2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561990

Direct Force and Pose NMPC with Multiple Interaction Modes for Aerial Push-and-Slide Operations

Lazar Peric*, Maximilian Brunner*, Karen Bodie*, Marco Tognon, Roland Siegwart

***Abstract*— In this paper, we present a model predictive controller for a fully actuated aerial manipulator to track a hybrid force and pose trajectory at the end-effector in an aerial interaction task. A force sensor at the end-effector is used to detect contact and to directly control the interaction force. We propose an approach for automatic transition between three operation modes which reflect the state of contact constraints, including free flight and two modes for force control based on static or dynamic friction at the end-effector. This division into three modes allows for different mode-specific controller tunings to optimize the desired performance throughout an interaction task. Results from flight experiments which combine force, position, and attitude tracking, show the performance of the controller in terms of accuracy and precision. The performance is further benchmarked against a hybrid force/impedance controller.**

I. INTRODUCTION

Increasing demand for industrial inspection has played a major role in the push for improved aerial manipulators (AMs), where dangerous tasks must be executed sparsely in extremely large workspaces. Far from a fixed grounding point, tasks such as force-controlled contact and push-and-slide on a surface are perfectly suited for micro aerial vehicles (MAVs) [1]. Incremental developments in aerial interaction over the past decades took a marked leap in the past few years with the emergence of fully-actuated AMs [2]. Their ability to decouple translational and rotational dynamics makes fully-actuated AMs capable of performing aerial interaction tasks with improved precision and stability [3].

Several demonstrations of push-and-slide tasks with fully-actuated AMs have emerged in the past few years. Applications include aerial writing [4], [5], peg-in-hole location [4], and contact-based infrastructure inspection [6], [5]. While most of these systems demonstrate the ability to exert high ($> 10\,\mathrm{N}$) static interaction forces, these forces are significantly lower during push-and-slide tasks, and typically with low-friction contact conditions.

Fixed base manipulators can overcome high friction forces via strong counteracting ground forces, but floating base manipulators (as depicted in Fig. 1) encounter a larger problem during the stick-slip transition. Hybrid force/pose trajectories typically separate the task space into constrained and unconstrained degrees of freedom (DoF), commanding forces in the constrained directions, and motion in the unconstrained directions. With high friction interaction, however, the transition between static and dynamic contact becomes nontrivial. Larger static friction forces, if not properly addressed, can cause saturation of the inputs, overshoots, and unexpected slips leading to possible crashes.

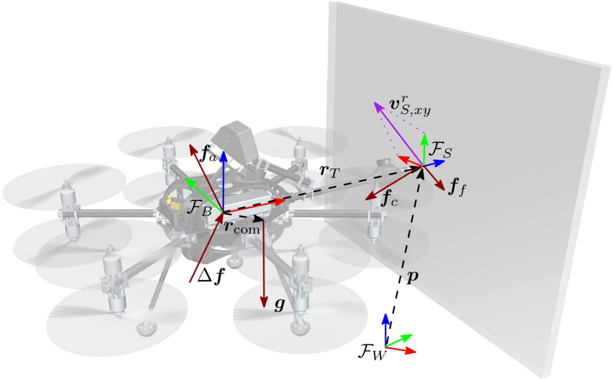

Fig. 1. Visual representation of the aerial manipulation system during a push-and-slide task, and its main variables.

Passivity-based techniques have been implemented for fully actuated AMs, such as the energy tank approach, which aim to handle the problem of unwanted slippage or loss of contact. This is achieved by virtually dissipating an energy tank in the controller, and restricting the system output accordingly [7]. However, forces during push-and-slide tasks remain small, and there is no use of direct force sensing, nor anticipation of the future trajectory.

The knowledge of interaction forces at the end-effector (EE) can provide key insight into the consideration of friction-based contact constraints. The estimation of EE forces for AMs, however, is typically degraded by unmodeled aerodynamic effects and delays due to system integration. With recent improvements of force-sensing technology, these concerns are addressed by directly measuring interaction forces with a Force-Torque (FT) sensor at the EE [8], [9].

Up to this point, existing control techniques for aerial interaction have been used for slow-moving, and predominantly low-friction sliding interaction tasks. Model predictive control (MPC), with its ability to account for constraints and to optimize over the future trajectory with a receding horizon, offers a solution to this problem. MPC has been previously implemented with underactuated-base AMs with fixed [10] and actuated [9] EE. However, due to underactuation of the platform, tasks remain slow, with low interaction forces during push-and-slide. Some MPC-based approaches have also been developed for fully-actuated MAVs [11], [12], but only in free-flight. In the domain of mobile robot manipulators,

* Authors contributed equally to this work and can all be considered as first authors.

This work was supported by the NCCR Digital Fabrication.

Authors are with the Autonomous Systems Lab, ETH Zürich, Leonhardstrasse 21, 8092 Zurich, Switzerland. e-mail: maximilian.brunner@mavt.ethz.ch

direct interaction force control has been implemented with MPC, achieving large forces applied to the environment [13]. Thanks to the fixed nature of the base, high friction forces could be treated as disturbances without causing instability.

The challenge remains in how to formulate the aerial interaction problem with a model-based controller in the presence of unknown environmental parameters, appropriately integrating information from direct force sensing. The method proposed in this paper aims to extend aerial interaction to high friction interaction up to high velocities. In particular, we present the following contributions:

- A novel nonlinear model predictive controller (NMPC) for hybrid 6-DoF pose and force tracking for a fully-actuated AM, incorporating direct force feedback for interaction.
- A practical implementation of contact constraint conditions for high friction aerial interaction, with autonomous transitions between 3 different control modes.
- Experimental validation of the controller and comparison against an impedance controller with a PI force feedback.

