influence in multirotor control. *International Journal of Aerospace Engineering*, 2017(1):1823056, 2017.

[50] David Shim, Hoam Chung, Hyoun Jin Kim, and Shankar Sastry. Autonomous Exploration In Unknown Urban Environments For Unmanned Aerial Vehicles. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, August 2005. American Institute of Aeronautics and Astronautics.

[51] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.

[52] S. Sieberling, Q. P. Chu, and J. A. Mulder. Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction. *Journal of Guidance, Control, and Dynamics*, 33(6):1732–1742, November 2010.

[53] E.J.J. Smeur, G.C.H.E. De Croon, and Q. Chu. Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection. *Control Engineering Practice*, 73:79–90, April 2018.

[54] Ewoud J. J. Smeur, Qiping Chu, and Guido C. H. E. De Croon. Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, March 2016.

[55] Sihao Sun, Angel Romero, Philipp Foehn, Elia Kaufmann, and Davide Scaramuzza. A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight, January 2024.

[56] Suseong Kim, Seungwon Choi, and H. Jin Kim. Aerial manipulation using a quadrotor with a two DOF robotic arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4990–4995, Tokyo, November 2013. IEEE.

[57] Ezra Tal and Sertac Karaman. Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness. *IEEE Transactions on Control Systems Technology*, 29(3):1203–1218, May 2021.

[58] The MathWorks, Inc. Simscape multibody, February 2021. [Online; accessed Feb 2021].

[59] Dimos Tzoumanikas, Felix Graule, Qingyue Yan, Dhruv Shah, Marija Popović, and Stefan Leutenegger. Aerial manipulation using hybrid force and position nmpc applied to aerial writing. *Proceedings of Robotics: Science and Systems XVI*, 2020.

[60] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Jonathan Frey, Niels van Duijkeren, Andrea Zanelli, Branimir Novoselnik, Thivaharan Albin, Rien Quirynen, and Moritz Diehl. acados – a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 2021.

[61] Xuerui Wang, Erik-Jan van Kampen, Qiping Chu, and Peng Lu. Incremental sliding-mode fault-tolerant flight control. *Journal of guidance, control, and dynamics*, 42(2):244–259, 2019.

[62] Xuerui Wang, Erik-Jan van Kampen, Qiping Chu, and Peng Lu. Stability Analysis for Incremental Nonlinear Dynamic Inversion Control. *Journal of Guidance, Control, and Dynamics*, 42(5):1116–1129, 2019.

[63] Jake Welde, James Paulos, and Vijay Kumar. Dynamically Feasible Task Space Planning for Underactuated Aerial Manipulators. *IEEE Robotics and Automation Letters*, 6(2):3232–3239, April 2021.

[64] Burak Yüksel, Gabriele Buondonno, and Antonio Franchi. Differential flatness and control of protocentric aerial manipulators with any number of arms and mixed rigid-/elastic-joints. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 561–566. IEEE, 2016.

[65] Guangyu Zhang, Yuqing He, Bo Dai, Feng Gu, Jianda Han, and Guangjun Liu. Robust Control of an Aerial Manipulator Based on a Variable Inertia Parameters Model. *IEEE Transactions on Industrial Electronics*, 67(11):9515–9525, November 2020.

[66] Weixuan Zhang, Lionel Ott, Marco Tognon, and Roland Siegwart. Learning Variable Impedance Control for Aerial Sliding on Uneven Heterogeneous Surfaces by Proprioceptive and Tactile Sensing. *IEEE Robotics and Automation Letters*, 7(4):11275–11282, October 2022.

Glossary

List of Acronyms

NDI	Nonlinear Dynamic Inversion
INDI	Incremental Nonlinear Dynamic Inversion
MIMO	Multiple-Input Multiple-Output
SISO	Single-Input Single-Output
UAV	Unmanned Aerial Vehicle
APhI	Aerial Physical Interaction
DoF	degrees-of-freedom
SMC	Sliding Mode Control
DFBC	Differential Flatness Based Control
NMPC	Nonlinear Model Predictive Control
ISS	Input-to-State Stable
FA	Fully-Actuated
EoM	Equations of Motion
CoG	Center of Gravity
PNEM	Projected Newton-Euler Method
OCP	Optimal Control Problem
MPC	Model Predictive Control
IMU	Inertial Measurement Unit
EKF	Extended Kalman Filter
CCW	Counterclockwise
CW	Clockwise
MRL	Mobile Robotics Lab
DSAM	Differential Shoulder Aerial Manipulator