Aerial Manipulation Using Hybrid Force and Position NMPC Applied to Aerial Writing

Dimos Tzoumanikas[*], Felix Graule[*†], Qingyue Yan[*], Dhruv Shah[‡], Marija Popović[*] and Stefan Leutenegger[*]
[*]Imperial College London, [†]ETH Zurich, [‡]University of California, Berkeley

Abstract—Aerial manipulation aims at combining the maneuverability of aerial vehicles with the manipulation capabilities of robotic arms. This, however, comes at the cost of the additional control complexity due to the coupling of the dynamics of the two systems. In this paper we present a Nonlinear Model Predictive Control (NMPC) specifically designed for Micro Aerial Vehicles (MAVs) equipped with a robotic arm. We formulate a hybrid control model for the combined MAV-arm system which incorporates interaction forces acting on the end effector. We explain the practical implementation of our algorithm and show extensive experimental results of our custom built system performing multiple 'aerial-writing' tasks on a whiteboard, revealing accuracy in the order of millimetres.

I. Introduction

Over the past decades, aerial manipulation has received great attention in the robotics research community, with many different systems in use [22, 13]. The tasks solved by aerial manipulators range from grasping, fetching, and transporting arbitrary objects to pushing against fixed surfaces. Potential use cases are numerous: inspection of infrastructure like bridges or manufacturing plants [8, 27, 3], physical interaction through tools like grinding, welding, drilling and other maintenance work in hard-to-reach places [19, 3], and the autonomous pick-up and transport of objects [12, 1]. All of these tasks require the MAV to be equipped with an additional mechanism which we refer to as the end effector. The coupling of the MAV dynamics with the moving end effector poses an interesting challenge from a control perspective given the inherent instability of MAVs.

The requirements for high precision in real world aerial manipulation applications further increases the difficulty of the task. Efforts made into this direction include [6, 3], where the authors use a fixed end effector on a underactuated and an omnidirectional MAV respectively. Compared to the former method, our approach achieves higher accuracy and flexibility in terms of potential use cases due to our moving end effector. In comparison to the second approach, we achieve on par precision while relying on a simpler, underactuated platform. In summary, we claim to show the following contributions:

- We present a hybrid model which captures the non-linear dynamics of the MAV and considers the quasi-static forces introduced by the attached manipulator.
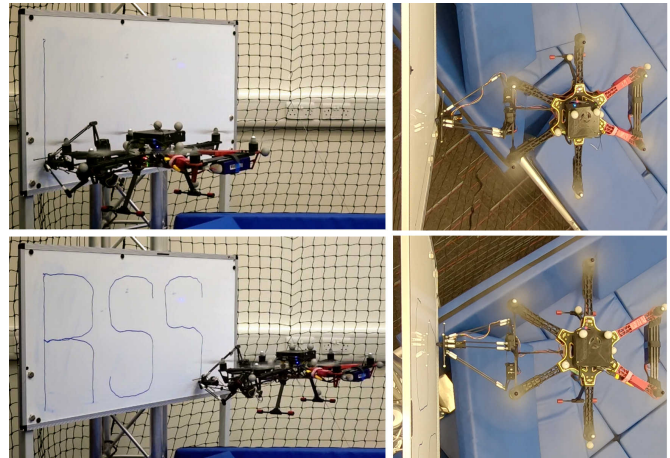
Fig. 1. Our MAV-arm system performing an 'aerial-writing' task.

- We use this generic model in an NMPC jointly controlling the MAV and arm motion.
- We experimentally evaluate our method in 'aerial-writing' tasks using our custom built system. Our results demonstrate high repeatability and accuracy in the order of millimetres across multiple trajectories of varying difficulty.

This paper is organised as follows: In Section II we give an overview of the related work on aerial manipulation. In Sections III and IV we describe our notation, geometric arrangement, and the software architecture of our system. We explain the method in detail in Section V followed by the experimental results in Section VI. Finally, in Section VII we discuss our findings and conclude in Section VIII.

II. Related Work

Aerial manipulation systems can be broadly distinguished based on the MAV type (as being omnidirectional or under-actuated) and the end effector (as being fixed or moving). In general, using an omnidirectional MAV to fulfill complex aerial tasks does not require a moving end effector as the necessary 6 Degrees of Freedom (DoFs) are provided by the MAV itself. Examples include the works presented by [4, 5, 23]. Brescianini and D'Andrea [4] show an omnidirectional MAV that achieves 6-DoF motion by using eight fixed rotors in a non-co-planar configuration. In a subsequent study [5], the same platform is used with a fixed end effector to fetch moving objects. Using a similar approach with a fixed configuration of six tilted rotors, Ryll et al. [23] propose a novel paradigm

to control all 6 DoFs of the MAV while using a rigid end effector to exert forces and torques independently. The system is demonstrated in numerous experimental tasks that include contact, e.g. surface sliding and tilted peg-in-hole task.

In a different approach [11], a setup consisting of six rotors which actively control the thrust direction is proposed. Bodie et al. [3] leverage this system to solve a variety of aerial manipulation tasks with a rigidly mounted, low complexity end effector. The authors further show precise force control when in contact with unstructured environments while relying on visual-inertial state estimation. While this platform allows for accurate 6-DoF flight and longitudinal force exertion with a relatively simple control method, it is mechanically more complex and thus more costly compared to classical multicopter platforms. Recently, a similar approach was followed by Ángel Trujillo et al. [27], who introduce AeroX, an omnidirectional octacopter for contact-based inspection. Their end effector design minimises the torque caused by contact forces and features wheels on its base to allow moving along a surface while remaining in contact. In [19, 20] the authors show a less complex but highly capable tri-tilt-rotor MAV for surface grinding and obstacle manipulation. The control model consists of two disjoint modes: one for free-flight and another for physical interaction.

Employing an underactuated MAV to perform aerial manipulation typically increases the complexity of the end effector since the latter has to provide the additional DoFs. To investigate this, many different end effector designs have been proposed over the last years. We categorise these works by the increasing complexity of the end effector. In [6] an underactuated MAV with a fixed end effector was used for performing contact-based tasks. The authors use a switching mode linear Model Predictive Controller (MPC) with different control models for free flight and in-contact operation. Similarly to our work, they benchmark their approach by performing 'aerial-writing'. In [16], the authors use different lightweight, low complexity grippers to perch, pick up, and transport payload. Using a least squares approach, they estimate the inertial parameters of the grasped objects and use this information to adapt the controller and improve tracking. Moving up in terms of complexity, Kim et al. [14] suggest mounting a 2-DoF robotic arm on a MAV to allow grasping and transporting of objects. The authors propose an adaptive sliding mode controller for the combined system. In [18] the authors present an aerial manipulator with two robotic 2-DoF arms to open a valve. The MAV and arms are controlled as a coupled system which is modelled as a switched nonlinear system during valve turning. In an approach very similar to ours, Lunni et al. [15] propose an NMPC which jointly controls the motion of an underactuated MAV and the attached manipulator. However, the control model is only formulated for free flight operation and the presented experiments do not include contact. In a more recent work, Suarez et al. [25] propose a lightweight, human-sized dual arm system designed to minimise the inertia transferred to the MAV. Each of the two arms add 5 DoFs to the system and the arm control law applied takes into account

that low-cost servo motors do not allow torque control but require position commands. Further, a torque estimator is used to predict the torques produced by the servos and inform the MAV control algorithm accordingly. In order to minimise such disturbances coming from the end effector, Nayak et al. [17] propose a lightweight design mounted on top of an MAV. While attaching a serial robotic arm on an MAV increases the number of tasks it can perform, they only provide limited precision when using low cost and lightweight actuators. Some previous efforts try to mitigate this by mounting a parallel delta arm on an MAV instead. [9, 10] demonstrate a multi-objective dynamic controller which also considers dynamic effects between the MAV and its 3-DoF delta arm. The same system is used in [24] to inspect tree cavities with a camera mounted at the end effector.