II. SYSTEM MODELING

In the course of this work we will refer to three reference frames as depicted in Fig. 1: the inertial world frame $\mathcal{F}_W = \{O_W, \boldsymbol{x}_W, \boldsymbol{y}_W, \boldsymbol{z}_W\}$, the body frame $\mathcal{F}_B = \{O_B, \boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}$ which is fixed to the geometric center of the AM, and the surface frame $\mathcal{F}_S = \{O_S, \boldsymbol{x}_S, \boldsymbol{y}_S, \boldsymbol{z}_S\}$, which is fixed to the surface (assumed flat in the work-space) and arbitrarily placed. $O_\star$ represents the center of the generic frame $\mathcal{F}_\star$, while $(\boldsymbol{x}_\star, \boldsymbol{y}_\star, \boldsymbol{z}_\star)$ represent its unit axes. Notice that $\mathcal{F}_S$ is defined s.t. $\boldsymbol{z}_S$ is orthogonal to and oriented into the surface. Additionally, $\mathcal{F}_W$ is defined s.t. $\boldsymbol{z}_W$ is aligned with the gravity vector $\boldsymbol{g} \in \mathbb{R}^3$.

The position of the EE and of the body center of mass (CoM) w.r.t. $\mathcal{F}_B$ are represented by the vectors $\boldsymbol{r}_T \in \mathbb{R}^3$ and $\boldsymbol{r}_{\text{com}} \in \mathbb{R}^3$, respectively. We assume that the AM and its EE are parts of a single rigid body, i.e., $\boldsymbol{r}_T$ and $\boldsymbol{r}_{\text{com}}$ are constant.

We introduce the flight control factor $\lambda \in \{0, 1\}$, where the system can be i) in free flight and $\lambda = 0$, or ii) in contact with a surface and $\lambda = 1$. In free flight the system's motion is unconstrained, while in contact, the EE can move only on a plane defined by $\{\boldsymbol{x}_S, \boldsymbol{y}_S\}$. To define interaction constraints according to the flight mode expressed by λ, we use the following selection matrix that will later be employed in the derivation of the dynamics:

$$\boldsymbol{S}_\lambda = \boldsymbol{R}_S \text{diag}\left(1, 1, 1 - \lambda\right) \boldsymbol{R}_S^\top, \tag{1}$$

where $\boldsymbol{R}_S \in SO(3)$ represents the attitude of $\mathcal{F}_S$ w.r.t. $\mathcal{F}_W$.

The system mass and inertia w.r.t. $\mathcal{F}_B$ are denoted by $m \in \mathbb{R}_{>0}$ and $\boldsymbol{J} \in \mathbb{R}^{3\times3}_{>0}$. Forces and torques acting on the system include i) those produced by the actuators of the aerial platform, $\boldsymbol{f}_a \in \mathbb{R}^3$ and $\boldsymbol{\tau}_a \in \mathbb{R}^3$, expressed in $\mathcal{F}_B$, ii) the contact force $\boldsymbol{f}_c \in \mathbb{R}^3$, expressed in $\mathcal{F}_W$, iii) the gravitational force $m\boldsymbol{g}$, expressed in $\mathcal{F}_W$, iv) and general disturbance forces and torques, $\Delta\boldsymbol{f} \in \mathbb{R}^3$ and $\Delta\boldsymbol{\tau} \in \mathbb{R}^3$, expressed in $\mathcal{F}_W$ and $\mathcal{F}_B$, respectively.

Since we are interested in controlling the pose of the EE, we consider as configuration of the system the position and orientation of the EE w.r.t. $\mathcal{F}_W$ described by the vector $\boldsymbol{p} \in \mathbb{R}^3$ and the quaternion $\boldsymbol{q} \in \mathbb{H}$, respectively. Their time derivatives, i.e., the linear and angular velocity w.r.t $\mathcal{F}_W$, are given by the vectors $\boldsymbol{v} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$, expressed in $\mathcal{F}_W$ and $\mathcal{F}_B$, respectively. We can write the system dynamics using the Newton-Euler equations:

$$\dot{\boldsymbol{p}} = \boldsymbol{S}_\lambda \boldsymbol{v} \tag{2a}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{S}_\lambda (\boldsymbol{a}_B + \boldsymbol{R}_B(\dot{\boldsymbol{\omega}} \times \boldsymbol{r}_T + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \boldsymbol{r}_T)) \tag{2b}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \tag{2c}$$

$$\begin{aligned} \dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}(\boldsymbol{\tau}_a - \boldsymbol{\omega} \times (\boldsymbol{J}\boldsymbol{\omega}) + \underbrace{\boldsymbol{r}_{\text{com}} \times (\boldsymbol{R}_B^\top m\boldsymbol{g})}_{\boldsymbol{\tau}_{\text{com}}} \\ + \underbrace{\boldsymbol{r}_T \times (\lambda \boldsymbol{R}_B^\top \boldsymbol{f}_c)}_{\boldsymbol{\tau}_{\text{c}}} + \Delta\boldsymbol{\tau}) \end{aligned} \tag{2d}$$

$$\boldsymbol{a}_B = m^{-1}(\boldsymbol{R}_B \boldsymbol{f}_a + \lambda \boldsymbol{f}_c + \Delta\boldsymbol{f}) + \boldsymbol{g}. \tag{2e}$$

The actuation and disturbance forces $\boldsymbol{f}_a$ and $\Delta\boldsymbol{f}$ are assumed to act directly on the geometric center, which results in the body acceleration $\boldsymbol{a}_B$.

