IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Predictive Admittance Control for Aerial Physical Interaction

Ayham Alharbat[1,2], Chiara Gabellieri[2], Abeje Y. Mersha[1] and Antonio Franchi[2,3]

***Abstract*—This paper introduces a novel approach for controlling aerial robots during physical interaction by integrating Admittance Control with Nonlinear Model Predictive Control (NMPC). Unlike existing methods, our technique incorporates the desired impedance dynamics directly into the NMPC prediction model, alongside the robot's dynamics. This allows for the explicit prediction of how the robot's impedance will respond to interaction forces within the prediction horizon. Consequently, our controller effectively tracks the desired impedance behavior during physical interaction while seamlessly transitioning to trajectory tracking in free motion, all while consistently respecting actuator constraints. The efficacy of this method is validated through real-time simulations and experiments involving physical interaction tasks with an aerial robot. Our findings demonstrate that, across most scenarios, our method significantly outperforms the state-of-the-art (which does not predict future impedance state), achieving a reduction in tracking error of up to 90%. Furthermore, the results indicate that our approach enables smoother and safer physical interaction, characterized by reduced oscillations and the absence of the unstable behavior observed with the state-of-the-art method in certain situations.**

***Index Terms*—Aerial Systems: Mechanics and Control, Compliance and Impedance Control, Optimization and Optimal Control, Aerial Physical Interaction, Physical Interaction Control.**

I. Introduction

AERIAL Robots (ARs) have been used in many tasks that require physical contact, such as physical inspection and maintenance and collaboration with humans, which lead to extensive research about aerial physical interaction control [1].

As in other robotics fields, physical interaction control methods in the aerial field can be categorized into two main categories, Direct Force control and Indirect Force control methods [2]. Many contributions have proposed the use of direct force control methods (also called hybrid force/motion control) for ARs when the task requires precise tracking of a desired force. For example, in [3], [4], [5], [6], hybrid force/motion controllers are proposed and validated for ARs with different configurations.

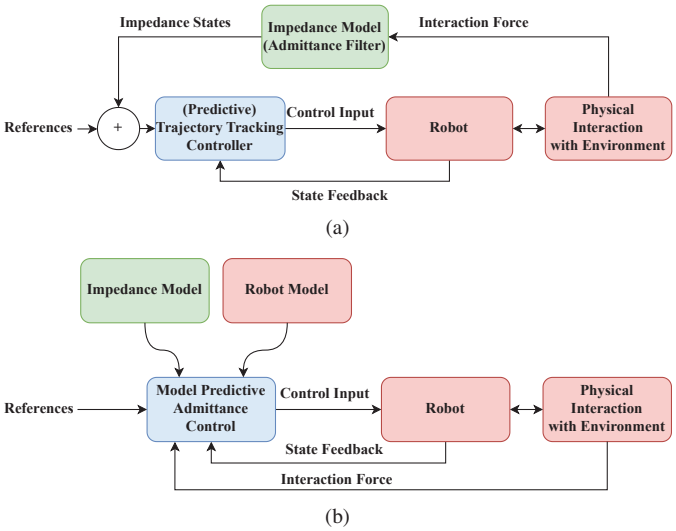

Fig. 1: Block diagram comparing (a) traditional admittance-control physical interaction control methods with (b) our proposed method that combines NMPC with admittance control.

The indirect force control methods have also been extensively studied for ARs when the task requires compliant, safe, or passive behavior, such as physical interaction with a human or an unknown environment. One of the most prominent paradigms in this category is Impedance Control [7], where the objective is to control the robot to behave as a desired mechanical impedance, characterized by a desired apparent mass, damping, and stiffness. One of the main limitations of this approach is the trade-off between compliance during physical interaction and motion tracking accuracy during free motion [2].

One proposed solution for this drawback is to separate the impedance control and motion control, which is called *Admittance Control*. This paradigm allows the robot to track the desired motion and reject disturbances in the absence of physical interaction, through an inner-loop trajectory tracking controller. Still, during physical interaction, an outer-loop admittance filter is used to modify the desired trajectory so that the robot exhibits the dynamic behavior of a desired impedance system, when subjected to the same interaction wrench, as illustrated in Fig. 1a. This method was studied and used extensively in the literature, as in [8], [9], [10], [11].

However, this approach does not guarantee that the controller will respect the system's constraints, which might lead

Manuscript received: May 19, 2025; Revised August 12, 2025; Accepted August 29, 2025.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments.

This work was partially supported by Regioorgaan SIA under projects MARS4EARTH (RAAK.PRO03.112) and MAESTRO-Drone (RAAK.MKB21.026), and the Horizon Europe research projects 101071270 (AeroSTREAM) and 101120732 (AUTOASSESS).

[1]Smart Mechatronics and Robotics Research Group, Saxion University of Applied Science, 7513 AB Enschede, The Netherlands. a.alharbat@saxion.nl, a.y.mersha@saxion.nl.

[2]Robotics and Mechatronics Department, Electrical Engineering, Mathematics, and Computer Science (EEMCS) Faculty, University of Twente, 7500 AE Enschede, The Netherlands. c.gabellieri@utwente.nl, a.franchi@utwente.nl.

[3]Department of Computer, Control and Management Engineering, Sapienza University of Rome, 00185 Rome, Italy.

Supplementary video: https://youtu.be/oJJVfXT0mME

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

to instability or unsafe behavior [12]. Moreover, it does not exploit knowledge of the system dynamics or the desired impedance dynamics to compute optimal control inputs.

To guarantee that the controller respects the hard constraints of the system, [13] proposed the use of an Nonlinear Model Predictive Control (NMPC)-based trajectory tracking controller in the inner loop with an outer loop admittance filter. While this architecture enforces constraint satisfaction at the trajectory tracking level, the outer-loop admittance filter remains reactive and only provides trajectory reference modifications, relying only on instantaneous measurements without anticipating future states or interactions. As a result, this approach does not fully exploit the key advantage of MPC — its predictive capability to optimize control actions based on the predicted evolution of the system and its references.

Aside from admittance control, several ideas were proposed in the literature to combine MPC and impedance control, for example, in [14], an MPC is designed to be equivalent to an impedance controller if the constraints are not active, and in [15], a control scheme that combines NMPC with impedance control is proposed, where an impedance control law is designed to shape the dynamic behavior of the robot as the desired impedance. However, these methods use impedance control rather than admittance control, which usually suffers from poor tracking behavior during free motion [2], and have to rely on variable impedance, such that the impedance is stiff to reject disturbances during free motion, and compliant during physical interaction [16].