In our work we use a low mechanical complexity setup consisting of an underactuated hexacopter and a 3-DoF parallel arm. We propose a general method which could be transferred to other aerial manipulation platforms. We extend the NMPC framework presented in [26] for manipulation tasks and use a hybrid model that captures the effect of contact and coupled MAV-arm dynamics. Our main focus is on the developed software while we use the presented hardware platform to showcase our method. To our best knowledge, we achieve unprecedented accuracy in experiments requiring contact by using an underactuated MAV-delta arm system.

III. NOTATION AND COORDINATE FRAMES

We denote vectors as bold lower case symbols, e.g. $\mathbf{v}$. We use left-hand subscripts, e.g. $_A\mathbf{v}$, to indicate the coordinate representation in the $\underrightarrow{\mathcal{F}}_A$ frame of reference. The rotation matrix $\mathbf{C}_{AB}$ changes the representation of the vector $_B\mathbf{v}$ from $\underrightarrow{\mathcal{F}}_B$ to $\underrightarrow{\mathcal{F}}_A$ as $_A\mathbf{v} = \mathbf{C}_{AB}\,_B\mathbf{v}$. Analogously to the rotation matrix $\mathbf{C}_{AB}$, we use the quaternion $\mathbf{q}_{AB}$ with $\otimes$ denoting the quaternion multiplication. We use $[\mathbf{v}]^\times$ to denote the skew symmetric matrix of the vector $\mathbf{v}$. The motion of the MAV body frame $\underrightarrow{\mathcal{F}}_B$ (x: forward, y: left, z: upward) is expressed with respect to the World frame $\underrightarrow{\mathcal{F}}_W$ (z: upward).

IV. SYSTEM OVERVIEW

An overview of the different software components of the proposed system is outlined in Figure 2. The NMPC is given full state trajectory commands for the MAV and the end effector corresponding to a given aerial manipulation task. Based on those references and the estimated system state, it produces the desired MAV body moments, collective thrust, and end effector position. The control allocation block is responsible for converting the moments and thrust into individual motor commands while the inverse kinematics block computes the desired link angles for the given end effector position. All algorithms run onboard at a rate of 100 Hz while the estimate of the MAV position and orientation is provided externally. The different coordinate frames are displayed in Figure 3.
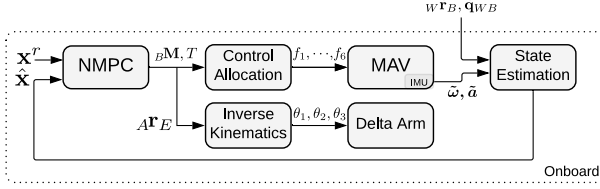
Fig. 2. An overview of the software running onboard our MAV in an aerial manipulation task. We use ROS to interface with the MAV and the motion capture system while the other software blocks form a single executable.
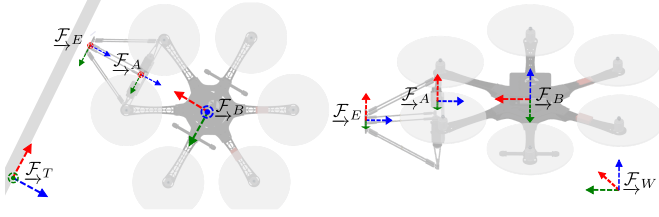

Fig. 3. The coordinate frames used in this paper. Specifically, $\mathcal{F}_W$, $\mathcal{F}_B$, $\mathcal{F}_A$, $\mathcal{F}_E$, and $\mathcal{F}_T$ stand for the World, MAV body, arm, end effector, and contact frame, respectively.

V. METHOD

A. Hybrid Modelling

The standard Newton-Euler equations are used to model the combined MAV-arm dynamics. We model the MAV as a single rigid body object and only consider quasi-static forces introduced by the arm dynamics and its interaction with the environment. Overall, the combined dynamics take the following form:

$$_W\dot{\mathbf{r}}_B = {}_W\mathbf{v}_B\,, \tag{1a}$$

$$\dot{\mathbf{q}}_{WB} = \frac{1}{2}\mathbf{\Omega}({}_B\boldsymbol{\omega})\mathbf{q}_{WB}\,, \tag{1b}$$

$$_W\dot{\mathbf{v}}_B = \frac{1}{m_c}\mathbf{C}_{WB}\,({}_B\mathbf{F}_r + {}_B\mathbf{F}_e) + {}_W\mathbf{g}, \tag{1c}$$

$$_B\dot{\boldsymbol{\omega}} = \mathbf{J_c}^{-1}({}_B\mathbf{M}_r + {}_B\mathbf{M}_e - {}_B\boldsymbol{\omega}\times\mathbf{J_c}{}_B\boldsymbol{\omega}), \tag{1d}$$

$$\mathbf{\Omega}({}_B\boldsymbol{\omega}) = \begin{bmatrix} {}_B\boldsymbol{\omega}^\times & {}_B\boldsymbol{\omega} \\ -{}_B\boldsymbol{\omega}^\top & 0 \end{bmatrix}, \tag{1e}$$

where m_c, $\mathbf{J_c}$ are the combined MAV-arm mass and inertia tensors, respectively and $_W\mathbf{g}$ the acceleration due to gravity. Regarding the forces and moments $_B\mathbf{F}_i, {}_B\mathbf{M}_i$, we use the subscript $i \in \{r, e\}$ to distinguish the ones generated by the MAV motors r from the ones caused by the end effector movement e and its potential contact with the environment. In our system, the MAV motor-generated forces and moments are given by:

$$_B\mathbf{F}_r := [0, 0, T]^\top\,, \quad T = \sum_{i=1}^{6} f_i, \tag{2a}$$

$$_B\mathbf{M}_r := \sum_{i=1}^{6}\left(f_{i\,B}\mathbf{r}_i \times {}_B\mathbf{e}_z + (-1)^{i+1}k_m f_{i\,B}\mathbf{e}_z\right), \tag{2b}$$

with $f_i \in \mathbb{R}$ the thrust produced by the ith motor, $_B\mathbf{r}_i$ its position with respect to $\mathcal{F}_B$, k_m the known thrust to moment coefficient and $_B\mathbf{e}_z = [0, 0, 1]^\top$. Equation (2) can be summarised as $\begin{bmatrix} {}_B\mathbf{M}_r^\top, & T \end{bmatrix} = \mathbf{A}\begin{bmatrix} f_1 & f_2 & \dots & f_6 \end{bmatrix}^\top$ with $\mathbf{A} \in \mathbb{R}^{4\times6}$ the allocation matrix related to the MAV geometry as described in [26]. $_B\mathbf{F}_e$ and $_B\mathbf{M}_e$ are given by:

$$_B\mathbf{F}_e := \mathbf{C}_{BE\,E}\mathbf{F}_c, \tag{3a}$$

$$_B\mathbf{M}_e := {}_B\mathbf{r}_E \times {}_B\mathbf{F}_e + ({}_B\mathbf{r}_E - {}_B\mathbf{r}_{E_0}) \times (\mathbf{C}_{BW}\,m_e\,{}_W\mathbf{g}), \tag{3b}$$

where $_E\mathbf{F}_c$ is the contact force acting on the end effector expressed in its frame $\mathcal{F}_E$ and $_B\mathbf{r}_{E_0} \in \mathbb{R}^3$ the nominal end effector position which results in no Centre of Mass (CoM) displacement. The two terms in (3b) represent the moments due to contact and due to the displacement of the CoM respectively. The combined mass $m_c := m + m_e$ is the sum of the MAV and end effector mass, respectively, while the combined rotational inertia can be computed as $\mathbf{J_c} := \mathbf{J} + m_e\text{diag}({}_B\mathbf{r}_E - {}_B\mathbf{r}_{E_0})^2$ with $\mathbf{J} = \text{diag}(J_x, J_y, J_z)$ the inertia tensor of the MAV (including the arm in nominal position) and $\text{diag}(\cdot)$ the corresponding diagonal matrix.