The matrix $\boldsymbol{S}_\lambda$ implements the constraints imposed by the contact with the surface. In particular, it truncates the linear dynamics in (2a) and (2b) along $\boldsymbol{z}_S$ (the surface normal direction) if $\lambda = 1$. Additionally, the contact force $\boldsymbol{f}_c$ is only considered in (2d) and (2e) if $\lambda = 1$.

III. CONTROL DESIGN

A. Overview

We aim to integrate and address the interaction effects discussed in Section I by designing a controller with the following properties:

(a) Hybrid tracking of the EE pose and contact force according to a pose/force reference trajectory. The force control should use a FT sensor for direct force feedback compensating uncertainties and external disturbances.

(b) Autonomous transitioning between contact-less and contact-based flight control, accounting for interaction constraints affecting the system dynamics.

(c) Autonomous transitioning between static and dynamic friction control.

(d) Incorporation of bounds on the dynamics and reference to account for infeasible trajectories and inputs.

In order to achieve these properties, we use an MPC approach with a variable model and variable weights. Both the model and the weights are dynamically updated according to the current control mode. Figure 2 gives an overview of the entire control structure.

B. Control modes

From practical and general friction model considerations, we separate a push-and-slide interaction task into three different control modes: 1) free flight, 2) stiction (i.e. static force control), and 3) sliding (i.e. dynamic force control).

As we use a model-based controller, we would like to integrate the knowledge about the current control mode.

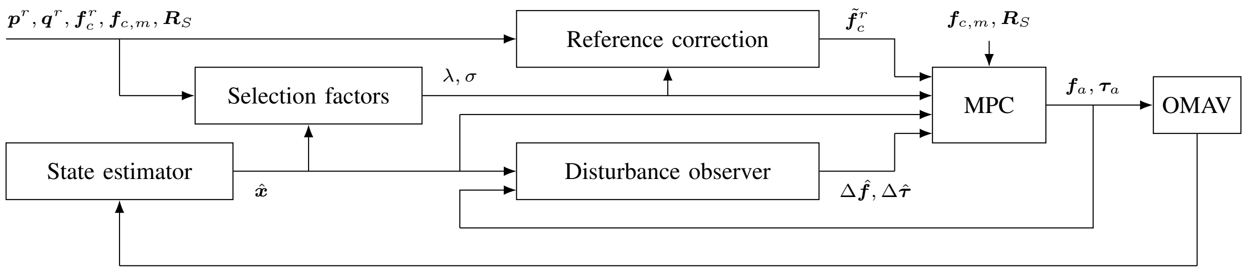
$\boldsymbol{p}^r, \boldsymbol{q}^r, \boldsymbol{f}_c^r, \boldsymbol{f}_{c,m}, \boldsymbol{R}_S$
Reference correction
$\tilde{\boldsymbol{f}}_c^r$
$\boldsymbol{f}_{c,m}, \boldsymbol{R}_S$
Selection factors
λ, σ
MPC
$\boldsymbol{f}_a, \boldsymbol{\tau}_a$
OMAV
State estimator
$\hat{\boldsymbol{x}}$
Disturbance observer
$\Delta\hat{\boldsymbol{f}}, \Delta\hat{\boldsymbol{\tau}}$

Fig. 2. Control diagram.

Control mode	λ	σ	Reference tracking
Free flight	0	N/A	3-DoF Position, Attitude
Static force control	1	0	2-DoF Position, Attitude, Force
Dynamic force control	1	1	2-DoF Position, Attitude, Force

TABLE I. Control modes as a result of the selection factors λ and σ.

Therefore, we will use both state and reference trajectory information to change the model dynamics of the MPC as well as the controller tuning.

With this aim, we introduce two *selection factors*, namely $\lambda \in [0,1]$ and $\sigma \in [0,1]$, to distinguish these three modes. The *flight control factor* λ (already introduced in Section II) represents the transition between contact-less and contact-based flights. Additionally, we use the *force control factor* σ to represent the transition between stiction and sliding control modes. In detail, the relation between control modes and selection factors is the following:

- Free flight (FF), $\lambda = 0$: the MAV moves in space, while there is no interaction with the surface.
- Static force control (SFC), $\lambda = 1, \sigma = 0$: the MAV is in contact with the surface and the EE position is constant. Notice that the orientation of the EE can change. The interaction force stays in the friction cone of the EE.
- Dynamic force control (DFC), $\lambda = 1, \sigma = 1$: the MAV is in contact with the surface and the EE linear velocity is non-zero. The orientation of the EE can change. In this case, dynamic friction is acting on the EE.

Table I gives an overview of the available control modes and which states are tracked accordingly.

The current control mode, i.e. the combination of λ and σ, is determined by fusing state and reference trajectory information. It is therefore not a direct result of the optimization, but rather a condition that influences the optimization.

In order to create smooth transitions between the modes, we will use a cosine step function that interpolates between 0 and 1 in a given range $[\underline{x}, \overline{x}] \subset \mathbb{R}$:

$$l(x, \underline{x}, \overline{x}) = \begin{cases} 0, & \text{if } x \leq \underline{x} \\ \frac{1}{2}\left(1 - \cos\left(\frac{x-\underline{x}}{\overline{x}-\underline{x}}\pi\right)\right) & \text{if } \underline{x} < x \leq \overline{x} \\ 1, & \text{otherwise} \end{cases} . \quad (3)$$

For the remainder of this text, we use the superscript $\cdot^r$ to denote reference variables.