To summarize, the state-of-the-art method for aerial physical interaction control based on Admittance Control and NMPC [13] exploits the prediction capabilities of MPC by optimizing control over the predicted robot state, however, the tracking of the desired impedance behavior remains reactive, thus missing the predictive capability of MPC, limiting their ability to control the interaction behavior and track the desired impedance behavior optimally.

A. Contributions

The contributions of this work are the following:

1) We propose a novel control method that combines Admittance Control with NMPC, such that NMPC can predict and optimize the control sequence over the predicted evolution of both the robot's state, and the desired impedance state. The key novelty is the inclusion of a dynamic model of the desired impedance within the NMPC prediction model, as shown in Fig. 1b. This enables the NMPC to jointly predict the evolution of both the robot state and the desired impedance state, and to compute control actions that minimize their deviation while satisfying system constraints.
2) We verify and evaluate the proposed method with real-time simulations and physical experiments to show the validity of the proposed method[1].
3) We compare the proposed method with the state-of-the-art controller [13] that uses a cascaded structure with two nested control loops. We provide empirical evidence that our method outperforms the state-of-the-art methods in tracking the desired impedance behavior and having smoother and safer physical interaction with unknown and unmodeled environments, while respecting the system's constraints and maintaining the system's stability.

[1]The supplementary video includes videos of the experiments and animations of the simulation results.

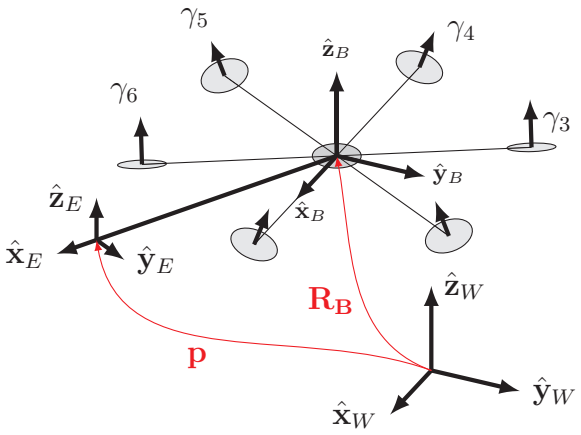

Fig. 2: Schematic representation of a fully-actuated multi-rotor aerial robot with its reference frames.

The rest of the paper is organized as follows: Section II introduces the prediction models for the robot and the desired impedance. Section III describes the proposed control method. Section IV presents the experimental setup, details the design of the experiments, and discusses the results of the simulations and physical experiments, which validate the proposed controller and compare its performance with the state-of-the-art controller. Finally, Section V concludes the paper.

II. Modelling

This section presents the prediction models for the NMPC controller. First, the reference frames are defined as shown in Fig. 2, where the inertial world frame is defined as $\mathcal{F}_W = O_W\{\hat{\mathbf{x}}_W, \hat{\mathbf{y}}_W, \hat{\mathbf{z}}_W\}$, while the frame $\mathcal{F}_B = O_B\{\hat{\mathbf{x}}_B, \hat{\mathbf{y}}_B, \hat{\mathbf{z}}_B\}$ is the body frame, which is attached to the geometric center of the Aerial Robot (AR) and aligned with the rigid body's principal inertia axes, i.e. the inertia matrix $\mathbf{J} \in \mathbb{R}^{3\times3}_{>0}$ is a diagonal matrix. The frame $\mathcal{F}_E = O_E\{\hat{\mathbf{x}}_E, \hat{\mathbf{y}}_E, \hat{\mathbf{z}}_E\}$ is rigidly attached to the end-effector. Finally, the frame $\mathcal{F}_{A_i} = O_{A_i}\{\hat{\mathbf{x}}_{A_i}, \hat{\mathbf{y}}_{A_i}, \hat{\mathbf{z}}_{A_i}\}$ is associated with the i-th actuator, such that $i \in \{1, \dots, n\}$, and n is the number of actuators. The origin O_{A_i} coincides with the thrust generation point, and $\hat{\mathbf{z}}_{A_i}$ is aligned with the thrust generation direction of the i-th actuator, which is assumed to be constant w.r.t. $\mathcal{F}_B$ in this paper.

A. Multi-rotor Aerial Robot Prediction Model

For aerial physical interaction tasks, it is convenient to model the translational dynamics of the end-effector instead of the AR Center of Mass (CoM), hence the translational dynamics will be derived to represent the acceleration of $\mathcal{F}_E$ w.r.t. $\mathcal{F}_W$. Therefore, $\mathbf{p}, \dot{\mathbf{p}}, \ddot{\mathbf{p}} \in \mathbb{R}^3$ will represent the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

position, velocity, and acceleration of $\mathcal{F}_E$ w.r.t. $\mathcal{F}_W$, expressed in $\mathcal{F}_W$. On the other hand, the orientation of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$ is denoted with $\mathbf{R}_B \in \mathrm{SO}(3)$, and with the unit quaternion $\mathbf{q}$, while $\boldsymbol{\omega} \in \mathbb{R}^3$ represents the angular velocity of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$ expressed in $\mathcal{F}_B$. Moreover, the angular acceleration is indicated with $\dot{\boldsymbol{\omega}} \in \mathbb{R}^3$. Finally, $\mathbf{o}_{B,E}$ denotes the position of O_E w.r.t. $\mathcal{F}_B$.

The state $\mathbf{x}$ of the AR model is defined as:

$$\mathbf{x} \coloneqq \begin{bmatrix} \mathbf{p}^\top & \dot{\mathbf{p}}^\top & \mathbf{q}^\top & \boldsymbol{\omega}^\top & \boldsymbol{\gamma}^\top & \mathbf{f}_\mathrm{c}^\top \end{bmatrix}^\top \tag{1}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^n$ are the magnitudes of the thrusts generated by each actuator along its $\hat{\mathbf{z}}_{A_i}$, and $\mathbf{f}_\mathrm{c} \in \mathbb{R}^3$ denotes the contact forces applied to the end-effector and expressed in $\mathcal{F}_W$.