Regarding the contact force, we assume that this can be approximated with a linear spring model as:

$$_E\mathbf{F}_c = \mathbf{C}_{ET}\,(k_s\,{}_T\mathbf{r}_{E_z}), \tag{4}$$

where k_s is a known spring coefficient and $_T\mathbf{r}_{E_z}$ is the normal component of the contact surface penetration. This way, the controller can anticipate contact before it even happens and there is no need for a switching mode controller (one for free flight and another one for contact dynamics).

B. Model Based Control

For the control formulation we define the following control state and input:

$$\mathbf{x} := \left[{}_W\mathbf{r}_B^\top, {}_W\mathbf{v}_B^\top, \mathbf{q}_{WB}^\top, {}_B\boldsymbol{\omega}^\top\right]^\top \in \mathbb{R}^6 \times S^3 \times \mathbb{R}^3, \tag{5a}$$

$$\mathbf{u} := \left[{}_B\mathbf{M}_r^\top, T, {}_A\mathbf{r}_E^\top\right]^\top \in \mathbb{R}^7. \tag{5b}$$

Note that we use $_B\mathbf{r}_E$ for the formulation of the control model, while $_A\mathbf{r}_E$ is used in the control input. We use the constant and known homogeneous transformation $\boldsymbol{T}_{BA}$ to change the coordinate representation of these position vectors.

We use the following error functions for the position of the MAV, the position of the end effector, the MAV linear and angular velocity, the orientation, the contact force and the control input, respectively:

$$\mathbf{e}_{rB} = {}_W\mathbf{r}_B - {}_W\mathbf{r}_B^r, \tag{6a}$$

$$\mathbf{e}_{rE} = {}_W\mathbf{r}_E - {}_W\mathbf{r}_E^r, \tag{6b}$$

$$\mathbf{e}_v = {}_W\mathbf{v}_B - {}_W\mathbf{v}_B^r, \tag{6c}$$

$$\mathbf{e}_\omega = {}_B\boldsymbol{\omega} - \mathbf{C}_{BB^r}\,{}_{B^r}\boldsymbol{\omega}^r, \tag{6d}$$

$$\mathbf{e}_q = [\mathbf{q}_{WB}^{-1} \otimes \mathbf{q}_{WB}^r]_{1:3}, \tag{6e}$$

$$e_f = f_c - f_c^r, \tag{6f}$$

$$\mathbf{e}_u = \mathbf{u} - \mathbf{u}^r, \tag{6g}$$

with $f_c := {}_E\mathbf{F}_{c_z}$ and the superscript r used to denote the time-varying reference quantities. The optimal input sequence $\mathbf{u}^*$ is obtained by the online solution of the following constrained optimisation problem:

$$\mathbf{u}^* = \underset{\mathbf{u}_0,\ldots,\mathbf{u}_{N_f}}{\operatorname{argmin}} \left\{ \Phi(\mathbf{x}_{N_f}, \mathbf{x}_{N_f}^r) + \sum_{n=0}^{N_f-1} L(\mathbf{x}_n, \mathbf{x}_n^r, \mathbf{u}_n) \right\}, \quad (7a)$$

$$\text{s.t.} : \mathbf{x}_{n+1} = \mathbf{f}_d(\mathbf{x}_n, \mathbf{u}_n), \quad (7b)$$

$$\mathbf{x}_0 = \hat{\mathbf{x}}, \quad (7c)$$

$$u_{\text{lb}} \le u_i \le u_{\text{ub}}, \quad i = 1, \ldots, 7, \quad (7d)$$

with N_f the discrete horizon length, $\mathbf{f}_d$ the discrete version of the dynamics given in Equations (1) – (3), $\hat{\mathbf{x}}$ the latest state estimate and $u_{\text{lb}}, u_{\text{ub}}$ appropriate lower and upper bounds for the control input defined in (5). For the intermediate L and final terms Φ we use quadratic costs of the form $\mathbf{e}_i^\top \mathbf{Q}_i \mathbf{e}_i$ $\forall \mathbf{e}_i \in \{\mathbf{e}_{rB}, \mathbf{e}_{rE}, \mathbf{e}_v, \mathbf{e}_\omega, \mathbf{e}_q, e_f, \mathbf{e}_u\}$ as defined in (6) where the gain matrices $\mathbf{Q}_i \succcurlyeq 0$ were experimentally tuned.

The optimal control problem is implemented using a modified version of the CT toolbox [7] with a 10 ms discretisation step and a 2 s constant prediction horizon. We use a Runge-Kutta 4 integration scheme followed by a re-normalisation for the quaternion. As common in receding horizon control, the first input $\mathbf{u}_0^*$ is applied to the system and the whole process is repeated once a new state estimate $\hat{\mathbf{x}}$ becomes available. The motor commands $\mathbf{f} = \begin{bmatrix} f_1 & f_2 & \ldots & f_6 \end{bmatrix}^\top$ for the MAV are obtained by solving the following Quadratic Program (QP):

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}} \left(\left\| \mathbf{A}\mathbf{f} - \mathbf{u}_{0_{1:4}}^* \right\|_{\mathbf{W}}^2 + \lambda \left\| \mathbf{f} \right\|_2^2 \right), \quad (8a)$$

$$\text{s.t.} : f_{\min} \le f_i \le f_{\max}, \quad i = 1, \ldots, 6, \quad (8b)$$

where $f_{\min}, f_{\max} \in \mathbb{R}$ are the minimum and the maximum motor thrust, $\mathbf{W} \in \mathbb{R}^{4\times4}$ is a gain matrix and $\lambda = 10^{-7}$ a regularisation parameter. The end effector position commands $\mathbf{u}_{0_{5:7}}^*$ are mapped into servo angle commands $\theta_1, \theta_2, \theta_3$ by solving the inverse kinematics problem for the delta arm explained in Section V-C. In the case of an infeasible (e.g. outside the arm's workspace) or unsafe end effector position command (e.g. one that results collision between the MAV propellers and the arm's links), the position command is reprojected onto the boundary of the feasible and safe to operate workspace. In practice, this was rarely the case as the MAV and end effector reference trajectories are designed so that the end effector operates close to its nominal position. In this way the usable workspace is maximised while the effect of the CoM displacement (which is captured by our control model) is minimised.

Our C++ implementation of the above, requires 6.7 ms with a standard deviation of 0.57 ms per iteration. On average, 98% of the computation time is spent on the optimisation problems defined in (7) and (8).