1) Computation of λ: The calculation of the flight control factor λ is done in a similar fashion as in [14] and depends on the current contact force measurement, $\boldsymbol{f}_{c,m} \in \mathbb{R}^3$, and the position error projected onto the contact force reference, $e_p \in \mathbb{R}$, computed as:

$$e_p = \begin{cases} (\boldsymbol{p} - \boldsymbol{p}^r) \cdot \frac{\boldsymbol{f}_c^r}{\|\boldsymbol{f}_c^r\|}, & \text{if } \|\boldsymbol{f}_c^r\| > 0 \\ (\boldsymbol{p} - \boldsymbol{p}^r) \cdot \boldsymbol{z}_S, & \text{otherwise} \end{cases} . \quad (4)$$

We design λ such that the current mode is detected as *interaction*, if the magnitude of the measured contact force, $\|\boldsymbol{f}_{c,m}\|$, is above a threshold $f_{c,max} \in \mathbb{R}$ and if simultaneously the projected position error e_p is below a threshold $e_{p,min} \in \mathbb{R}$. In any other case the mode defaults to being in free flight. While in interaction, we intend to not track the EE position in the surface normal direction. Therefore, we use e_p as trigger to detach from the surface once the reference demands it.

In order to allow smooth transitions, we additionally use a minimum force measurement and a maximum position error threshold, $f_{c,min} \in \mathbb{R}$ and $e_{p,max} \in \mathbb{R}$, respectively. We therefore construct two auxiliary variables $\lambda_f \in [0,1]$ and $\lambda_e \in [0,1]$ which are later merged to compute λ.

$$\begin{aligned} \lambda_f &= l\left(\|\boldsymbol{f}_{c,m}\|, f_{c,min}, f_{c,max}\right) \\ \lambda_e &= 1 - l\left(e_p, e_{p,min}, e_{p,max}\right). \end{aligned} \quad (5)$$

Based on the definition of λ_f and λ_e, the AM is in interaction if both variables are 1. To avoid high noise in the evaluation of this AND condition on continuous variables, we employ a min function and apply a low-pass filter with gain $a_\lambda \in [0,1]$. This results in the following relation for the selection factor at time step $k \in \mathbb{N}_{\geq 0}$:

$$\lambda_k = \lambda_{k-1} + a_\lambda(\min(\lambda_f, \lambda_e) - \lambda_{k-1}). \quad (6)$$

2) Computation of σ: The force control factor σ differentiates between static and dynamic force control, i.e. between stiction and sliding friction. We use this distinction to enable different tuning of the controller for different control modes.

We combine a simple assumption about stiction physics and the current velocity reference to determine if the controller should expect static or dynamic friction:

- If the friction force norm is below a certain threshold $f_{f,min} \in \mathbb{R}$, *or*
- if the norm of the reference velocity projected onto the surface, is above a threshold $v^r_{S,max} \in \mathbb{R}$,

then the control mode is DFC, otherwise it is SFC.

By following this approach, we assume that a friction force below a threshold $f_{f,min}$ indicates dynamic friction (i.e. sliding), while a friction force estimate above a higher threshold $f_{f,max} \in \mathbb{R}$ indicates static friction (i.e. no lateral movement possible).

Instead of estimating the friction force, we form a friction force envelope $f_f \in \mathbb{R}$ to keep track of the recent maximal friction force norm. This is implemented by applying a low-pass filter with different rise and fall times to the projection of the measured interaction force onto the surface:

$$\begin{aligned} f_{f,k} &= f_{f,k-1} + a_{f,k}\Delta f_{f,k}, \quad \text{with} \\ a_{f,k} &= \begin{cases} a_{\text{rise}}, & \text{if } \Delta f_{f,k} > 0 \\ a_{\text{fall}}, & \text{otherwise} \end{cases}, \quad a_{\text{rise}} \gg a_{\text{fall}} \\ \Delta f_{f,k} &= \|\text{diag}\,(1,1,0)\, \boldsymbol{R}_{S,k}^{\top} \boldsymbol{f}_{c,m,k}\| - f_{f,k-1}. \end{aligned} \tag{7}$$

We use a high $a_{\text{rise}} \in [0,1]$ to be responsive to increasing friction forces and a low $a_{\text{fall}} \in [0,1]$ to avoid too fast transitions to DFC. That way, f_f represents the friction force envelope, i.e. an estimate of the highest achieved friction force. Furthermore, we use the projected norm of the velocity reference which is computed as:

$$v_{S,xy}^{r} = \|\text{diag}\,(1,1,0)\, \boldsymbol{R}_S^{\top} \boldsymbol{v}^r\|. \tag{8}$$

As before, for the computation of σ we rely on two auxiliary variables $\sigma_f \in [0,1]$ and $\sigma_v \in [0,1]$ computed as follows

$$\begin{aligned} \sigma_f &= 1 - l\left(f_f, f_{f,min}, f_{f,max}\right) \\ \sigma_v &= l\left(v_{S,xy}^r, v_{S,min}^r, v_{S,max}^r\right). \end{aligned} \tag{9}$$

Since by its definition, σ is 0 if both σ_f *and* σ_v are 0, and 1 otherwise, we use a $\max$ function and a low-pass filter with gain $a_\sigma \in [0,1]$ to compute the force control factor:

$$\sigma_k = \sigma_{k-1} + a_\sigma(\max(\sigma_f, \sigma_v) - \sigma_{k-1}). \tag{10}$$

