Explicit Computations, Simulations and additional Results for the Dynamic Decentralized Control for Protocentric Aerial Manipulators

Technical Attachment to:
"Dynamic Decentralized Control for Protocentric Aerial Manipulators"
2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, May 2017

Marco Tognon1 Burak Yüksel2 Gabriele Buondonno3 Antonio Franchi1

INTRODUCTION

This document is a technical attachment to [1] for explicit computations of the nominal states and the inputs of a Protocentric Aerial Manipulator (PAM) in 2D, using differential flatness property. In [1] these values are used to control a PAM in 3D. Furthermore, considering the aerial manipulator design used for the experiments in that paper, here we investigate the case when the system is non-protocentric: i.e., the manipulating arm is not exactly attached to the CoM of the flying robot, P0. We show the effect of the distance between this attachment point and P0 on the performance tracking a composite trajectory. Finally some additional plots related to the experimental results are provided.

A. Aerial physical interaction

For the reader interested in aerial vehicles physically interacting with the external environment, a rapidly expanding and broad topic, we also suggest the reading of [2], where a force nonlinear observer for aerial vehicles is proposed, of [3], where an IDA-PBC controller is used for modulating the physical interaction of aerial robots, of [4], [5] where fully actuated platforms for full wrench exertion are presented, of [6]–[8] where the capabilities of exerting forces with a tool are studied, and finally of [9]–[11] where aerial manipulators with elastic-joint arms are modeled and their controllability properties discovered. The reader interested in the analysis and control of tethered aerial vehicles (another form of physical interaction) is also referred to [12], where flatness, controllability and observability is studied, to [13] where the case of a moving base is thoroughly analyzed, to [14] where real experiments for tethered landing on sloped surfaces is shown, and to [15], [16] where the case of multiple tethered vehicles is investigated.

I. EXPLICIT COMPUTATION OF THE NOMINAL STATES AND THE INPUTS

In Sec. IV of [1] we describe how the nominal states and the inputs of a PAM can be computed as sole functions of the flat outputs and their derivatives up to the fourth order. These flat outputs are given in the Fact. 1 of [1] as $y = [p^T_{0,x}, \theta^T] \in \mathbb{R}^{n+2}$, where the generalized coordinates of the PAM in 2D is chosen as $q_2 = [p^T_{0,x}, \theta_0, \theta^T] \in \mathbb{R}^{3+n}$ and the control inputs of the system are $u_2 = [u_x, u_r, \tau^T] \in \mathbb{R}^{2+n}$. This means that while the CoM positions of the VTOL and the absolute orientations of the links are direct functions of the flat outputs, $y$, the computations of $\theta_0, \dot{\theta}_0$ (flying base pitch and its time derivatives), and $u_x, u_r, \tau_{xw}$ as functions of $y, \dot{y}, \ddot{y}, \dot{\theta}_0$, $\ddot{\theta}_0$ are implicit.

Let us start with the computations of $\theta_0, \dot{\theta}_0$, and $u_x$. As it is given in (10) of the paper, these values are functions of the CoM position of the overall system, $\mathbf{p}_c$ and its derivative $\dot{\mathbf{p}}_c$ (actually we also need to compute $\ddot{\mathbf{p}}_c$, because $\dot{\theta}_0$ depends on it, and $\dot{\theta}_0$ is needed for computing the torque $u_r$). Let us then compute the derivatives of $\mathbf{p}_c$. Taking the time derivatives of (11) of the paper (from second up to the fourth order), we have:

$$\dot{\mathbf{p}}_c = \frac{1}{m_a} \left( m_0 \mathbf{p}_{0,x} + \sum_{j=1}^{m} \left( \sum_{i=1}^{n_j} \mathbf{m}_{i,j} \dot{\mathbf{p}}_{i,j} + m_{m,j} \mathbf{p}_{m,j} \right) \right) \tag{1}$$

$$\ddot{\mathbf{p}}_c = \frac{1}{m_a} \left( m_0 \mathbf{p}_{0,x} + \sum_{j=1}^{m} \left( \sum_{i=1}^{n_j} \mathbf{m}_{i,j} \ddot{\mathbf{p}}_{i,j} + m_{m,j} \dddot{\mathbf{p}}_{m,j} \right) \right)$$

$$\dddot{\mathbf{p}}_c = \frac{1}{m_a} \left( m_0 \mathbf{p}_{0,x} + \sum_{j=1}^{m} \left( \sum_{i=1}^{n_j} \mathbf{m}_{i,j} \dddot{\mathbf{p}}_{i,j} + m_{m,j} \ddddot{\mathbf{p}}_{m,j} \right) \right).$$

Now, for $\mathbf{p}_c, \dot{\mathbf{p}}_c, \ddot{\mathbf{p}}_c$ to be functions of the flat outputs only we must show that $\mathbf{p}_{i,j}, \mathbf{p}_{m,j}, \mathbf{p}_{i,j}, \mathbf{p}_{i,j}$ can be computed using only the flat outputs and their derivatives.
TABLE I: A summarizing table of the differential flatness of PAMs in 2D. Different outputs of the system are given on the top. The nominal states and inputs as implicit functions of the flat outputs and their derivatives up to the fourth order are provided. Note that for the $\nu^\mu$-th element of the system, $p_{\nu^\mu}$ and $p_{m^\nu^\mu}$ are the (time varying) individual link and motor CoM positions represented in $\mathcal{F}_W$, respectively.