We would like to highlight that our method is generic enough to be applied to different types of vehicles such as omnidirectional ones and/or other types of manipulators. In these cases, the control input, the control allocation and the arm kinematics have to be adapted based on the vehicle and manipulator type. Similarly, the model can easily be extended to capture aerodynamic friction, gyroscopic moments, handle multiple contact points, or use more sophisticated contact models (e.g. ones that include a combination of linear springs and dampers). Similarly, the writing task which we use for the experimental evaluation, is just an example application that requires precision. We believe that our algorithms are adaptable to other tasks such as inspection through contact.

C. Delta Arm Kinematics

Our MAV is equipped with a custom built 3-DoF delta arm [21]. Its main advantages are speed, as its few moving parts are made of lightweight materials, precision and the easy to solve forward and inverse kinematics. The forward kinematics problem (i.e. determining the position of the end effector ${}_A\mathbf{r}_E$ given the joint angles $\theta_1, \theta_2, \theta_3$) can be solved by computing the intersection points of three spheres (shown in Figure 4) of radius l with the following centres:

$$\begin{aligned} {}_A\mathbf{r}_{J_1} &= \big(R - r + L\cos(\theta_1)\big){}_A\mathbf{e}_x - \sin(\theta_1){}_A\mathbf{e}_z, \\ {}_A\mathbf{r}_{J_2} &= \mathbf{C}_z(120^o)\Big(\big(R - r + L\cos(\theta_2)\big){}_A\mathbf{e}_x - \sin(\theta_2){}_A\mathbf{e}_z\Big), \\ {}_A\mathbf{r}_{J_3} &= \mathbf{C}_z(240^o)\Big(\big(R - r + L\cos(\theta_3)\big){}_A\mathbf{e}_x - \sin(\theta_3){}_A\mathbf{e}_z\Big), \end{aligned}$$

where R, r, L correspond to the arm physical parameters shown in Figure 4, ${}_A\mathbf{e}_x = [1, 0, 0]^\top$, ${}_A\mathbf{e}_z = [0, 0, 1]^\top$ and $\mathbf{C}_z(120^o), \mathbf{C}_z(240^o)$ rotation matrices of 120° and 240° degrees around ${}_A\mathbf{e}_z$. The maximum number of intersection points is two which corresponds to an end effector position above (${}_A\mathbf{r}_{E_z} > 0$) and below (${}_A\mathbf{r}_{E_z} < 0$) the arm base. In our setup, solutions above the arm base are mechanically impossible and thus rejected. For the inverse kinematics the intersection between a sphere with radius l and a circular disk with radius L has to be computed for every joint angle. For the first joint, as shown in Figure 4, the centre of the sphere is ${}_A\mathbf{r}_{P_1} = {}_A\mathbf{r}_E + r\,{}_A\mathbf{e}_x$ with ${}_A\mathbf{e}_x = [1, 0, 0]^\top$ while the centre of the circular disk is ${}_A\mathbf{r}_{S_1} = R\,{}_A\mathbf{e}_x$. Given the intersection point ${}_A\mathbf{r}_{I_1}$, the joint angle can be recovered as $\theta_1 = \arcsin({}_A\mathbf{r}_{I_{1_z}}/L)$. The joint angles θ_2 and θ_3 can be computed by performing the same procedure for the spheres with centres ${}_A\mathbf{r}_{P_2}, {}_A\mathbf{r}_{P_3}$, same radius l and the unit disks centred at ${}_A\mathbf{r}_{S_2}$ and ${}_A\mathbf{r}_{S_3}$ with radius L. The points ${}_A\mathbf{r}_{P_i}, {}_A\mathbf{r}_{S_i}$ $\forall i = 2, 3$ can be easily computed as follows:

$$\begin{aligned} {}_A\mathbf{r}_{P_2} &= {}_A\mathbf{r}_E + r\,\mathbf{C}_z(120^o){}_A\mathbf{e}_x, & (10a) \\ {}_A\mathbf{r}_{P_3} &= {}_A\mathbf{r}_E + r\,\mathbf{C}_z(240^o){}_A\mathbf{e}_x, & (10b) \\ {}_A\mathbf{r}_{S_2} &= \mathbf{C}_z(120^o){}_A\mathbf{r}_{S_1}, & (10c) \\ {}_A\mathbf{r}_{S_3} &= \mathbf{C}_z(240^o){}_A\mathbf{r}_{S_1}. & (10d) \end{aligned}$$

D. Trajectory Generation

We use a trajectory generator to map arbitrary sets of characters to end effector trajectories. We use a constant acceleration motion model to generate trajectories with a smooth velocity profile. This is of special importance when
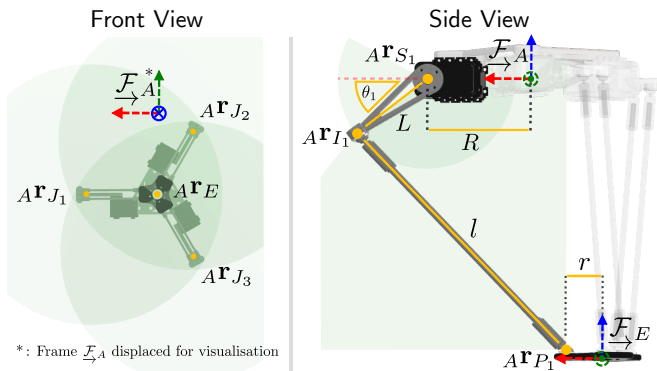
Fig. 4. Two different views of a 3D model of the delta arm used. The green areas show the virtual spheres and disks used for the solution of the forward and inverse kinematics.

the reference path contains sharp edges. Through our software we can adjust the velocity and acceleration profile by changing the maximum $\|_W\mathbf{v}_E^r\|$ and $\|_W\mathbf{a}_E^r\|$. Once the trajectory for the end effector has been computed, we proceed with the computation of the reference position $_W\mathbf{r}_B^r$ and velocity $_W\mathbf{v}_B^r$ for the MAV as follows:

$$_W\mathbf{r}_B^r = {}_W\mathbf{r}_E^r - \mathbf{C}_{WB}{}_B\mathbf{r}_{E_0}, \tag{11a}$$
$$_W\mathbf{v}_B^r = {}_W\mathbf{v}_E^r - \mathbf{C}_{WB}{}_B\boldsymbol{\omega}^r \times {}_B\mathbf{r}_{E_0}. \tag{11b}$$

The reference MAV orientation $\mathbf{q}_{WB}^r$ is chosen such that the end effector is always perpendicular to the contact surface, assuming perfect position tracking, while the reference force is obtained using the spring model and the nominal displacement of the end effector into the contact surface's normal direction. In our framework each trajectory is accompanied by an appropriate flag which disables or enables the position tracking for the end effector. This is achieved by setting the appropriate gains to zero. In that case the NMPC may decide to move the arm to assist the reference tracking of the MAV due to the CoM displacement. This potentially unwanted behaviour can be avoided by further penalising the arm displacement from its nominal position (i.e. by increasing the input gains). However, it is an interesting capability enabled by our hybrid modelling.