C. Model Predictive Control

As a state vector, we use the force and torque generated by the actuators, $\boldsymbol{f}_a$ and $\boldsymbol{\tau}_a$, the EE position $\boldsymbol{p}$ and velocity $\boldsymbol{v}$, as well as the body orientation $\boldsymbol{q}$ and angular velocity $\boldsymbol{\omega}$. The input vector is composed of the actuator force and torque time derivatives, $\dot{\boldsymbol{f}}_a$ and $\dot{\boldsymbol{\tau}}_a$:

$$\begin{aligned} \boldsymbol{x} &= \begin{bmatrix} \boldsymbol{f}_a^{\top} & \boldsymbol{\tau}_a^{\top} & \boldsymbol{p}^{\top} & \boldsymbol{v}^{\top} & \boldsymbol{q}^{\top} & \boldsymbol{\omega}^{\top} \end{bmatrix}^{\top} \in \mathcal{X} \subset \mathbb{R}^{19} \\ \boldsymbol{u} &= \begin{bmatrix} \dot{\boldsymbol{f}}_a{}^{\top} & \dot{\boldsymbol{\tau}}_a{}^{\top} \end{bmatrix}^{\top} \in \mathcal{U} \subset \mathbb{R}^{6}. \end{aligned} \tag{11}$$

The system dynamics employed in the MPC, defined by $\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)$, include discretized equations (2a) to (2e) and the relation $[\dot{\boldsymbol{f}}_a{}^{\top} \;\; \dot{\boldsymbol{\tau}}_a{}^{\top}]^{\top} = \boldsymbol{I}_6\boldsymbol{u}$, where $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$ is the identity matrix of dimension $m \in \mathbb{N}_{>0}$. The receding horizon MPC problem is given by:

$$\begin{aligned} \min_{\boldsymbol{u}_0,\dots \boldsymbol{u}_{N-1}} & \sum_{k=0}^{N-1} \left(\|\boldsymbol{h}_x(\boldsymbol{x}_k, \boldsymbol{x}_k^r)\|_{\boldsymbol{Q}_x}^2 + \|\boldsymbol{h}_u(\boldsymbol{u}_k, \boldsymbol{u}_k^r)\|_{\boldsymbol{Q}_u}^2 \right) \\ & \quad + \|\boldsymbol{h}_x(\boldsymbol{x}_N, \boldsymbol{x}_N^r)\|_{\boldsymbol{Q}_N}^2 \\ \text{s.t.} \quad & \boldsymbol{x}_k \in \mathcal{X}, \boldsymbol{u}_k \in \mathcal{U}, \\ & \boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad \boldsymbol{x}_0 = \hat{\boldsymbol{x}}(t), \\ & \lambda_k = \lambda(t), \quad \boldsymbol{R}_{S,k} = \boldsymbol{R}_S(t), \quad \boldsymbol{f}_{c,k} = \boldsymbol{f}_{c,m}(t), \\ & \Delta \boldsymbol{f}_k = \Delta\hat{\boldsymbol{f}}(t), \quad \Delta\boldsymbol{\tau}_k = \Delta\hat{\boldsymbol{\tau}}(t). \end{aligned} \tag{12}$$

The disturbance estimates $\Delta\hat{\boldsymbol{f}} \in \mathbb{R}^3$ and $\Delta\hat{\boldsymbol{\tau}} \in \mathbb{R}^3$ are obtained by a disturbance observer. The cost function is composed of the cost vectors weighted norms with matrices $\boldsymbol{Q}_x \in \mathbb{R}_{\geq 0}^{15\times 15}$, $\boldsymbol{Q}_u \in \mathbb{R}_{>0}^{6\times 6}$, and $\boldsymbol{Q}_N \in \mathbb{R}_{\geq 0}^{15\times 15}$. The cost matrices $\boldsymbol{Q}_x$ and $\boldsymbol{Q}_u$ are constant throughout the prediction horizon. We formulate the cost vectors in a way to meet two objectives simultaneously:

- Minimize the EE pose error;
- Minimize the force tracking error, if a reference force is available and if contact with a surface is established.

The first objective is implemented by using cost terms on the predicted pose and twist error at each time step of the horizon. The attitude error is approximated in $\mathbb{R}^3$ according to [15]. The second objective is achieved by penalizing the difference between the predicted contact force and the force reference at the tip of the EE. The cost vectors are therefore defined as:

$$\begin{aligned} \boldsymbol{h}_x(\boldsymbol{x}_k, \boldsymbol{x}_k^r) &= \begin{bmatrix} \boldsymbol{e}_{\boldsymbol{f}_c k} \\ \boldsymbol{p}_k - \boldsymbol{p}_k^r \\ \boldsymbol{v}_k - \boldsymbol{v}_k^r \\ \text{Im}(\boldsymbol{q}_k \otimes (\boldsymbol{q}_k^r)^{-1}) \\ \boldsymbol{\omega}_k - \boldsymbol{\omega}_k^r \end{bmatrix} \in \mathbb{R}^{15} \\ \boldsymbol{h}_u(\boldsymbol{u}_k, \boldsymbol{u}_k^r) &= \begin{bmatrix} \dot{\boldsymbol{f}}_{a,k} \\ \dot{\boldsymbol{\tau}}_{a,k} \end{bmatrix} \in \mathbb{R}^{6}. \end{aligned} \tag{13}$$

To compute the contact force error $\boldsymbol{e}_{\boldsymbol{f}_c}$, we define a corrected force reference $\tilde{\boldsymbol{f}}_c^r \in \mathbb{R}^3$, whose computation will be detailed in the next paragraph. The contact force error is then defined as the difference between the contact force and the contact force reference:

$$\boldsymbol{e}_{\boldsymbol{f}_c} = \underbrace{m\boldsymbol{S}_\lambda \boldsymbol{a}_B^r - \boldsymbol{R}_B \boldsymbol{f}_a - m\boldsymbol{g} - \Delta\boldsymbol{f}}_{\boldsymbol{f}_c} - \tilde{\boldsymbol{f}}_c^r. \tag{14}$$