For an AR with a mass $m \in \mathbb{R}_{>0}$, and an inertia matrix $\mathbf{J}$, the Newton–Euler equations of motion are:

$$\begin{aligned} \ddot{\mathbf{p}} &= -g\hat{\mathbf{z}}_W + m^{-1}\mathbf{R}_B\mathbf{G}_f\boldsymbol{\gamma} + m^{-1}\mathbf{f}_\mathrm{c} \\ &\quad + \mathbf{R}_B\left([\dot{\boldsymbol{\omega}}]_\times \mathbf{o}_{B,E} + [\boldsymbol{\omega}]_\times^2 \mathbf{o}_{B,E}\right) && \text{(2a)} \\ \dot{\boldsymbol{\omega}} &= \mathbf{J}^{-1}\left(-\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{G}_\tau\boldsymbol{\gamma} + [\mathbf{o}_{B,E}]_\times\left(\mathbf{R}_B^\top \mathbf{f}_\mathrm{c}\right)\right) && \text{(2b)} \\ \dot{\mathbf{q}} &= \frac{1}{2}\,\mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} && \text{(2c)} \end{aligned}$$

where g is the gravitational acceleration, $\mathbf{G}_f$ and $\mathbf{G}_\tau \in \mathbb{R}^{3\times n}$ are respectively the force and moment allocation matrices [17]. $[\bullet]_\times \in \mathfrak{so}(3)$ represents the skew symmetric matrix of the vector $\bullet \in \mathbb{R}^3$, and $\otimes$ represents the Hamilton quaternion multiplication.

The dynamics of the actuators' thrusts are defined as:

$$\dot{\boldsymbol{\gamma}} = \mathbf{u} \tag{3}$$

where $\mathbf{u} \in \mathbb{R}^n$ are the system control inputs, which are directly mapped to the torques applied by the DC brushless motors, see [12].

Then, the dynamics of the robot can be written as:

$$\dot{\mathbf{x}} = \boldsymbol{\phi}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{\mathbf{p}}^\top & \ddot{\mathbf{p}}^\top & \dot{\mathbf{q}}^\top & \dot{\boldsymbol{\omega}}^\top & \mathbf{u}^\top & \mathbf{0}_{3\times1}^\top \end{bmatrix}^\top \tag{4}$$

Note that the contact forces $\mathbf{f}_\mathrm{c}$ are part of the robot state, and hence, they have to be measured or estimated. We assume that the contact forces are constant in the prediction horizon, hence, the zero vector, $\mathbf{0}_{3\times1}$, in (4). This model was chosen because of its simplicity, and it does not require any a priori knowledge about the nature of the physical interaction, or the properties of the interaction environment. Of course, this assumption might not hold if the forces rate of change is high, but this model can be replaced with a different competent force model.

B. Impedance Model

This section introduces a model of a 3-Degree of Freedom (DoF) mechanical impedance comprising a 2nd-order mass damper spring system. In the admittance control approach, this is the system that the robot should imitate during the physical interaction.

The impedance state are $\boldsymbol{\chi} \coloneqq \begin{bmatrix} \boldsymbol{\delta}^\top & \dot{\boldsymbol{\delta}}^\top \end{bmatrix}^\top$, where $\boldsymbol{\delta}, \dot{\boldsymbol{\delta}}, \ddot{\boldsymbol{\delta}} \in \mathbb{R}^3$ are the position, velocity, and acceleration of the impedance, respectively.

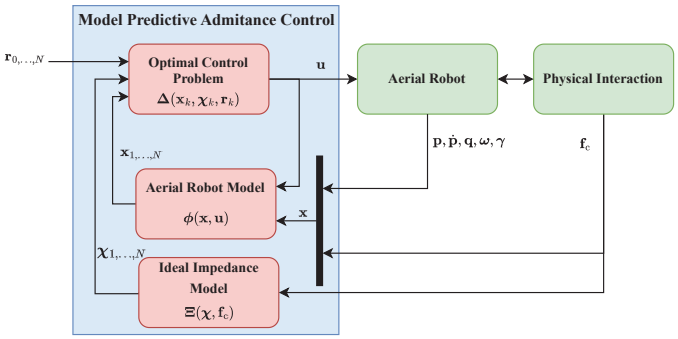

Fig. 3: Block diagram representing the proposed controller.

Then the impedance dynamics are:

$$\dot{\boldsymbol{\chi}} = \boldsymbol{\Xi}(\boldsymbol{\chi}, \mathbf{f}_\mathrm{c}) \begin{bmatrix} \dot{\boldsymbol{\delta}} \\ \mathbf{M}^{-1}(-\mathbf{D}\dot{\boldsymbol{\delta}} - \mathbf{K}\boldsymbol{\delta} + \mathbf{f}_\mathrm{c}) \end{bmatrix} \tag{5}$$

which is defined by $\mathbf{M},\ \mathbf{D},\ \mathbf{K} \in \mathbb{R}^{3\times3}_{>0}$, the apparent inertia, damping, and stiffness diagonal matrices, respectively, and $\mathbf{f}_\mathrm{c} \in \mathbb{R}^3$ is the external force applied to the impedance.

The impedance parameters, which are the apparent inertia, damping, and stiffness coefficients, can be chosen to satisfy some desired time response characteristics, such as rise time, and peak overshoot, or frequency domain characteristics such as natural frequency and damping ratio.

Note that the impedance model in (5) is defined only for the 3D translational dynamics, however, this can also be extended to a 6-DoF mechanical impedance to include the rotational dynamics as illustrated in [11].

III. Control Design

This section presents the proposed controller design, see Fig. 3. The controller is designed to be a standard trajectory tracking controller, similar to [12], in the absence of physical interaction. When physical interaction occurs, the controller's objective will be to track the dynamic behavior of a specified desired impedance.

This is similar to the controller proposed in [11] with two key differences: the control structure and the predictivity of the controller. First, the physical interaction control structure proposed in [11] is based on two nested control loops, where the inner-loop controller is a trajectory tracking controller, and the outer-loop controller is an admittance filter that modifies the trajectory reference when there is physical interaction. On the contrary, our control structure has a single control loop that can control both the trajectory tracking and the physical interaction control.

Second, the trajectory tracking controller in [11] is a reactive controller based on feedback linearization, while our proposed controller is an NMPC that can exploit the model of the system to optimize the control and respect the constraints of the system. More importantly for physical interaction control, our controller uses a model of the desired impedance to predict its future evolution and track its dynamic behavior. This leads to better tracking of the desired dynamic behavior and improved physical interaction characteristics, as will be shown in Section IV.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

The proposed NMPC controller uses the following discrete-time Optimal Control Problem (OCP), over the receding horizon t_h, discretized into N shooting points, at a given instant t:

$$\min_{\substack{\mathbf{x}_0 \dots \mathbf{x}_N \\ \mathbf{u}_0 \dots \mathbf{u}_{N-1}}} \sum_{k=0}^{N} \left\| \boldsymbol{\Delta}(\mathbf{x}_k, \boldsymbol{\chi}_k, \mathbf{r}_k) \right\|^2_{\mathbf{Q}_k} \tag{6a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \boldsymbol{\phi}(\mathbf{x}_k, \mathbf{u}_k), \quad k \in \{0, N-1\} \tag{6b}$$

$$\boldsymbol{\chi}_{k+1} = \boldsymbol{\Xi}(\boldsymbol{\chi}_k, \mathbf{x}_k), \quad k \in \{0, N-1\} \tag{6c}$$

$$\mathbf{x}_0 = \hat{\mathbf{x}}(t) \tag{6d}$$

$$\boldsymbol{\chi}_0 = \boldsymbol{\chi}_{t-1,1} \tag{6e}$$

$$\underline{\boldsymbol{\gamma}} \leq \boldsymbol{\gamma}_k \leq \overline{\boldsymbol{\gamma}}, \quad k \in \{0, N\} \tag{6f}$$

$$\underline{\dot{\boldsymbol{\gamma}}}_k \leq \mathbf{u}_k \leq \overline{\dot{\boldsymbol{\gamma}}}_k, \quad k \in \{0, N-1\} \tag{6g}$$

where the subscript k denotes the time step kT, such that $\mathbf{x}_k = \mathbf{x}(kT)$, and T is the sampling time.