For the $\nu$-th motor of the $\mu$-th manipulator ($\nu^\mu$-th motor of the system) it is:

$$\begin{align*}
\tilde{p}_{m^\nu^\mu}(y, y, \dot{y}) &= \tilde{p}_{0^\mu} + \gamma_1(q_{\nu^\mu}, q_{\nu^\mu}, \dot{q}_{\nu^\mu}) \\
\tilde{p}_{m^\nu^\mu}(y, y, \ddot{y}, \dot{y}) &= \tilde{p}_{0^\mu} + \gamma_2(q_{\nu^\mu}, q_{\nu^\mu}, q_{\nu^\mu}, \dot{q}_{\nu^\mu}) \\
\tilde{p}_{m^\nu^\mu}(y, y, \dddot{y}, y, \dot{y}) &= \tilde{p}_{0^\mu} + \gamma_3(q_{\nu^\mu}, q_{\nu^\mu}, q_{\nu^\mu}, q_{\nu^\mu}, \dot{q}_{\nu^\mu})
\end{align*}$$

where

$$\begin{align*}
\gamma_1 &= \sum_{i=1}^{\nu^{\mu-1}} (R_{0^\mu_i} \cdot d_{\nu^\mu_i} \cdot \dot{\theta}_{0^\mu_i} - R_{0^\mu_i} \cdot d_{\nu^\mu_i} \cdot \dot{\theta}_{0^\mu_i}^2) \\
\gamma_2 &= \sum_{i=1}^{\nu^{\mu-1}} (R_{0^\mu_i} \cdot d_{\nu^\mu_i} \cdot \ddot{\theta}_{0^\mu_i} - 3R_{0^\mu_i} \cdot d_{\nu^\mu_i} \cdot \dot{\theta}_{0^\mu_i} \cdot \dot{\theta}_{0^\mu_i} + R_{0^\mu_i} \cdot d_{\nu^\mu_i} \cdot \dot{\theta}_{0^\mu_i}^3) \\
\gamma_3 &= \sum_{i=1}^{\nu^{\mu-1}} (R_{0^\mu_i} \cdot d_{\nu^\mu_i} \cdot \dddot{\theta}_{0^\mu_i} - 4R_{0^\mu_i} \cdot d_{\nu^\mu_i} \cdot \dot{\theta}_{0^\mu_i} \cdot \dddot{\theta}_{0^\mu_i} + 3R_{0^\mu_i} \cdot d_{\nu^\mu_i} \cdot \dddot{\theta}_{0^\mu_i}^2)
\end{align*}$$

This gives the linear accelerations of the $\nu^\mu$-th motor CoM in the $x_W - z_W$ plane, and its first and second derivatives w.r.t. time as sole functions of the flat outputs. For the $\nu^\mu$-th link it is:

$$\begin{align*}
\ddot{p}_{\nu^\mu} &= \tilde{p}_{m^\nu^\mu} + \tilde{R}_{0^\nu^\mu} \cdot d_{\nu^\mu} \cdot \dddot{\theta}_{0^\nu^\mu} - \tilde{R}_{0^\nu^\mu} \cdot d_{\nu^\mu} \cdot \dot{\theta}_{0^\nu^\mu}^2 \\
\dddot{p}_{\nu^\mu} &= \tilde{p}_{m^\nu^\mu} + \tilde{R}_{0^\nu^\mu} \cdot d_{\nu^\mu} \cdot \ddot{\theta}_{0^\nu^\mu} - 3\tilde{R}_{0^\nu^\mu} \cdot d_{\nu^\mu} \cdot \dot{\theta}_{0^\nu^\mu} \cdot \dddot{\theta}_{0^\nu^\mu} \\
\dddot{p}_{\nu^\mu} &= \tilde{p}_{m^\nu^\mu} + \tilde{R}_{0^\nu^\mu} \cdot d_{\nu^\mu} \cdot \dot{\theta}_{0^\nu^\mu} - 6\tilde{R}_{0^\nu^\mu} \cdot d_{\nu^\mu} \cdot \ddot{\theta}_{0^\nu^\mu} ^2 + \tilde{R}_{0^\nu^\mu} \cdot d_{\nu^\mu} \cdot \dddot{\theta}_{0^\nu^\mu}.
\end{align*}$$

Using (2) in (3) we compute the linear acceleration of the $\nu^\mu$-th link CoM in the $x_W - z_W$ plane, and its first and second time derivatives as sole functions of the flat outputs. Substituting both (2) and (3) in (1) we do the same for the overall system CoM accelerations and its derivatives. Finally applying what derived so far to (10) in [1] and taking into account (9) in [1], we compute the nominal values of $\theta_0(y, y, \dot{y}), \ddot{\theta}_0(y, y, \ddot{y}),$ and $\dddot{\theta}_0(y, y, \dddot{y})$ as sole functions of the flat outputs and their finite number of derivatives.