a) Force reference correction: We apply a contact force reference correction with the following two objectives: Firstly, in SFC, i.e. when the EE is not moving, we want the measured contact force $\boldsymbol{f}_{c,m}$ to converge to the reference force $\boldsymbol{f}_c^r$. In this case, we correct the reference force by the current force tracking error. We use a proportional-integral correction to avoid steady-state tracking offsets. Secondly, in DFC, we do not apply this correction, as EE needs to overcome static friction forces to track a position reference. We differentiate between these two cases by introducing the control-mode dependent matrix $\boldsymbol{S}_\sigma$:

$$\begin{aligned} \boldsymbol{S}_\sigma &= \lambda_f \boldsymbol{R}_S \text{diag}\,(1-\sigma, 1-\sigma, \lambda)\, \boldsymbol{R}_S^{\top}, \quad \boldsymbol{S}_\sigma \in \mathbb{R}^{3\times 3} \\ \tilde{\boldsymbol{f}}_c^r &= \boldsymbol{f}_c^r + \boldsymbol{S}_\sigma \left(\boldsymbol{K}_p(\boldsymbol{f}_c^r - \boldsymbol{f}_{c,m}) + \boldsymbol{K}_i \int (\boldsymbol{f}_c^r - \boldsymbol{f}_{c,m}) dt \right). \end{aligned} \tag{15}$$

b) Constraints on inputs and states: The sets of states $\mathcal{X}$ and inputs $\mathcal{U}$ are constructed to limit actuator wrench and its derivative similarly as in [11]. In this way, the MPC is aware of the system's physical constraints.

c) Dynamic updating of weight matrices: We define three independent weight matrices $\boldsymbol{Q}_{FF}$, $\boldsymbol{Q}_{SFC}$, and $\boldsymbol{Q}_{DFC} \in \mathbb{R}^{15\times 15}$, and fuse them according to the current control mode. The terminal stage weight $\boldsymbol{Q}_N$ is computed as the scaled version of the stage weight $\boldsymbol{Q}_x$ by the static gain $a_N \in \mathbb{R}_{>0}$. The block-diagonal matrix $\overline{\boldsymbol{R}}_S$ transforms the axis-specific weight tuning from the surface frame into the world frame.

$$\begin{aligned}\boldsymbol{Q}_x &= (1-\lambda)\boldsymbol{Q}_{FF} \\ &\qquad + \lambda\overline{\boldsymbol{R}}_S((1-\sigma)\boldsymbol{Q}_{SFC} + \sigma\boldsymbol{Q}_{DFC})\overline{\boldsymbol{R}}_S^{\top}, \quad (16a)\\ \boldsymbol{Q}_N &= a_N \cdot \boldsymbol{Q}_x, \quad \text{with} \quad (16b)\\ \overline{\boldsymbol{R}}_S &= \text{blockdiag}(\boldsymbol{R}_S, \boldsymbol{R}_S, \boldsymbol{R}_S, \boldsymbol{I}_3, \boldsymbol{I}_3). \quad (16c)\end{aligned}$$

D. Disturbance observer

We employ the same EKF-based disturbance observer as in [11]. Similarly to equations (2a) to (2e), it models the AM as a rigid body with disturbance forces and torques applied on $\mathcal{F}_B$. For simplicity, the model does not include interaction forces and no CoM offset. Therefore, its dynamics are driven by the commanded wrench, $\boldsymbol{f}_a$ and $\boldsymbol{\tau}_a$, corrected by contact force and CoM torque offsets. We use the measured body pose and twist as measurement vector $\boldsymbol{z}_{EKF} = \begin{bmatrix}\boldsymbol{p}_{B,m}^{\top} & \boldsymbol{v}_{B,m}^{\top} & \boldsymbol{q}_m^{\top} & \boldsymbol{\omega}_m^{\top}\end{bmatrix}^{\top}$. The EKF state and input vector are then defined as

$$\hat{\boldsymbol{x}} = \begin{bmatrix}\hat{\boldsymbol{p}}_B^{\top} & \hat{\boldsymbol{v}}_B^{\top} & \hat{\boldsymbol{q}}^{\top} & \hat{\boldsymbol{\omega}}^{\top} & \Delta\hat{\boldsymbol{f}}^{\top} & \Delta\hat{\boldsymbol{\tau}}^{\top}\end{bmatrix}^{\top}, \quad (17a)$$

$$\boldsymbol{u}_{EKF} = \begin{bmatrix}\boldsymbol{f}_a + \boldsymbol{R}_B^{\top}\lambda\boldsymbol{f}_{c,m} \\ \boldsymbol{\tau}_a + \boldsymbol{r}_{\text{com}} \times (\boldsymbol{R}_B^{\top}m\boldsymbol{g}) + \boldsymbol{r}_T \times (\boldsymbol{R}_B^{\top}\lambda\boldsymbol{f}_{c,m})\end{bmatrix}. \quad (17b)$$

E. Actuator allocation

The final rotor speed and tilt angle commands are computed according to the allocation method presented in [16].

IV. EXPERIMENTS

A. Experimental setup

We use a custom built AM similar to the one in [11]. Its design can be seen in Fig. 1. The platform consists of a body with 6 arms that each have 2 counter rotating propellers mounted. Each arm can individually be rotated by servos to allow for full 6-DOF wrench generation. A rigid carbon fiber tube is attached horizontally to the body center. At its end, a Rokubi FT sensor[1] is placed in between the tube and an EE. The raw force measurements are filtered with a low-pass filter and then applied directly in the MPC formulation and in the EKF input (see (17b)). The body center comprises an Intel NUC and a Pixhawk flight controller. State estimation is performed by an EKF that fuses motion capture pose measurements from a VICON system and onboard inertial measurement unit (IMU) measurements. The AM is powered through a ground-based power supply[2] whose lightweight cable can be extended to $80\,\mathrm{m}$, making it feasible for real-world applications. The total mass of the system is $5.2\,\mathrm{kg}$.