The function $\boldsymbol{\Delta} \in \mathbb{R}^{n_y}$ is the objective function that describes the controller tasks, $\mathbf{Q}_k \in \mathbb{R}^{n_y \times n_y}_{>0}$ is the weight matrix, and $n_y \in \mathbb{N}_{>0}$ is the number of outputs in $\boldsymbol{\Delta}$.

The motion reference trajectory at time step k is defined as:

$$\mathbf{r}_k = \begin{bmatrix} \mathbf{p}^\top_{r,k} & \dot{\mathbf{p}}^\top_{r,k} & \ddot{\mathbf{p}}^\top_{r,k} & \mathbf{q}^\top_{r,k} & \boldsymbol{\omega}^\top_{r,k} & \dot{\boldsymbol{\omega}}^\top_{r,k} \end{bmatrix} \in \mathbb{R}^{19} \tag{7}$$

Additionally, $\boldsymbol{\phi}(\mathbf{x}_k, \mathbf{u}_k) \in \mathbb{R}^{n_x}$ and $\boldsymbol{\Xi}(\boldsymbol{\chi}_k, \mathbf{x}_k)$ are numerical integration operators that solve initial value problems that represent the nonlinear discrete prediction model of the AR from (4), and the desired impedance model from (5), respectively. Note that this OCP uses the impedance model as a prediction model together with the robot's model.

The initial conditions of the robot state $\mathbf{x}_0$ are defined using the estimated robot's state $\hat{\mathbf{x}}(t)$, while the initial condition for the impedance state $\boldsymbol{\chi}_0$ are defined using the predicted evolution of the impedance state from the last time step, where $\boldsymbol{\chi}_{t-1,1}$ refers to the second prediction ($k = 1$), from the previous time step ($t - 1$).

Finally, the actuator thrusts lower and upper limits are $\underline{\boldsymbol{\gamma}}, \overline{\boldsymbol{\gamma}} \in \mathbb{R}^n$, respectively, while $\underline{\dot{\boldsymbol{\gamma}}}, \overline{\dot{\boldsymbol{\gamma}}} \in \mathbb{R}^n$ represent the lower and upper limits of the control inputs. These constraints represent realistic physical limitations of the actuators, and they are directly related to the velocity and acceleration constraints of the DC brushless motors, which are experimentally identified as in [12].

The objective function, which describes the cost that should be minimized by the controller, includes the trajectory tracking errors in the translational and rotational trajectories. The translational error also includes the tracking error of the impedance motion in the presence of physical interaction. This objective function is defined as:

$$\boldsymbol{\Delta}(\mathbf{x}_k, \boldsymbol{\chi}_k, \mathbf{r}_k) = \begin{bmatrix} \mathbf{p}_k - \mathbf{p}_{r,k} - \boldsymbol{\delta}_k \\ \dot{\mathbf{p}}_k - \dot{\mathbf{p}}_{r,k} - \dot{\boldsymbol{\delta}}_k \\ \ddot{\mathbf{p}}_k - \ddot{\mathbf{p}}_{r,k} - \ddot{\boldsymbol{\delta}}_k \\ \mathbf{q}_k \ominus \mathbf{q}_{r,k} \\ \boldsymbol{\omega}_k - \boldsymbol{\omega}_{r,k} \\ \dot{\boldsymbol{\omega}}_k - \dot{\boldsymbol{\omega}}_{r,k} \end{bmatrix} \in \mathbb{R}^{18} \tag{8}$$

where $\mathbf{q}_k \ominus \mathbf{q}_{r,k} \in \mathbb{R}^3$ denotes the quaternion error as defined in [18], for example. This objective function does not include an explicit penalty on the control input, thanks to the hard constraints on the control input and its integral as part of the state constraints, which will limit the control effort, as was shown in [12].

In the absence of physical interaction[2]: $\boldsymbol{\delta}_k = \dot{\boldsymbol{\delta}}_k = \ddot{\boldsymbol{\delta}}_k = \mathbf{0}_{3\times1}$, which means that the controller acts as a trajectory tracker. On the other hand, when there is physical interaction, that is $\mathbf{f}_c \neq \mathbf{0}_{3\times1}$, the impedance state changes, as a response to the external forces, leading to non-zero $\boldsymbol{\delta}_k, \dot{\boldsymbol{\delta}}_k, \ddot{\boldsymbol{\delta}}_k$, which leads to a deviation from the reference trajectory that is equivalent to the desired impedance motion.

IV. Validation and Evaluation

A. Experimental Setup

A fully-actuated hexarotor aerial robot is used in this experimental campaign. This robot is developed in-house, and it uses off-the-shelf or 3-D printed components. The physical properties of the robot are listed in Table I. The robot is equipped with a fixed tool for physical interaction purposes. This tool is rigidly attached to the robot's mechanical frame and has a 6D force/torque sensor, FT45 developed by the Italian Institute of Technology (IIT). A Butterworth second-order filter is applied to the sensor's measurements with a cutoff frequency of 10 Hz, and then a deadband of 0.2 N is applied to the filtered measurements. The tuning of the filters was based on a heuristic evaluation of the sensor's noise characteristics.

The rotors of the AR are tilted about the arm axis (the arm connecting the rotor with the center of the robot body) by $\alpha = 20°$. Every second rotor is tilted with a negative α angle. The rotors comprise 13-inch propellers and MikroKopter's MK3638 brushless motors driven by MikroKopter's Electronic Speed Controllers (ESCs) BL-Ctrl-2.0. The ESCs control the rotational velocities of the motors using the controller presented in [19]. The physical limits of the rotors' rotational velocities and rotational accelerations are identified empirically using a thrust stand setup, explained in detail in [12]. The robot is powered by a 3200 mAh 4S LiPo battery.