VI. Experiments

A. Experimental Setup

The experiments presented in this section were performed using a custom built hexacopter equipped with a sideways mounted delta arm manipulator. The MAV features a frame with a 550 mm diameter, a Pixracer flight controller running a modified version of the PX4 firmware, and an Intel NUC-7567U onboard computer running Ubuntu 16.04. It uses 960 KV motors and DJI 9450 propellers. The delta arm uses magnetic universal joints for the connection of the servos with the end effector which maximizes the workspace, minimises backlash and allows the arm to disassemble during possible crashes preventing it from breaking. The arm uses three Dynamixel AX-18A servo motors which are comparably fast and accurate but have limited maximum torque. The system

TABLE I
NUMERIC VALUES OF MAV AND ARM PARAMETERS

J_x	0.042 kg m^2	k_s	42.95 N/m
J_y	0.054 kg m^2	R	7.2 cm
J_z	0.110 kg m^2	r	2.5 cm
k_m	1.58×10^{-2} Nm/N	L	6.5 cm
m_e	0.058 kg	l	20.2 cm

is powered by a 4S 4500 mAh battery and has total weight of 2.6 kg. The end effector holds the pen which is mounted on a spring to provide additional compliance. We set the coefficient of the contact model in (4) to match the used spring. The applied force is measured by a SingleTact force sensor mounted at the end of the spring. We estimate the spring coefficient k_s by measuring the applied force for known tip displacements. The dimensions of the delta arm were obtained from a highly detailed CAD file and were verified manually. We measured the inertia of the MAV $\mathbf{J}$ by measuring its angular response to constant input torque while it is hanging to freely rotate. The thrust to moment coefficient k_m is measured using a thrust stand. A table with the numeric values of the system parameters is given in Table I and a photo showing the platform and its different components is shown in Figure 5.
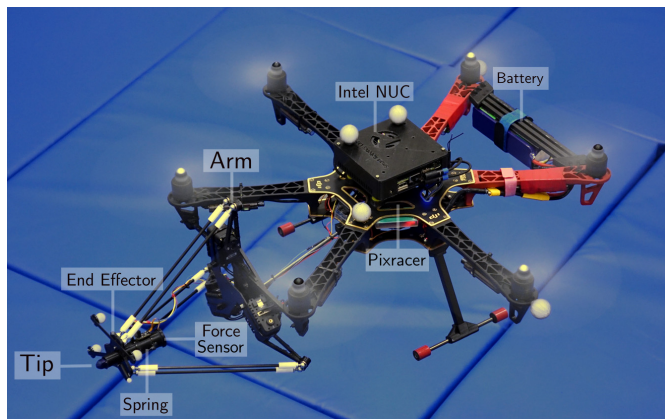

Fig. 5. The aerial manipulation platform used in the 'aerial writing' experiments in Section VI with its individual components labelled.

We use a Vicon motion capture system to provide external pose estimates. The contact surface is a 1×0.5 m whiteboard for which we estimate its pose $\boldsymbol{T}_{WT}$ based on Vicon measurements. Each experiment consisted of the following different trajectory stages: (i) approach, (ii) write, and (iii) return home. The end effector was enabled, using the appropriate flags as mentioned in Section V-D, for the trajectory writing in (ii) and disabled for the rest. Our analysis mainly focuses on the trajectory writing which includes contact whereas for the other two parts (approach/return) the MAV performs simple position tracking. We evaluate the accuracy of our system by comparing the reference trajectories to those estimated by the Vicon motion capture system. In addition, we use a vision-based

error as a performance metric. This is because we observed inaccuracies in the Vicon measurements stemming from either bad calibration, poor object visibility, or marker reflections on the whiteboard surface. The visual error is computed by running a 2D Iterative Closest Point (ICP) method [2] on a filtered and rectified photo of the final writing and a rendering of the planned path. After registration of the two point sets, we use the nearest neighbour distance to evaluate the accuracy.

In the following we show four experiments: in Section VI-B we present detailed and repeatable results for two different trajectories, namely RSS and $E = mc^2$. We then show consistent tracking performance across varying MAV velocities and text sizes in Sections VI-C and VI-D, respectively.

B. Trajectory Tracking

Figure 6 shows the tracking of the RSS trajectory visualised in the contact frame $\underrightarrow{\mathcal{F}}_T$ for the end effector and the MAV. The maximum reference velocity was set to 7.5 cm/s and the maximum acceleration to 2.5 cm/s^2. The trajectory consists of four contact segments with a combined duration of 65 s. Based on the Vicon estimates the tracking error, shown in Figure 7, of the end effector almost always remains in the $[-10, 10]$ mm range during the contact segments while the same quantity for the MAV is in $[-40, 40]$ mm range. This highlights the efficacy of using a manipulator with faster dynamics than the MAV's for precision tasks such as 'aerial-writing'.

Similarly to the above, Figure 8 shows the trajectory tracking for the more challenging $E = mc^2$ experiment which contains ten contact segments with a combined duration of 63 s. Tracking accuracy is similar as before with the end effector and MAV tracking error in the $[-10, 10]$ and $[-50, 50]$ mm range. The accuracy can be visually verified since the overlapping segments of the 'R' and 'm' coincide almost perfectly. Additionally, the consistent approaching and retracting from the contact surface leads to identical starting points of individual letter segments, e.g. the three horizontal lines of the letter 'E'. In both cases, the maximum error based on the visual error analysis is 10 mm mostly originating from temporary loss of contact. Possible reasons for this are bad estimation of the orientation part of the contact frame transformation $\boldsymbol{T}_{WT}$, the assumption of a perfectly flat contact surface being incorrect but also the finite accuracy of the delta arm. The imperfect tracking along the contact frame normal direction (shown in blue in Figures 7, 9) is also reflected in the reference force tracking.

To prove the repeatability of our approach, we conducted each experiment thrice. We give the relevant tracking statistics for the MAV and arm separately in Figure 10, in which the textured box plots correspond to MAV data and the plain ones to that of the end effector. The median and extreme values for the end effector are significantly lower than the ones for the MAV and consistent with the values reported above. This further shows the benefit of using an aerial manipulator for precise tasks including contact. The MAV tracking accuracy along the z axis was the lowest amongst all axes, as this was most affected by the interaction forces and unmodelled torque disturbances due to the movement of the servos.

C. Velocity Sweep

The aim of the next experiments is to demonstrate the effects of the input velocity and acceleration on the writing accuracy. We performed five iterations of the same Hello trajectory experiment with different velocity and acceleration profiles. The different iterations correspond to maximum velocity $v_{max} \in \{7.5, 12.5, 17.5, 22.5, 27.5\}$ cm/s and maximum acceleration $a_{max} \in \{3.75, 6.25, 8.75, 11.25, 13.75\}$ cm/s^2.

Figure 11 shows the box plots for the MAV and end effector tracking accuracy based on the Vicon measurements. The plots show consistent tracking results in all the different velocity and acceleration settings tested. The numeric values of the tracking error are similar to the ones presented in Section VI-B with the end effector achieving sub centimetre accuracy (per axis) while the MAV error is consistently less than 50 mm. However, by observing the visual error, we see that as the reference velocity increases, the system struggles more with the segments containing curvature e.g. 'e' and 'o'. In contrast, the performance on the straight line segments remains similar.

We believe that the tracking error of the MAV can be further reduced by giving the NMPC dynamically feasible trajectories not only for position and velocity but also acceleration, jerk, and snap. Regarding the end effector tracking error, we generally expect this to increase for reference velocities beyond the ones tested here. This is due to the mismatch between the formulated control model and the real one.

D. Text Size Sweep

In Figure 12 we show the visual error of our system for the same trajectory in four different text sizes ranging from 10 to 40 cm. The consistent accuracy shows that the system can handle the fast direction changes imposed by the small scale.

