Decoupled Dynamic Modeling by Decomposing the Cross-Coupled Dynamics and Tube-Based LPV-MPC Control Scheme for Aerial Manipulation

A. Eskandarpour
M. Soltanshah
K. Gupta
Faculty of Engineering Science, Simon Fraser University, 8888 University Dr, Burnaby, BC, Canada

M. Mehrandezh
Faculty of Engineering, University of Regina, 3737 Wascana Pkwy, Regina, SK, Canada

***Abstract*—In this paper, we provide a decoupled method for modeling and controlling unmanned aerial manipulators (UAMs) to improve trajectory tracking by addressing challenges arising from the cross-coupled dynamics of the UAV and manipulator. Existing methods are: 1) estimating reaction torque and force (ERTF); and 2) modeling the cross-coupled dynamics via changing the UAV's inertia matrix and center of mass (CoM), but they suffer from high tracking errors or ineffective handling of inertia parameter derivatives during fast trajectory tracking. In this work, we systematically decompose and represent the reaction force and torque in UAM states, facilitating effective adjustments to the dynamic model for both the UAV and manipulator while addressing all cross-coupled effects. This decoupling enables the development of individual controllers for both the UAV and the manipulator. Additionally, We partition the translational and rotational components of the UAV's dynamics, and then we design a linear MPC controller for the translational dynamics, aiming to achieve a desired rotational trajectory by expressing reaction forces in relation to the UAV's force vector. We develop linear parameter-varying (LPV) dynamic models for both the UAV's rotational dynamics and the manipulator, followed by designing tube-based LPV-MPC controllers for them, considering varying parameters as uncertainties. This approach ensures system feasibility, stability, and robustness. This is achieved by computing robust positively invariant (RPI) terminal sets and incorporating state and control input constraints through reachable sets over prediction horizons. The proposed approach's performance is validated through simulations. Furthermore, we show that our approach outperforms the ERTF in trajectory tracking.**

***Index Terms*— Cross-coupled dynamics, Decoupled Control Design, Trajectory Tracking, Tube-Based LPV-MPC, Unmanned Aerial Manipulator**

I. INTRODUCTION

Today's Unmanned Aerial Vehicles (UAVs), with their autonomy and remote control capabilities, facilitate a wide range of difficult tasks. Equipping the UAV with a manipulator, alternatively known as an Unmanned Aerial Manipulator (UAM), enables it to execute manipulation tasks, referred to as aerial manipulation. Examples of such tasks carried out by a UAM are grasping [1], valve turning [2], delivery [3], door opening [4], interacting [5], and so on. However, designing an efficient and robust controller for UAMs is challenging due to their very complicated and coupled dynamics, i.e., the manipulator's motion impacts the UAV's motion, and vice versa, often referred to as coupled dynamics. It must be noted that we use the words manipulator and arm interchangeably throughout this paper.

On a general note, control of UAMs may be investigated from several viewpoints depending on how they are utilized, such as position control [6], force and impedance control [7], vision-based control [8], teleoperation, and human-UAM interaction control [9]. In this paper, we aim to design a position control, i.e., steering the UAV's rotational and translational dynamics and the manipulator's dynamics along a specific trajectory. A key challenge in UAM's trajectory tracking control design is to include the complete dynamic interaction between the manipulator and the UAV due to their respective motions (cross-coupled dynamics).

Moreover, the underactuation of fixed-rotor UAVs, such as quadrotors or hexarotors, as a mechanical system may make it difficult to completely eliminate cross-coupled effects and follow an arbitrary trajectory. More specifically, the reaction force vector exerted by the manipulator motion will typically have components along all three axes in body frame: x, y, and z. In a fixed-rotor UAV, on the other hand, the net actuated force vector has only one component, which is along the z axis. So, it's challenging for the UAV to entirely cancel out the reaction force, which might result in undesirable closed-loop performance and inaccurate trajectory tracking for the manipulator and UAV. Therefore, for accurate trajectory tracking, it is crucial to have a dynamic model for UAM that is complete, i.e., it incorporates all cross-coupled dynamics.