The computation of $u_r, \tau^\nu, \nu^\mu,$ and hence $\tau$ are explained in [1]. Any further computation necessary to create the explicit mapping from the flat outputs to these inputs are already given in [10] and in [17]. A summary of these results is given in Table I.

II. Extensive Simulations

In [1], the experimental validation of the proposed controller using differential flatness is presented. The experimental setup consists of a Quadrotor VTOL and a two DoF manipulating arm (see Fig. 3 of the paper). The arm is attached to the quadrotor, with a 6 [cm] of offset along the $z$-axis of $\mathcal{F}_0$ between the position of CoM of the VTOL ($P_0$) and the first joint. Despite this not being conform with the protocentricity assumption of the design, performance degradation effects on the controller were not particularly noticeable.

This leads us to the question; what is the effect of the non-protocentricity, i.e., what if in the real system there is a generic offset between the CoM of the VTOL and the first joint of which the controller is not aware? An attempt to answer this
question is done in a simulation environment, using a physics-based model of our experimental setup built in Sim-Mechanics of Matlab–Simulink (see Fig. 1 for details).

Define \( \mathbf{d}_0 = [d_{0x}, d_{0y}]^T \in \mathbb{R}^2 \) as the relative position between the attachment point of the first motor of the manipulating arm and \( \mathcal{P}_0 \) expressed in \( \mathcal{F}_0 \). We run a set of simulations testing the tracking of composite trajectories in the flat output space using the proposed controller, while varying the value \( \mathbf{d}_0 \) between the ranges

\[
\begin{align*}
d_{0x}^{\text{min}} &= 0 \,[\text{m}], & d_{0x}^{\text{max}} &= 0.4 \,[\text{m}], \\
d_{0y}^{\text{min}} &= -0.4 \,[\text{m}], & d_{0y}^{\text{max}} &= 0.4 \,[\text{m}]
\end{align*}
\]

with a discrete sampling of 15 points. Thrust and torque limits of the quadrotor VTOL are also imposed (VTOL torque limits are chosen as \(+/- 4.55 \,[\text{Nm}]\); the minimum thrust is set to \(1 \,[\text{N}]\) while maximum thrust is set to \(21 \,[\text{N}]\)). The results are given in Fig. 2.

The initial condition is set far from the start of the desired trajectory in order to test the transient behavior. However from Fig. 2 it is clear that all the tracked outputs converge to their desired ones quickly. It is clearly shown that for the protocentric design, \( \mathbf{d}_0 = 0 \), the controller lets the system track the desired outputs almost perfectly. When the system is not protocentric (\( \mathbf{d}_0 \neq 0 \)), then the outputs of the system start to diverge from the desired ones, which in turn forces the system actuators towards their limits. This is due to the dynamic effect of the non-protocentric design which is not considered in the controller, as well as the actuation limits that do not allow the feedback term of the controller to steer the system quickly to the desired values (see the last row of Fig. 1 for the control inputs). Nonetheless, the feedback term is still able to keep the real behavior close enough to the desired one, showing a good robustness, as explained in Sec. III of [1].

In the future, we plan to investigate the non-protocentric aerial manipulators and their control more in detail.

III. DETAILED EXPERIMENTAL RESULTS

In this section, and in particular in Figs 3, 4 and 5, we integrate the plots related to the experimental results presented in [1] with some additional informations, i.e., , the full state of the VTOL (position and orientation) and its thrust. The considerations and comments of the figures are the same of the ones in the paper.

REFERENCES


Fig. 2: Simulations for an aerial manipulator which is not protocentric. The controller assumes the protocentric design as presented in [1]. The coordinates in the plane of interest ($x_W \times z_W$) are shown in the first two columns, and all the other coordinates are gathered in the last row of the second column. The control inputs of the VTOL are given in the last row. The desired trajectories are depicted with black dashed curves. The blue solid curve used for the coordinates of the PAM when the model is protocentric ($d_0 = 0$). The red color is used to depict the evolution of the system with different non-protocentric designs ($d_0 \neq 0$): the maximum and the minimum values of all coordinates out of 15 simulations are plotted, and the space between them is filled in pink. A similar color code is used also for the control inputs. In the last sub-figures of the second and third rows, a different color code is used, but the same way of plotting is applied (notice that all values are almost zero). Notice the transient phase at the beginning of the simulations.


Fig. 5: Experimental results for trajectory (c).