We use a customary whiteboard as a contact surface, and a rubber ball with a diameter of approx. $2\,\mathrm{cm}$ as EE. Due to its suspension by cables, the whiteboard is not entirely rigid. Slight movements are regarded as disturbances in the dynamic modeling. The rubber ball has shown to generate high stiction and friction forces, making sliding tasks more difficult to control than low-friction EEs.

Relative weights	$e_{f_{xy}}$	e_{f_z}	$e_{p_{xy}}$	e_{p_z}	$e_{\boldsymbol{q}}$
FF	0	0	1	1	1
SFC	1	1	0.1	0	1
DFC	0	1	0.25	0	1

TABLE II. Relative tuning weights for different control modes.

The MPC was implemented using the ACADO framework [17] in Robot Operating System (ROS). The controller computes control inputs at $100\,\mathrm{Hz}$ with an average MPC solving time of $5\,\mathrm{ms}$. The MPC step size was chosen as $50\,\mathrm{ms}$ with $N = 20$ steps, yielding a horizon length of $1\,\mathrm{s}$.

B. Experiments

Experiments are designed to evaluate the following characteristics of the controller:

- Position tracking of the EE on the surface,
- interaction force tracking,
- autonomous transitioning between the control modes and its effect on the tracking performance.

We use the following trajectory to evaluate these characteristics: Starting in front of a surface, the reference trajectory brings the EE in contact and commands a horizontal translation of $0.8\,\mathrm{m}$ with a velocity of up to $0.6\,\mathrm{m\,s^{-1}}$ while applying a constant force of $3\,\mathrm{N}$.

We compare the performance of our proposed MPC against an impedance controller that employs a PI force-feedback controller on the force measurements at the EE as used in [14].

C. Controller and threshold tuning

The weight matrices were manually tuned for each control mode, balancing aggressive tracking that could lead to slippage of the tool and too soft tracking that could lead to stiction or to too large errors in attitude, resulting in the AM touching the surface.

We use different weight tunings for the three control modes as shown in Table II. Note that we apply different weights on different axes in $\mathcal{F}_S$, such that lateral and normal errors are penalized differently.

The thresholds for the control mode transitions are tuned as follows: To reliably detect contact with a surface, we set $f_{c,min}$ and $f_{c,max}$ to be well over the force sensor noise, and with enough difference to allow a continuous transition. In order to achieve continuous force control during the movement on the surface, we set $e_{p,min}$ and $e_{p,max}$ large enough to allow small position errors due to possible irregularities in the surface shape.

The reference velocity thresholds $v_{S,min}^r$ and $v_{S,max}^r$ are set to small values, as we expect sliding friction for any change in the reference position. The friction envelope thresholds $f_{f,min}$ and $f_{f,max}$ are set based on experiments, such that they are larger than the friction force that we measured during sliding experiments. They are therefore dependent on the materials of the EE and the surface, as well as on the desired contact force.

[1] https://www.botasys.com/rokubi

[2] https://elistair.com/safe-t-tethered-drone-station/

The friction force is filtered according to (7) with $a_{\text{rise}} = 0.05$ and $a_{\text{fall}} = 0.0005$. Throughout all presented experiments, we used the tuning values as shown in Table III.

Threshold	f_c [N]	e_p [m]	f_f [N]	v_S^r [m s^{-1}]
$\cdot_{min}$	1.4	0.1	2.0	0.005
$\cdot_{max}$	1.6	0.15	4.0	0.01

TABLE III. Threshold tuning for mode transitions.

During preliminary experimental testing, we found that the use of the contact force measurement in the angular dynamics equation Eq. (2d) led to instabilities. We therefore removed the explicit computation of the contact torque $\boldsymbol{\tau}_c$ from the system dynamics and the disturbance observer input Eq. (17b). As a consequence, its influence is captured by the disturbance estimate $\Delta\hat{\boldsymbol{\tau}}$ only. We assume that biases in the FT measurements, multiplied with the distance of the EE from the body center, lead to quickly varying torque compensation terms that destabilize the system.

D. Results

Figure 3 shows the position and force tracking during the sliding trajectory with a reference normal force of 3 N for both MPC and impedance controller. Additionally, the bottom plot shows the selection factors that trigger the transitions between control modes.

While the impedance controller shows good accuracy in position tracking, the sudden impact onto the surface and high friction forces during the translation lead to multiple detachments (at 12 s) and unsteady contact force tracking.

The position tracking of the MPC shows that the EE overshoots the reference position after the translation, but then remains stable at the reached destination. We see the delay in the estimated disturbance torque as the main reason for the inaccurate position tracking. Nevertheless, the MPC is able to generate a smooth transition between free flight and contact, without creating force spikes. The flight control factor λ correctly tracks the transitions between FF and interaction control. Once the EE is in contact with the surface, the transition of λ from 0 to 1 enables the force tracking and disables the position tracking along the surface normal. Since the lateral friction force does not grow large enough to reach $f_{f,min}$, the controller remains in DFC.

In Fig. 4, we repeated the sliding experiment with a higher force reference of 5 N. This resulted in a higher friction force, which triggered the transition from DFC to SFC after finishing the second translation. Under the influence of these high friction forces, DFC can lead to instabilities as a result of too aggressive lateral position tracking. On the other hand, SFC prioritizes force tracking in all directions and puts less emphasis on lateral position tracking, therefore stabilizing the AM after finishing the trajectory.