VII. DISCUSSION

Overall, our system achieves accurate and consistent results over a series of different trajectories. The tracking error of the end effector is significantly lower than that of the MAV; highlighting the accuracy boost due to the utilisation of the arm. We would like to mention that our system was built using relatively low-cost off-the-shelf components and 3D printed parts. This leads to errors in the manufacturing with respect to the reference model, e.g. errors in the true inverse kinematics of the arm due to non-identical dimensions of its links.

Another important issue that we faced during our experiments was the reliance on the motion capture system for localisation. Apart from issues related to WiFi delays, which resulted in temporary loss of tracking, we faced problems with poor object visibility resulting in unreliable estimates of both the static objects, such as the contact frame, and moving ones such as the MAV. In fact, during our data analysis we realised that there are segments where Vicon returned mechanically impossible configurations for our system e.g. end effector position below the surface of the contact frame. Despite these
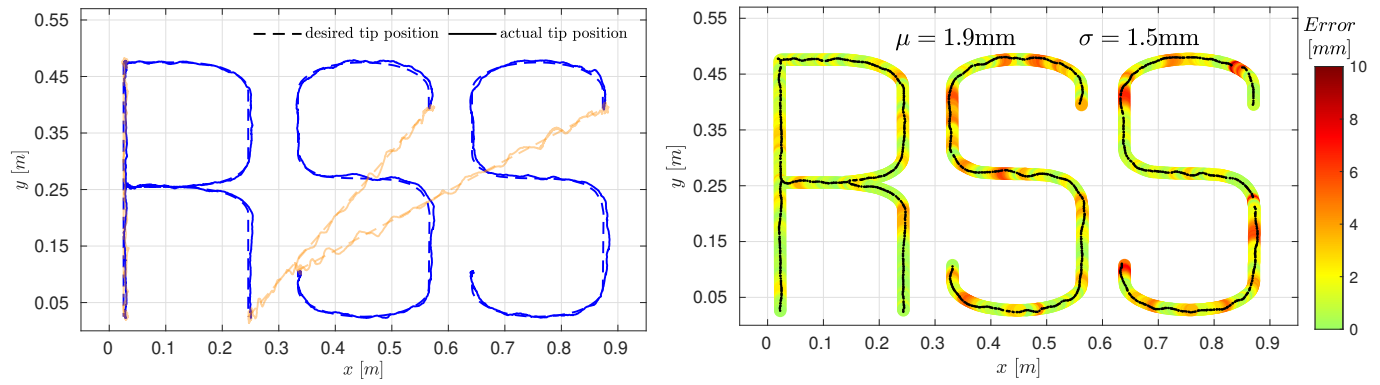
Fig. 6. Reference and actual tip position (left) as estimated by Vicon. Blue corresponds to contact segments while orange refers to free flight. Visual error (right) between reference and actual tip position. The maximum estimated error is lower than 10 mm and is located at discontinuous segments as expected.
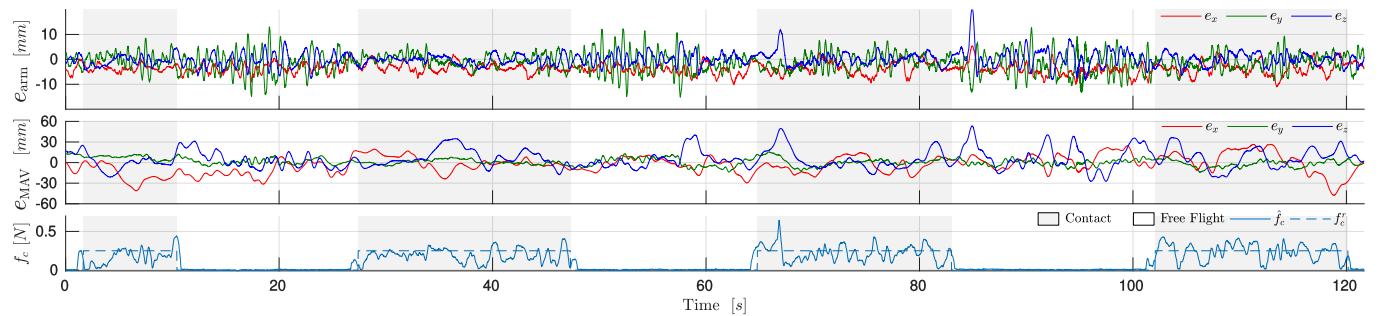

Fig. 7. Reference tracking error of the tip position (top), MAV (middle), and measured contact force(bottom). The tracking error is plotted in the contact frame $\overrightarrow{\mathcal{F}}_T$. The tracking accuracy of the end effector is significantly greater than that of the MAV, given that they remain in the $[-10, 10]$ mm and $[-40, 40]$ mm ranges, respectively.
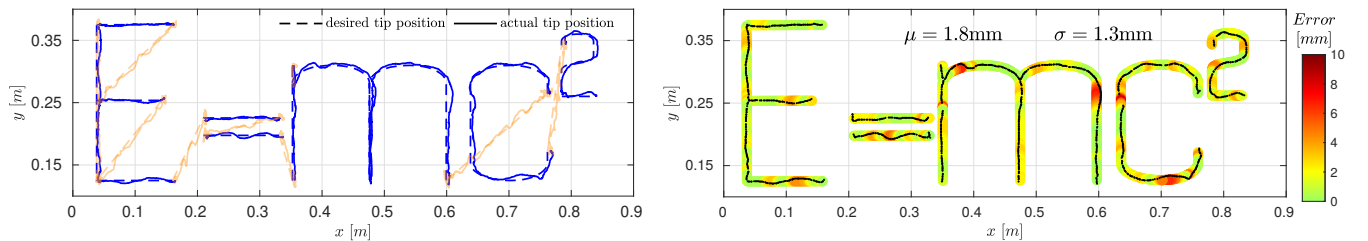

Fig. 8. Reference and actual tip position (left) as estimated by Vicon. Blue corresponds to contact segments while orange refers to free flight. Visual error (right) between reference and actual tip position. Similarly as in the RSS experiment shown in Figure 6, maximum error does not exceed 1 cm.
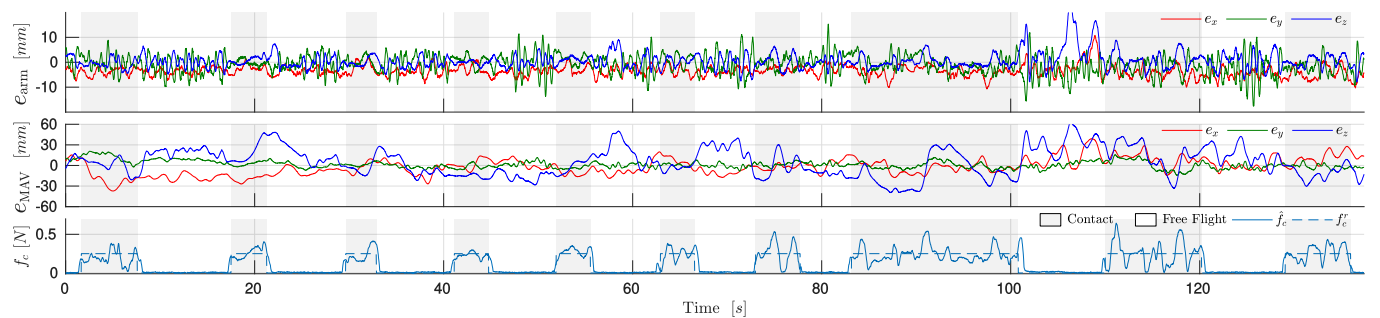