The control framework is implemented using MATLAB-Simulink and runs at 250 Hz on a stationary base PC connected to the robot through a serial cable. The PC is equipped with a 12th Gen Intel® Core™ i7-12800HX×24 and 32 GB RAM running the operating system of Linux Ubuntu 20.04 LTS. The robot state is estimated by an Unscented Kalman Filter (UKF) state estimator, providing full state estimation at 250 Hz. The UKF fuses the onboard measurements from the Inertial Measurement Unit (IMU) (which includes gyroscopes and accelerometers), provided at 500 Hz, with the position and orientation measurements from an external motion capture system (Optitrack MoCap) provided at 150 Hz.

MATMPC [20], a MATLAB-based NMPC toolbox, is used to implement the proposed NMPC controller. Additionally, qpOASES [21], which is an open-source Quadratic Programming (QP) solver, is used to solve the QP problem with the Real-Time Iteration scheme [22]. Also, a fixed step 4^{th} order

[2]Assuming that the initial condition of these states at $t = 0$ is zero.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

explicit Runge-Kutta integrator is used to simulate the prediction models using a multiple shooting scheme [23], while CasADi [24], is used for symbolic algorithmic differentiation. We set the prediction horizon to $t_h = 1.5$s, discretized into $N = 15$ steps, yielding a discretization step of 0.1s.

B. Experimental Design

We validate the proposed method in simulation and physical experiments in different testing scenarios that emulate different physical interaction tasks:

1) The first scenario is a Physical Interaction during Free-Flight, where the AR is hovering, and a physical interaction force is applied to the robot, along $\hat{\mathbf{x}}_W$. In this scenario, the robot is free to move and respond to the physical interaction. This scenario emulates multiple possible tasks, such as human-robot physical interaction. The desired behavior in this scenario is that the robot will respond to the physical interaction force by a motion profile that is identical to the motion of the desired impedance when the same interaction force is applied to it. Therefore, in this scenario, we observe the error between the position and velocity of the robot $(\mathbf{p}_x, \dot{\mathbf{p}}_x)$ and the desired impedance $(\boldsymbol{\delta}_x, \dot{\boldsymbol{\delta}}_x)$ along $\hat{\mathbf{x}}_W$, such that:

$$e_p = \mathbf{p}_x - \boldsymbol{\delta}_x \tag{9}$$

$$e_v = \dot{\mathbf{p}}_x - \dot{\boldsymbol{\delta}}_x \tag{10}$$

2) The second scenario is Physical Interaction with a Wall, where the AR reference trajectory leads the robot to push against the rigid body. In this scenario, the robot's motion is constrained along the normal axis of the rigid body surface. This scenario emulates tasks such as the inspection and maintenance of infrastructures, with uncertainty about the position of the infrastructure.
Hence, in this scenario, we observe the stability and compliance of the contact together with the interaction forces that are induced by changing the position reference deeper into the surface of the rigid body.

Our method is compared with the state-of-the-art controller that uses NMPC for trajectory tracking while a (reactive) admittance filter modifies the reference trajectory to respond to interaction forces (as proposed in [13] and similar to other admittance-filter methods in the literature [8], [9], [10], [11], [25]), we denote this method as *SoTA*. The weights of the two controllers are identical, and they are presented in Table II, where $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is the identity matrix. These weights reflect the desire to prioritize the stability of the rotational dynamics over the translational trajectory tracking, which is necessary for the stability of the robot.

TABLE I: Physical parameters of the aerial robot

Parameter	Value
Mass	2.8 Kg
Inertia $[I_{xx}, I_{yy}, I_{zz}]$	$[0.115, 0.114, 0.194]$ Kg m^2
Thrust coefficient c_t	11.75×10^{-4} N/Hz^2
Torque drag coefficient c_d	2.388×10^{-5} Nm/Hz^2
Rotors tilt angle α	$\pm 20°$
Rotor to rotor distance	0.7 m

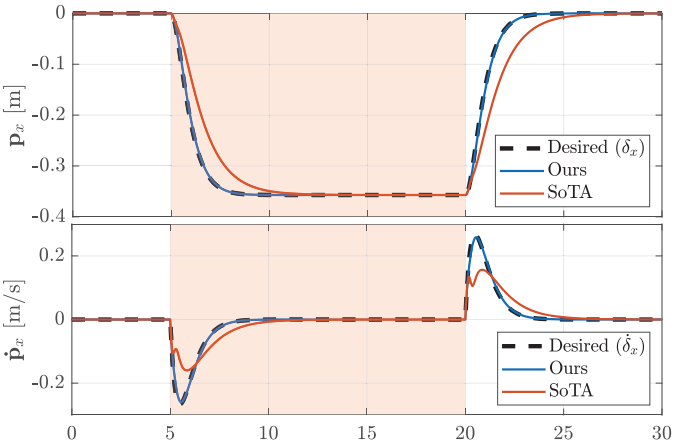

Fig. 4: Simulation of Physical Interaction during Free-Flight with a $4N$ pulse force applied along $-\hat{x}_W$ during the highlighted period while the AR hovers. The plots show the position and velocity tracking.

We chose to test the first scenario in real-time simulation because we can simulate the external forces as pulse signals, allowing for clearer analysis. Additionally, the dynamics of quasi-static hovering are well understood, and their models are close to reality. On the other hand, simulating rigid contact is more challenging [26], hence the second scenario, which involves physical interaction with a wall, is tested with physical experiments with the real robot.

C. Simulation Results

In this simulation, a 4 N external pulse force is applied along $-\hat{x}_W$ while the robot is hovering. The desired impedance parameters are chosen such that the system is critically-damped and has an apparent inertia equal to the robot's mass, namely: $\mathbf{M} = 2.8\,\mathbf{I}_3$, $\mathbf{D} = 11.2\,\mathbf{I}_3$, $\mathbf{K} = 11.2\,\mathbf{I}_3$. As shown in Fig. 4, when the pulse force is applied at 5 s, the desired impedance is displaced along $-\hat{x}_W$, and our proposed controller accurately tracks the desired impedance behavior, following the ideal impedance position and velocity during both the transient and steady-state phases. Similarly, when the external force is removed at 20 s, the robot's position and velocity match the desired impedance ideal position and velocity.

In comparison, the SoTA controller failed to track desired impedance dynamics during the transient phase. Position tracking exhibited a noticeable lag relative to the ideal reference, while velocity tracking was poor and did not match the velocity profile of the desired impedance.

This simulation is repeated with different desired impedance parameters, where the desired impedance parameters are var-

TABLE II: Weights of the objective function for NMPC controllers during the simulations and physical experiments.