There are two main categories of UAM control scheme: decoupled control design [10] and coupled [11]. Whereas the latter refers to employing a single controller for the entire UAM system, the former refers to using individual controllers for both the UAV and the manipulator. The coupled method controls them as a unified

(Corresponding author: A. Eskandarpour)

A. Eskandarpour, M. Soltanshah and K. Gupta are with Engineering Science department, Simon Fraser University, 8888 University Dr, Burnaby, BC, Canada, (e-mail: eskandar@sfu.ca; mohammad_soltanshah@sfu.ca;kamal@sfu.ca). M. Mehrandezh is with the Engineering department, University of Regina, 3737 Wascana Pkwy, Regina, SK, Canada, (e-mail: mehran.mehrandezh@uregina.ca).

0018-9251 © 2022 IEEE

platform by considering their highly nonlinear dynamics. The common control practices are robust PID [12], LQR [13], [14], backstepping [15], sliding mode [16], and MPC controllers [17]. As our emphasis is on designing a real-time optimal controller, we should keep in mind that the main drawback of coupled approaches is that developing a constrained model-based controller would experience significant computational costs [18]. For instance, in [17], they succeeded in only implementing an open-loop nonlinear MPC (NMPC) controller for generating a reference trajectory. Instead, the generated reference trajectory is followed by feedback linearization and PID controllers at the lower level.

Decoupled control design generally assumes a minimal dynamic impact of manipulators on the UAV base targeted for light-weight manipulators. However, it can cause significant tracking errors at high speeds/accelerations [19]. As suggested in [19], there are some intermediate approaches that can be categorised as partially coupled [20], [21], in which case the independent controllers for the UAV and manipulator exchange information with each other to deal with the dynamic couplings. In the following, we will consider only decoupled and partially coupled approaches. Furthermore, concerning the modeling of a UAM, common methods in the existing literature include: 1) estimating reaction torque and force (ERTF), and 2) modeling cross-coupled dynamics by adjusting the UAV's inertia matrix and its center of mass (CoM).

In the first method, some researchers modeled the cross-coupled dynamics as reaction torques and forces and attempted to compensate for them. To do this, the common approaches are using torque and force sensors [22], [23], recursive Newton-Euler [24], [25], and force and torque observers [20], [26]. After estimating the reaction torque and force, an appropriate control strategy is designed for compensation. PID controllers are widely used to deal with this problem [27], [28]. However, [29] mentioned that the PD controller's performance is not desirable for the attitude dynamics of the UAM. They provided a robust nonlinear quaternion-based backstepping controller in the presence of disturbance, with favorable performance over that by a PD controller. In addition, [21] provides a multi-layer PID-based controller, coupled with estimating the generalized reaction moments and forces exerted on the UAV via an estimator, to counterbalance the effects of the manipulator's motion. In [30], an efficiently computed and robust controller is employed to guide the end-effector following a 6-DOF reference trajectory, utilizing the fewest possible actuators through the solution of an inverse kinematics problem. A UAM made up of a 6-DoF quadrotor and a manipulator with two links is built using an adaptive intelligent fuzzy controller in [31]. They demonstrated the controller's ability to track a trajectory while maintaining stability with comparable performance with the feedback linearization techniques. In [32], the kinematic effects on a robotic arm are taken into account without considering the dynamic coupling effects on the UAM.

Moreover, [33] proposes a multi-stage MPC controller by defining two pairs of input-output systems and assuming that the thrust forces of rotors do not directly affect the pose of the manipulator and also that the capability of the UAM in tracking a trajectory is not directly affected by the manipulator. In [34], a cascaded controller is designed for a UAM. At the first layer, functioning as the motion controller, they provide an artificial neural network (ANN)-based adaptive controller and a PID controller for the UAV and the arm, respectively. For the second layer, an MPC controller is designed to work as a coordinate planner. Optimal ANN-based controllers are designed in [35] and [36], and then the proposed approaches are extended and provided in [25]. In this paper, a partitioned controller, made of a PID and MPC, is devised for the trajectory tracking problem and the object retrieval task. Also, the reacting torques and forces that the arm applies to the UAV are estimated using recursive Newton-Euler (RNE) equations.