Fig. 9. Reference tracking error of the tip position (top), MAV (middle) and measured contact force (bottom). The tracking accuracy of the end effector is significantly greater than that of the MAV, given that they remain in the $[-10, 10]$ mm and $[-50, 50]$ mm ranges, respectively
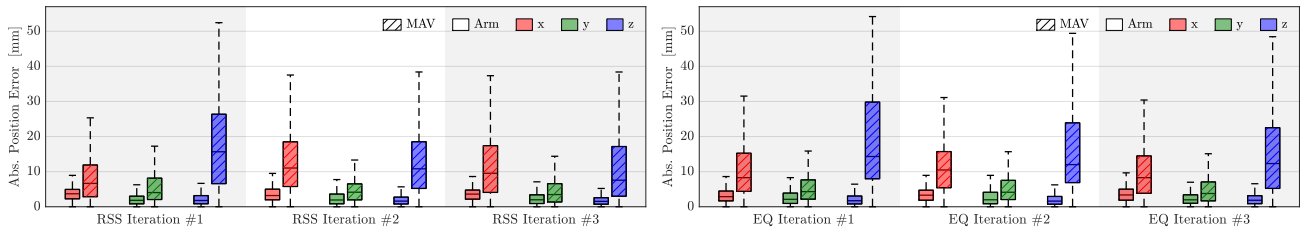
Fig. 10. MAV and end effector box plots of the contact segments for 3 iterations of the RSS trajectory experiment (left) and the more challenging $E = mc^2$ trajectory experiment (right).
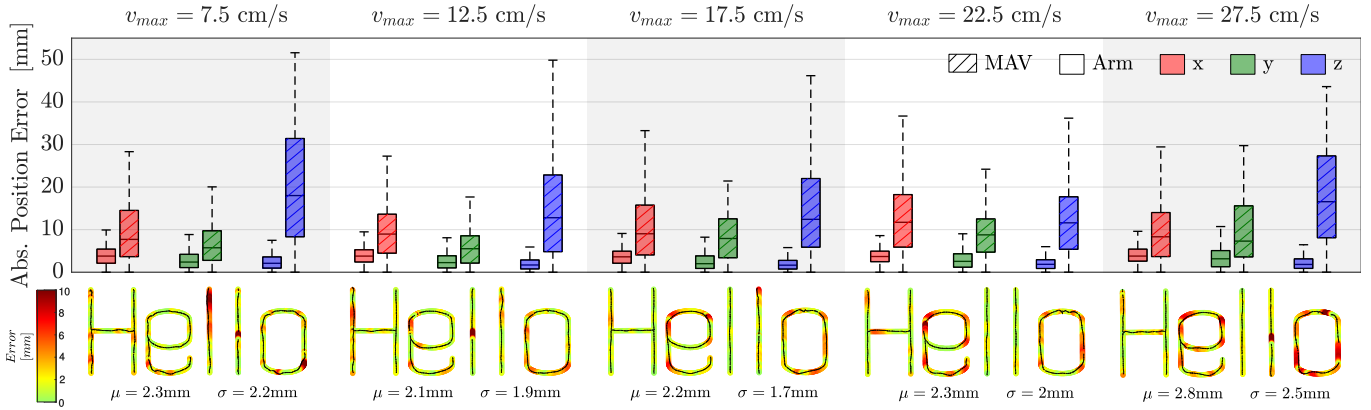

Fig. 11. MAV and end effector box plots (top) and visual errors (bottom) for 5 iterations of the Hello trajectory. Different iterations correspond to different velocity and acceleration profiles.
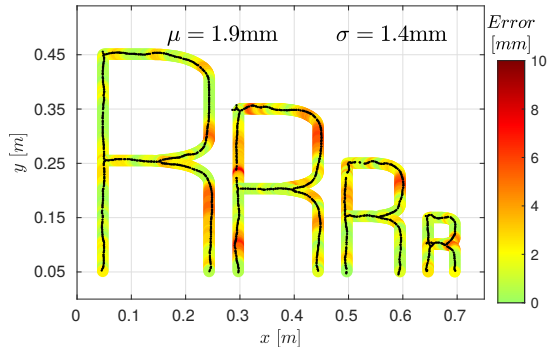

Fig. 12. Visual error plot showing consistent results for varying text sizes.

problems which further propagate into tracking errors, our system successfully handled multiple transitions to contact during the same experiment.

We experimentally verified that for contact tasks, where the main objective is accuracy instead of speed, using a planner respecting full state dynamic feasibility is not an absolute necessity. Despite our simplified motion planner, our system achieves sub-centimetre accuracy. However, we argue that for more aggressive maneuvers, a full state dynamically feasible planner would be required.

VIII. CONCLUSION

We have presented a hybrid model-based algorithm for aerial manipulation using an underactuated MAV with an attached end effector performing 'aerial-writing' tasks on a whiteboard. We demonstrated our system in a series of experiments with trajectories requiring multiple transitions from free-flight to contact and vice-versa. The end effector tracking error consistently remains in the $[-10, 10]$ mm range for trajectories with maximum velocities ranging from 7.5 to 27.5 cm/s, maximum acceleration from 3.75 to 13.75 cm/s^2 and text sizes from 10 to 40 cm. All algorithms run in real-time onboard the MAV. We believe that our method is generic enough to be applied to different types of MAVs and manipulators. We further believe that our framework can be extended to more practical applications such as physical interaction with surfaces e.g. drilling, welding, grinding or inspection through contact.

Regarding future work, online estimating $\boldsymbol{T}_{WT}$ and closing the manipulation task loop with visual feedback would constitute a major improvement to correct for errors stemming from bad calibration, Vicon delays and inaccuracies of the arm. This will allow us to better handle the system's sensitivity to imperfect estimates of transformations such as the contact frame $\boldsymbol{T}_{WT}$ and arm to body transformation $\boldsymbol{T}_{BA}$. The current method further assumes an instantaneous end effector position response, which we plan to substitute with a more realistic model or ultimately formulate the full multi-body dynamics model which would result in a performance boost mainly for faster maneuvers. Finally, we aim to implement and test our method using omnidirectional MAVs which due to their ability of generating lateral forces have superior force exertion capabilities compared to the one presented in this work.

REFERENCES

[1] Frederico Augugliaro, Sergei Lupashin, Michael Hamer, Cason Male, Markus Hehn, Mark W. Mueller, Jan S. Willmann, Fabio Gramazio, Matthias Kohler, and Raffaello D'Andrea. The flight assembled architecture installation: Cooperative construction with flying machines. *IEEE Control Systems Magazine*, 34(4):46–64, Aug 2014. ISSN 1941-000X. doi: 10.1109/MCS.2014.2320359. URL https://ieeexplore.ieee.org/document/6853477.

[2] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992. ISSN 1939-3539. doi: 10.1109/34.121791. URL https://ieeexplore.ieee.org/document/121791.

[3] 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*, June 2019. doi: 10.15607/RSS.2019.XV.019. URL http://www.roboticsproceedings.org/rss15/p19.html.

[4] Dario Brescianini and Raffaello D'Andrea. Design, modeling and control of an omni-directional aerial vehicle. In *IEEE International Conference on Robotics and Automation*, pages 3261–3266, May 2016. doi: 10.1109/ICRA.2016.7487497. URL https://ieeexplore.ieee.org/abstract/document/7487497.