V. CONCLUSION

We presented an NMPC framework that considers the dynamics of an AM in free flight and two different force interaction modes to optimize the control inputs. Different control modes are used to change the tuning of the MPC weights as well as the constraints on the system dynamics. The controller autonomously transitions between these different modes based on the reference trajectory and state estimates. The use of a FT sensor allows for direct force control at the EE. Push-and-slide experiments with high friction coefficients have validated the transitioning between control modes as well as accurate normal force tracking, showing improvements with respect to the impedance controller. Nonlinear friction behaviours due to stiction remain a difficult challenge that needs to be tackled in the future. This would enable sliding interaction tasks with higher required normal forces. While experiments have been performed with high accuracy pose tracking provided by VICON, in real-world applications, we will investigate the use of onboard Visual-Inertial Odometry in combination with GPS to estimate the AM's relative pose to its environment.

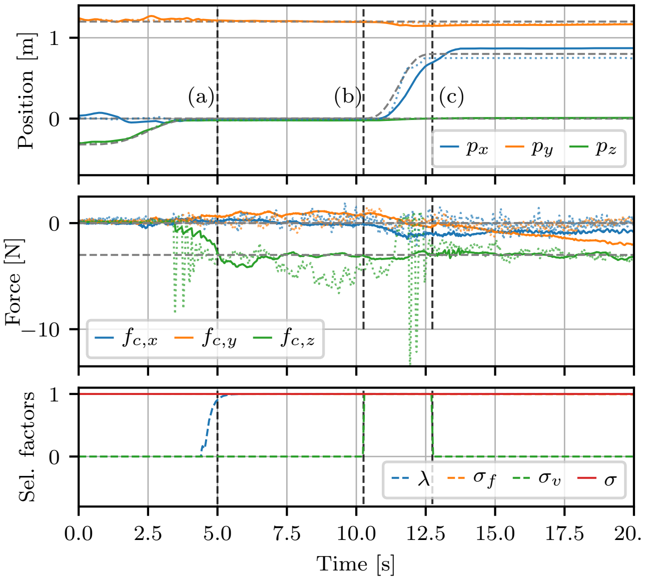

Fig. 3. Position and force tracking of MPC (continuous) and Impedance controller (dotted line), in $\mathcal{F}_S$, with $\|\boldsymbol{f}_c^r\| = 3$ N. Dashed lines represent the reference position and force. Vertical dashed lines indicate the different interaction modes: before (a): free flight, (a)-(b): static interaction, (b)-(c): dynamic interaction, after (c): static interaction. The bottom plot shows the selection factors λ and σ.

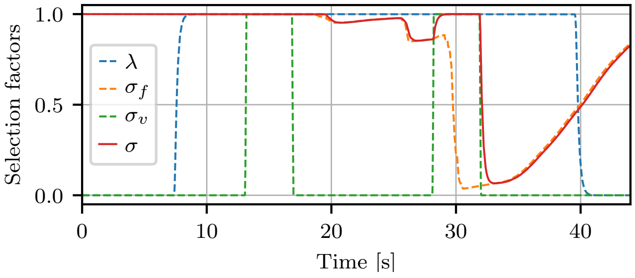

Fig. 4. Selection factors for sliding force tracking with $\|\boldsymbol{f}_c^r\| = 5$ N.

REFERENCES

[1] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial Manipulation: A Literature Review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[2] R. Rashad, J. Goerres, R. G. Aarts, J. B. Engelen, and S. Stramigioli, "Fully Actuated Multirotor UAVs: A Literature Review," *IEEE Robotics & Automation Magazine*, 2020.

[3] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, and A. Franchi, "6D Physical Interaction with a Fully Actuated Aerial Robot," in *2017 International Conference on Robotics and Automation (ICRA)*, Singapore, May 2017, pp. 5190–5195.

[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] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfndler, U. Angst, R. Siegwart, and J. Nieto, "An Omnidirectional Aerial Manipulation Platform for Contact-Based Inspection," *Robotics: Science and Systems XV*, 2019.

[6] M. Tognon, H. A. T. Chávez, E. Gasparin, Q. Sablé, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortés *et al.*, "A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1846–1851, 2019.

[7] R. Rashad, J. B. Engelen, and S. Stramigioli, "Energy tank-based wrench/impedance control of a fully-actuated hexarotor: A geometric port-hamiltonian approach," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6418–6424.

[8] G. Nava, Q. Sablé, M. Tognon, D. Pucci, and A. Franchi, "Direct Force Feedback Control and Online Multi-Task Optimization for Aerial Manipulators," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 331–338, 2020.

[9] D. Tzoumanikas, F. Graule, Q. Yan, D. Shah, M. Popovic, and S. Leutenegger, "Aerial Manipulation Using Hybrid Force and Position NMPC Applied to Aerial Writing," 2020.

[10] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: planning and control," *Autonomous Robots*, vol. 40, no. 4, pp. 631–655, 2016.

[11] M. Brunner, K. Bodie, M. Kamel, M. Pantic, W. Zhang, J. Nieto, and R. Siegwart, "Trajectory Tracking Nonlinear Model Predictive Control for an Overactuated MAV," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5342–5348.

[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*, pp. 1–35, 2020.

[13] J. Pankert and M. Hutter, "Perceptive Model Predictive Control for Continuous Mobile Manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6177–6184, 2020.

[14] K. Bodie, M. Brunner, M. Pantic, S. Walser, P. Pfändler, U. Angst, R. Siegwart, and J. Nieto, "Active Interaction Force Control for Omnidirectional Aerial Contact-Based Inspection," 2020.

[15] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[16] K. Bodie, Z. Taylor, M. Kamel, and R. Siegwart, "Towards Efficient Full Pose Omnidirectionality with Overactuated MAVs," *Proceedings of the 2018 International Symposium on Experimental Robotics*, p. 85–95, 2020.

[17] B. Houska, H. Ferreau, and M. Diehl, "An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 2011.