In [24], after finding a dynamic model for the UAM using RNE equations, the feedback linearization technique is used to compensate for the reaction torque and force imposed on the UAV in a trajectory tracking problem. Although the aforementioned approaches can provide favourable performance, they might fail when there is a high demand for the UAM to track a fast trajectory (high-speed reference trajectories are intended to be tracked simultaneously by the arm and the UAV), while still maintaining stability. Nevertheless, by replacing the coupling effects merely by the value of reaction torques and forces, one would lose coupling information between UAM's states. As we will see in the rest of this paper, we can formulate reaction torques or forces based on UAV and manipulator states, which enables us to include the coupling information for the respective UAV and arm controller designs.

Using the second method, the cross-coupled dynamics are represented by the system inertia matrix and the changing CoM. Essentially, a UAM can be viewed as an aerial vehicle whose mass distribution may alter as the manipulator moves. The movement of the manipulator, not its speed or acceleration, is reflected in changes to the UAV's CoM and the inertia matrix. Gain scheduling approaches have been used to cope with this problem [37], [38]. In [37], a gain scheduling approach along with adaptive control is used for a UAM system. The simulation results are used to show that their approach could stabilize the UAM when the attitude dynamic is controlled by a PID controller. In [38], by assuming that aerial manipulation is functioning near hover flight, the system is stabilized using a gain schedule PID. In [39], first, the cross-coupled dynamics in the UAM are modeled as changing inertial parameters that depend on the system states, then providing a H_∞ controller to preserve the system's feasibility and stability. For grasping a moving object using a UAM constructed from a 7-DoF manipulator and a hexarotor, a decoupled controller structure is provided in [40]. They approximate the coupled dynamics via changing merely the UAV's CoM.

In [41], a hierarchical nested NMPC is provided to make the UAM follow a trajectory by improving the arm movement and aligning its center of gravity (CoG) along the gravity vector of the UAM. In [42], the controller relies on an online CoG estimator to stabilize the UAM while it performs position tracking. These methods performed well in tracking a quasi-static trajectory, but since they don't take the derivative of UAM states into account when modifying the CoM and inertia matrices, they may perform poorly in a rapid trajectory tracking scenario. In this regard, a UAM's dynamic model based on varying CoM is provided in [43]. The provided dynamic model includes the derivatives of the inertia parameters, which enables it to outperform the aforementioned approaches and makes it more suitable for a quicker trajectory tracking scenario. Although this approach can account for the disturbance arising from the arm's movement affecting the UAV, it may not capture all the cross-coupled dynamics present in the UAM.

This paper addresses both the UAM's modeling and control design. Regarding modeling, we undertake three main steps. Firstly, we develop a decoupled dynamic model for UAMs, providing separate dynamic models for the UAV and manipulator while addressing their cross-coupled dynamics. This involves characterizing cross-coupled effects as reaction torques and forces, covering all possible coupling states and their associated derivatives. Secondly, we reformulate the UAV's dynamic model by decomposing and redefining the force and torque reaction equations in terms of the UAV's state factors. This reformulation yields linear parameter-varying (LPV) dynamic equations for the UAV, integrating all coupling information. Furthermore, in the case of an underactuated UAV, we separate translational and rotational dynamics, with attitude reference trajectories (Euler angles and angular body rates) derived from translational dynamics, which should then be followed by rotational dynamics. In the simulation results, we will demonstrate that our method performs much better for trajectory tracking than the ERTF.

Lastly, we derive a dynamic model for the manipulator, considering the dynamic influence of the UAV on the manipulator, an aspect that has not been adequately addressed in the literature. This step is essential to ensuring optimal performance in the context of end-effector trajectory tracking, particularly when the UAV experiences rapid accelerations, like grasping a moving target from the air [44]. Also, we provide an LPV model for the manipulator, which includes all coupling states and their corresponding derivatives.