Weight	$\mathbf{Q}_\mathbf{p}$	$\mathbf{Q}_{\dot{\mathbf{p}}}$	$\mathbf{Q}_{\tilde{\mathbf{p}}}$	$\mathbf{Q}_\mathbf{q}$	$\mathbf{Q}_{\boldsymbol{\omega}}$	$\mathbf{Q}_{\dot{\boldsymbol{\omega}}}$
Ours	$50\,\mathbf{I}_3$	$50\,\mathbf{I}_3$	$0\,\mathbf{I}_3$	$5 \times 10^5\,\mathbf{I}_3$	$1\,\mathbf{I}_3$	$0\,\mathbf{I}_3$
SoTA	$50\,\mathbf{I}_3$	$50\,\mathbf{I}_3$	$0\,\mathbf{I}_3$	$5 \times 10^5\,\mathbf{I}_3$	$1\,\mathbf{I}_3$	$0\,\mathbf{I}_3$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

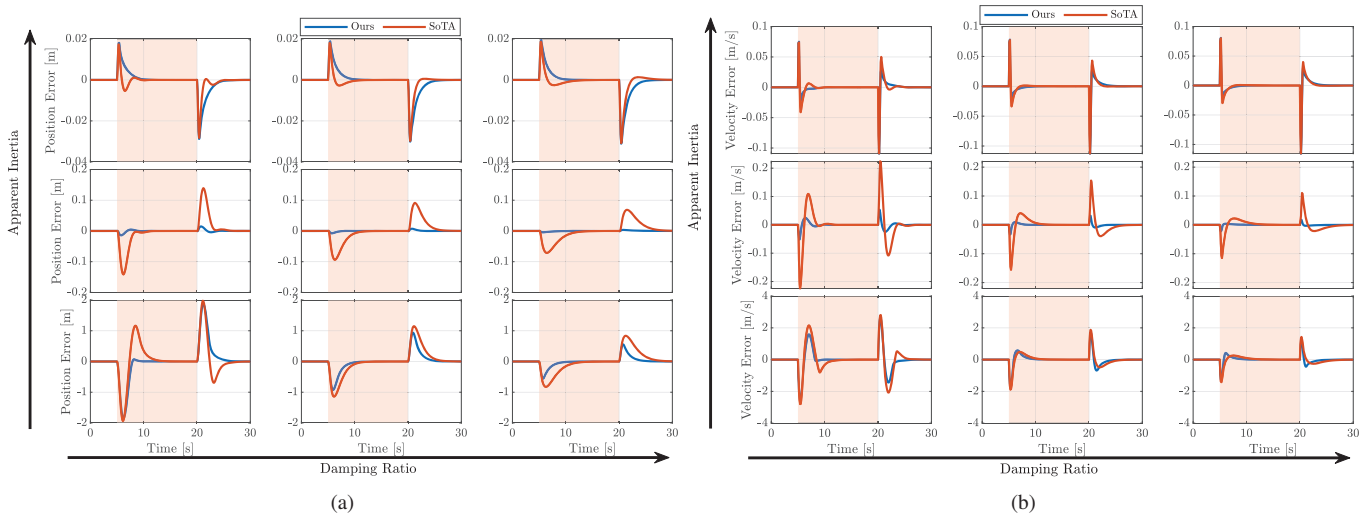

Fig. 5: Physical Interaction during Free-Flight simulations. A $4N$ pulse force is applied along $-\hat{x}_W$ during the highlighted period. The left, middle, and right columns correspond to desired impedance damping ratios of 0.5, 1, and 1.5, respectively. The bottom, middle, and top rows correspond to apparent inertias of 0.1, 1, and 10 times the robot's mass, respectively. The plots show (a) position error e_p and (b) velocity error e_v between the desired impedance and the robot position and velocity along the x-axis.

ied to cover scenarios where higher, equal, or lower apparent inertia, $\bar{m}$, compared to the robot mass, is desired. Also, we tested desired impedance parameters with different damping ratios ζ: under-damped, critically-damped, and over-damped systems. The natural frequency of the desired impedance was set $\omega_n = 2\,\text{rad/s}$.

Given the desired apparent inertia $\bar{m}$, the natural frequency ω_n, and the damping ratio ζ, we can calculate the desired stiffness k and damping coefficients d as:

$$k = \bar{m}\,\omega_n^2 \tag{11a}$$

$$d = 2\,\zeta\sqrt{k\,\bar{m}} \tag{11b}$$

As shown in Fig. 5, our method has smaller position and velocity errors than SoTA in most of the study cases. In the case of high desired apparent inertia, the performance of the two controllers is similar, with slightly smaller errors for SoTA in the transient phase. When the desired apparent inertia is equal to the robot's mass, the proposed method has significantly smaller errors than the SoTA, both in velocity and position errors. More specifically, the position error was 90% smaller in the under-damped case. Finally, when the desired apparent inertia is smaller than the robot's mass, our method exhibits smaller position and velocity errors.

Investigating the advantages of predicting the future impedance states: To further examine the benefits of predicting the future evolution of the desired impedance state, Fig. 6a shows the control input (of the first three rotors for brevity, the behavior of the other three motors is almost identical) in the case of a low apparent inertia and over-damped system. In response to the negative external force along $\hat{\mathbf{x}}_W$, our method increases the thrust of rotors 1 and 3, and decreases that of rotor 2. This generates a negative total force along the $\hat{\mathbf{x}}_W$ direction, denoted as $\mathbf{f}_{\boldsymbol{\gamma},x}$, as shown in Fig. 6b. Although this response appears counterintuitive (further accelerating in the negative x-direction, rather than counteracting the external force, as SoTA does), it aligns with the desired low apparent inertia behavior. Specifically, the system is expected to exhibit

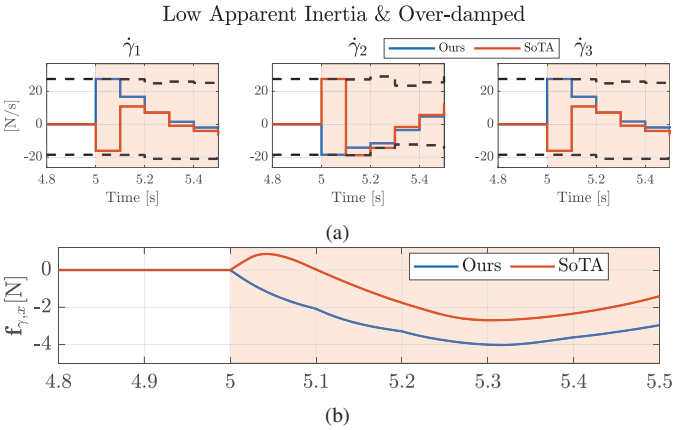

Fig. 6: Physical Interaction during Free-Flight simulation with low apparent inertia and over-damped desired impedance: (a) control inputs for the first three rotors; (b) generated actuator forces along $\hat{x}_W$.

larger accelerations under external disturbances, therefore, our controller uses the actuator forces to increase the acceleration along the force direction, resulting in smaller position and velocity tracking errors, as shown in Fig. 5 and the supplementary video.