[5] Dario Brescianini and Raffaello D'Andrea. Computationally efficient trajectory generation for fully actuated multirotor vehicles. *IEEE Transactions on Robotics*, 34:555–571, 2018. URL https://ieeexplore.ieee.org/document/8336503.

[6] Georgios Darivianakis, Kostas Alexis, Michael Burri, and Roland Siegwart. Hybrid predictive control for aerial robotic physical interaction towards inspection operations. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 53–58, May 2014. doi: 10.1109/ICRA.2014.6906589. URL https://ieeexplore.ieee.org/document/6906589.

[7] Markus Giftthaler, Michael Neunert, Markus Stauble, and Jonas Buchli. The control toolbox — an open-source c++ library for robotics, optimal and model predictive control. *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, May 2018. doi: 10.1109/simpar.2018.8376281. URL http://dx.doi.org/10.1109/SIMPAR.2018.8376281.

[8] Takahiro Ikeda, Shogo Yasui, Motoharu Fujihara, Kenichi Ohara, Satoshi Ashizawa, Akihiko Ichikawa, Akihisa Okino, Takeo Oomichi, and Toshio Fukuda. Wall contact by octo-rotor uav with one dof manipulator for bridge inspection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5122–5127, Sep. 2017. doi: 10.1109/IROS.2017.8206398. URL https://ieeexplore.ieee.org/document/8206398.

[9] Mina Kamel, Kostas Alexis, and Roland Siegwart. Design and modeling of dexterous aerial manipulator. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4870–4876, Oct 2016. doi: 10.1109/IROS.2016.7759715. URL https://ieeexplore.ieee.org/document/7759715.

[10] Mina Kamel, Simone Comari, and Roland Siegwart. Full-body multi-objective controller for aerial manipulation. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 659–664, June 2016. doi: 10.1109/MED.2016.7536005. URL https://ieeexplore.ieee.org/document/7536005.

[11] Mina Kamel, Sebastian Verling, Omar Elkhatib, Christian Sprecher, Paula Wulkop, Zachary Taylor, Roland Siegwart, and Igor Gilitschenski. Voliro: An Omnidirectional Hexacopter With Tiltable Rotors. *arXiv*, 2018. URL http://arxiv.org/abs/1801.04581.

[12] Chad C. Kessens, Justin Thomas, Jaydev P. Desai, and Vijay Kumar. Versatile aerial grasping using self-sealing suction. In *IEEE International Conference on Robotics and Automation*, pages 3249–3254, 2016. doi: 10.1109/ICRA.2016.7487495. URL https://ieeexplore.ieee.org/document/7487495.

[13] Hossein Bonyan Khamseh, Farrokh Janabi-Sharifi, and Abdelkader Abdessameud. Aerial manipulation—A literature survey. *Robotics and Autonomous Systems*, 107:221 – 235, 2018. doi: 10.1016/j.robot.2018.06.012. URL http://www.sciencedirect.com/science/article/pii/S0921889017305535.

[14] Suseong Kim, Seungwon Choi, and H. Jin Kim. Aerial manipulation using a quadrotor with a two DOF robotic arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4990–4995, 2013. doi: 10.1109/IROS.2013.6697077. URL https://ieeexplore.ieee.org/document/6697077.

[15] D. Lunni, A. Santamaria-Navarro, R. Rossi, P. Rocco, L. Bascetta, and J. Andrade-Cetto. Nonlinear model predictive control for aerial manipulation. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 87–93, 2017. URL https://ieeexplore.ieee.org/document/7991347.

[16] Daniel Mellinger, Quentin Lindsey, Michael Shomin, and Vijay Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2668–2673, 2011. doi: 10.1109/IROS.2011.6094871. URL https://ieeexplore.ieee.org/document/6094871.

[17] Varun Nayak, Christos Papachristos, and Kostas Alexis. Design and control of an aerial manipulator for contact-based inspection. *ArXiv*, abs/1804.03756, 2018. URL https://arxiv.org/abs/1804.03756.

[18] Matko Orsag, Christopher Korpela, Stjepan Bogdan, and Paul Oh. Valve turning using a dual-arm aerial ma-

nipulator. In *2014 International Conference on Un-manned Aircraft Systems (ICUAS)*, pages 836–841, May 2014. doi: 10.1109/ICUAS.2014.6842330. URL https://ieeexplore.ieee.org/document/6842330.

[19] Christos Papachristos, Kostas Alexis, and Anthony Tzes. Technical activities execution with a tiltrotor uas employing explicit model predictive control. *IFAC Proceedings Volumes*, 47(3):11036 – 11042, 2014. ISSN 1474-6670. doi: https://doi.org/10.3182/20140824-6-ZA-1003.02692. URL http://www.sciencedirect.com/science/article/pii/S1474667016433690. 19th IFAC World Congress.

[20] Christos Papachristos, Kostas Alexis, and Anthony Tzes. Efficient force exertion for aerial robotic manipulation: Exploiting the thrust-vectoring authority of a tri-tiltrotor uav. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4500–4505, May 2014. doi: 10.1109/ICRA.2014.6907516. URL https://ieeexplore.ieee.org/document/6907516.

[21] F. Pierrot, C. Reynaud, and A. Fournier. Delta: a simple and efficient parallel robot. *Robotica*, 8(2):105–109, 1990. doi: 10.1017/S0263574700007669. URL https://doi.org/10.1017/S0263574700007669.

[22] Fabio Ruggiero, Vicenzo Lippiello, and Anibal Ollero. Aerial Manipulation: A Literature Review. *IEEE Robotics and Automation Letters*, 3(3):1957–1964, 2018. doi: 10.1109/LRA.2018.2808541. URL https://ieeexplore.ieee.org/document/8299552.

[23] 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, 2019. doi: 10.1177/0278364919856694. URL https://doi.org/10.1177/0278364919856694.

[24] Kelly Steich, Mina Kamel, Paul Beardsley, Martin K. Obrist, Roland Siegwart, and Thibault Lachat. Tree cavity inspection using aerial robots. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4856–4862, Oct 2016. doi: 10.1109/IROS.2016.7759713. URL https://ieeexplore.ieee.org/abstract/document/7759713.

[25] Alejandro Suarez, Antonio Enrique Jimenez-Cano, Victor Manuel Vega, Guillermo Heredia, Angel Rodriguez-Castaño, and Anibal Ollero. Design of a lightweight dual arm system for aerial manipulation. *Mechatronics*, 50:30 – 44, 2018. ISSN 0957-4158. doi: https://doi.org/10.1016/j.mechatronics.2018.01.005. URL http://www.sciencedirect.com/science/article/pii/S0957415818300011.

[26] Dimos Tzoumanikas, Qingyue Yan, and Stefan Leutenegger. Nonlinear MPC with motor failure identification and recovery for safe and aggressive multicopter flight, 2020. URL https://arxiv.org/abs/2002.06598.

[27] Miguel Ángel Trujillo, José Ramiro Martínez de Dios, Carlos Martín, Antidio Viguria, and Aníbal Ollero. Novel aerial manipulator for accurate and robust industrial ndt contact inspection: A new tool for the oil and gas inspection industry. *Sensors*, 19(6), 2019. ISSN 1424-8220. doi: 10.3390/s19061305. URL https://www.mdpi.com/1424-8220/19/6/1305.