D. Physical Experiments Results

In this experiment, the position reference leads the AR to push against a wall by asking the robot to move inside the wall, as shown in Fig. 7. Between 1 s and 2 s, the position reference is increased, leading the robot to physically interact with the wall, as shown in Fig. 8. This results in an increase in the normal force, which triggers the controller to modify the position reference to match the dynamic behavior of the desired impedance, as shown in the modified reference plot in Fig. 8, where the reference is decreasing smoothly between 1.5 s and 5 s. This smooth modification of the position reference resulted in a stable and smooth normal force in the same period between 1 s and 12 s. Then, at 12 s, the position

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

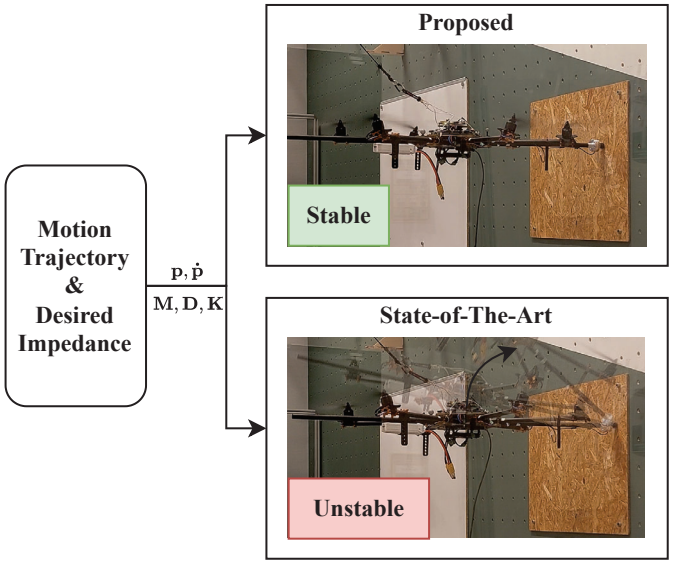

Fig. 7: Snapshots of the fully-actuated hexarotor aerial robot while interacting with a wall, comparing the proposed method with the state-of-the-art method [13].

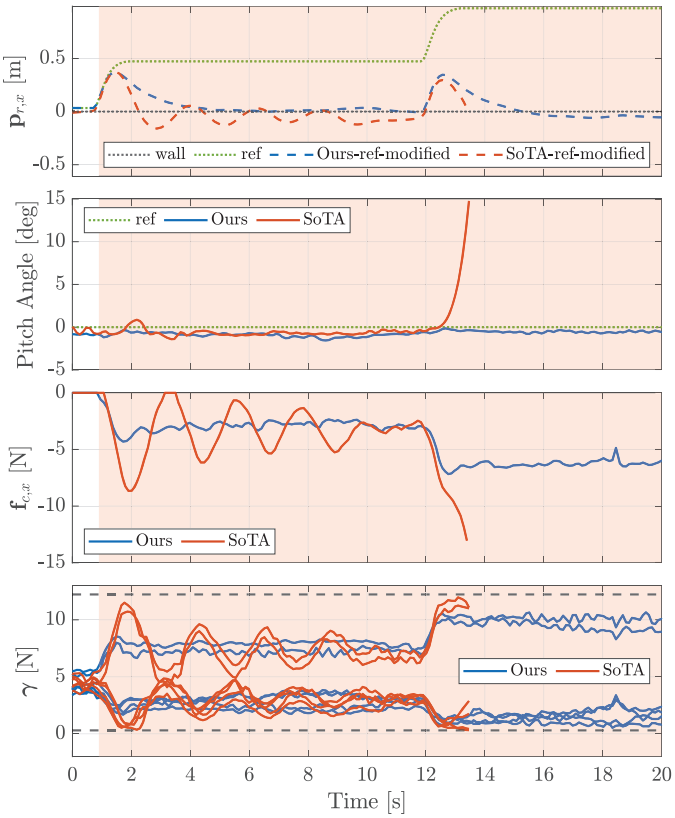

Fig. 8: Plots of position reference ($\mathbf{p}_r$) and modified references ($\mathbf{p}_r + \boldsymbol{\delta}$) along $\hat{\mathbf{x}}_W$, pitch angle tracking, contact normal force, and rotor thrusts (top to bottom) from an experiment of physical interaction with a wall. The highlighted interval indicates the in-contact phase. The position is omitted in the top plot as it remains fixed during the interaction with the static wall.

reference is moved further into the wall, to test the robustness of the proposed method. This led to a higher normal force. However, our controller was able to stabilize the interaction by smoothly modifying the position reference, leading to almost constant normal force between 13 s and 20 s.

On the contrary, the response of the state-of-the-art controller, SoTA, is oscillatory in terms of the resulting contact forces and the commanded rotor thrusts, as shown in Fig. 8. The observed behavior results from the SoTA controller attempting to minimize the position error by driving the robot into the wall, leading to high contact normal force at 2 s. This force induced substantial reference modification by the admittance filter, displacing the position reference beyond the wall at 3 s. Consequently, contact was lost (zero normal force) around 3.5 s. Repeating this cycle leads to oscillations between 1 s and 10 s. Then, at 12 s, when the position reference is moved further inside the wall, the system that is controlled by SoTA becomes unstable and crashes into the wall, as shown in Fig. 7 and the supplementary video.

This happened because the SoTA controller saturated the thrust at 13 s, leading to a big deviation in the pitch angle, as shown in Fig. 8. When the controller tried to correct the error, it generated a higher normal contact force, which further contributed to a larger pitch angle. From Fig. 9, we can deduce that the SoTA has prioritized the correction of the pitch angle over the position tracking because it was predicting that the position will be driven back out of the wall. This result has been replicated multiple times. By reducing the weight of attitude tracking from 5×10^5 to 5×10^2, tolerating higher attitude errors, we were able to make the interaction stable with the SoTA, but the oscillatory behavior remained the same.

This significant difference in the behavior between our controller and the SoTA controller is mainly because our controller has a prediction of how the modified position reference will evolve in the future, based on the current contact forces. This is clear in Fig. 9, where the predicted modified position reference of the two controllers is shown, together with the predicted evolution of the position. From the start of the physical interaction between 1 s and 1.5 s, our controller was predicting that the position reference would move back towards the wall, while SoTA had a modified reference that is constant within the prediction horizon, because of the admittance filter reactivity. This is also reflected in the predictions of the position, where SoTA was predicting to go deeper into the wall between 1 s and 1.5 s, and then out of the wall between 1.5 s and 2 s, while our controller predictions were less oscillatory around the contact point, leading to almost no oscillations.

It is worth noting that the average computation time of our proposed method is 2.19 ms, while this was 2.13 ms for the SoTA method, indicating that our method does not add any significant computational load.

E. Summary

To summarize the results of the simulations and physical experiments, we have shown in the simulation results the efficacy of our proposed method in tracking the desired impedance behavior, which outperforms the SoTA method with smaller position and velocity errors in most of the different desired impedances that we tested. The improved admittance control allows the AR to interact with a wall smoothly and safely, maintaining stability when operating at the actuator limits, as we have shown in the physical experiments results. While

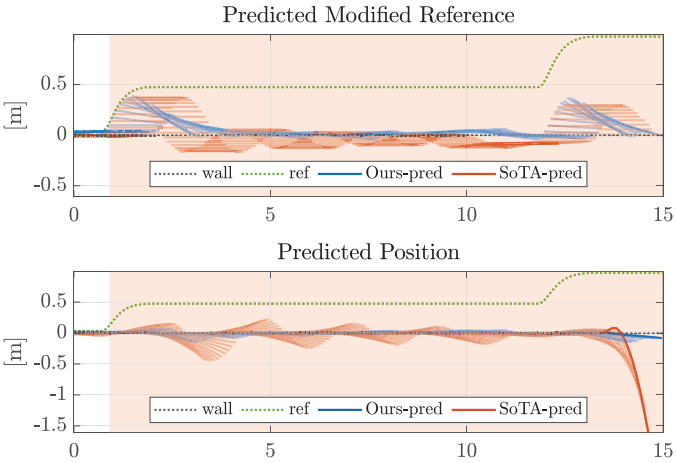

Fig. 9: Plots of the predicted modified reference and predicted position along the $\hat{\mathbf{x}}_W$ for the experiment of pushing against a wall. Each solid line in these plots represents the NMPC prediction of the modified position reference ($\mathbf{p}_r + \boldsymbol{\delta}$) or position ($\mathbf{p}$) from time t to $t + t_h$ (prediction horizon).

the SoTA suffered from high oscillations in the contact force, leading to loss of contact, and eventually leading to unstable behavior when reaching the actuator limits. We have also shown that this improved performance is mainly due to the ability of our method to jointly predict the future evolution of the desired impedance behavior, together with the robot state.

V. Conclusions

This paper presented a novel admittance control framework based on NMPC for aerial robots physically interacting with the environment. The approach incorporates a model of the desired impedance dynamics within the NMPC predictive model and formulates a receding-horizon optimal control problem to minimize the tracking error between the robot state and the desired impedance state. Through real-time simulations and physical experiments of different physical interaction tasks, we demonstrated that the proposed method achieves better tracking of desired impedance behavior compared to the state-of-the-art controllers. Notably, our experiments show that the improved tracking performance can determine the difference between stable and unstable behavior during sustained physical interaction with rigid environments.

References

[1] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 626–645, 2022.

[2] L. Villani and J. De Schutter, *Force Control*, pp. 195–220. Springer Handbook of Robotics, 2016.

[3] H.-N. Nguyen and D. Lee, "Hybrid force/motion control and internal dynamics of quadrotors for tool operation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3458–3464, IEEE, 2013.

[4] S. Park, J. Lee, J. Ahn, M. Kim, J. Her, G.-H. Yang, and D. Lee, "Odar: Aerial manipulation platform enabling omnidirectional wrench generation," *IEEE/ASME Transactions on mechatronics*, vol. 23, no. 4, pp. 1907–1918, 2018.

[5] X. Meng, Y. He, and J. Han, "Hybrid force/motion control and implementation of an aerial manipulator towards sustained contact operations," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3678–3683, 2019.

[6] T. Wang, K. Umemoto, T. Endo, and F. Matsuno, "Dynamic hybrid position/force control for the quadrotor with a multi-degree-of-freedom manipulator," *Artificial Life and Robotics*, vol. 24, pp. 378–389, 2019.

[7] N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, 1985.

[8] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Variable impedance control for aerial interaction," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3435–3440, IEEE, 2014.

[9] T. Tomić and S. Haddadin, "A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4197–4204, IEEE, 2014.

[10] E. Cataldi, G. Muscio, M. A. Trujillo, Y. Rodríguez, 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)*, pp. 3848–3853, IEEE, 2016.

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

[12] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1213–1247, 2020.

[13] A. Alharbat, H. Esmaeeli, D. Bicego, A. Mersha, and A. Franchi, "Three fundamental paradigms for aerial physical interaction using nonlinear model predictive control," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 39–48, IEEE, 2022.

[14] M. Bednarczyk, H. Omran, and B. Bayle, "Model predictive impedance control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4702–4708, IEEE, 2020.

[15] R. Jiao, J. Li, Y. Rong, and T. Hou, "Nonlinear model predictive impedance control of a fully actuated hexarotor for physical interaction," *Sensors*, vol. 23, no. 11, p. 5231, 2023.

[16] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto, "Active interaction force control for contact-based inspection with a fully actuated aerial vehicle," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 709–722, 2020.

[17] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, "Design of multirotor aerial vehicles: A taxonomy based on input allocation," *The International Journal of Robotics Research*, p. 02783649211025998, 2021.

[18] L. Peric, M. Brunner, K. Bodie, M. Tognon, and R. Siegwart, "Direct force and pose NMPC with multiple interaction modes for aerial push-and-slide operations," in *International Conference on Robotics and Automation (ICRA 2021)*, 2021.

[19] A. Franchi and A. Mallet, "Adaptive closed-loop speed control of bldc motors with applications to multi-rotor aerial vehicles," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5203–5208, IEEE, 2017.

[20] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, "MATMPC - a Matlab based toolbox for real-time nonlinear model predictive control," in *2019 18th European Control Conference (ECC)*, pp. 3365–3370, IEEE, 2019.

[21] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[22] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002.

[23] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[24] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[25] A. Afifi, M. van Holland, and A. Franchi, "Toward physical human-robot interaction control with aerial manipulators: Compliance, redundancy resolution, and input limits," in *2022 International conference on robotics and automation (ICRA)*, pp. 4855–4861, IEEE, 2022.

[26] Q. Le Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier, "Contact models in robotics: a comparative analysis," *IEEE Transactions on Robotics*, 2024.