For control purposes, we introduce a cascaded decoupled MPC controller for the UAM to address rapid trajectory tracking. In handling the UAV's translational dynamics, a constrained linear MPC (LMPC) controller is devised to trace a desired trajectory in 3D space. Additionally, leveraging the expressed rotational nonlinear dynamics within a state-space LPV dynamic representation and assuming bounded variations in time-varying parameters, we formulate a tube-based LPV-MPC control to manage uncertainty arising from cross-coupled dynamics. Moreover, utilizing the obtained LPV model for the manipulator, we propose an LPV-MPC control scheme to enable the manipulator to pursue a reference trajectory while considering the cross-coupled effects imposed by the UAV. In summary, this paper's primary contributions are listed below:

- A decoupled dynamic model that incorporates all cross-coupled dynamics is provided for the UAM by decomposing reaction torques and forces in terms of UAM states.
- The obtained dynamics are formulated in LPV form, which preserves the nonlinear behaviour of the system, while we can use powerful techniques in control of affine systems to design a robust controller enabling it to fast-follow a trajectory.
- Following the representation of the cross-coupled dynamics as an element of uncertainty within the system, for the UAV's rotational dynamics, we propose a tube-based MPC control scheme.
- With the provided manipulator's LPV dynamic, which includes the UAV's effects, we have formulated a tube-based MPC controller for the manipulator to have effective trajectory tracking while the UAV accelerates.

We organise the remainder of the paper this way: A decoupled dynamic equations for a UAM is formulated in section II, which includes all cross-coupled dynamics. We propose the suggested control scheme for the UAM in section III, and the stability and robustness of the proposed UAM's control scheme are discussed in section IV. Lastly, we assess the provided control approach's performance in two trajectory following scenarios for the UAM via simulation in section V.

II. UAM'S DYNAMIC MODELING

A UAM's dynamic model is highly complex, especially when the robotic arm's degrees of freedom (DoF) increase. We now present a general dynamic model for a UAM that consists of a n_l-link manipulator and an underactuated multi-rotor UAV. As we intend to propose a decoupled control scheme for the UAV and the robotic arm, we will provide a dynamic model for each of them while taking into account all the cross-coupled dynamics between them.

Let us assume that $\boldsymbol{I} = \{x_I, y_I, z_I\}$ denotes the inertia frame and $\boldsymbol{B} = \{x_b, y_b, z_b\}$ is the body frame associated with the UAV. For a robotic arm with n_l links, the associated coordinate frame for ith arm's joint is $\boldsymbol{M}_i = \{x_{M_i}, y_{M_i}, z_{M_i}\}$, where $i = 1, ..., n_l$, and the joint angle vector is $\boldsymbol{\Theta} = [q_1, ..., q_{n_l}]^T$. Let us also represent the CoM coordinates of the UAV in space by $\boldsymbol{\zeta} = [x, y, z]^T$, and the Euler angles of the body in the in-

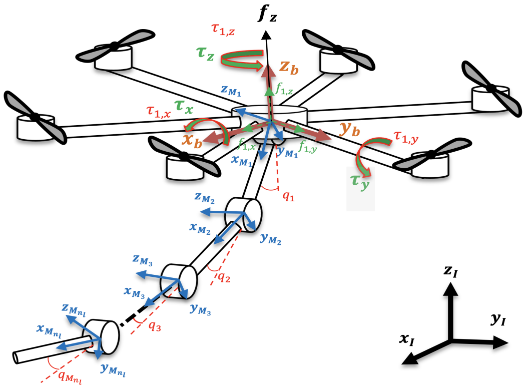

Fig. 1: UAM's Scheme

ertial frame by $\boldsymbol{\eta} = [\phi, \theta, \psi]^T$. Moreover, $\boldsymbol{V} = [u, v, w]^T$ and $\boldsymbol{\omega} = [p, q, r]^T$, respectively, represent the angular and linear velocities of the UAV in the body frame (Fig. 1). Assuming that $s_{(\cdot)}$ and $c_{(\cdot)}$ represent $\sin(\cdot)$ and $\cos(\cdot)$, respectively, then ${}^{I}\boldsymbol{R}_B$ is the rotation matrix [45], which is given by:

$$
{}^{I}\boldsymbol{R}_B = \begin{bmatrix} c_\psi c_\theta & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & -s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \quad (1)
$$

Moreover, $\boldsymbol{\omega} = {}^{B}\boldsymbol{T}_I \; \dot{\boldsymbol{\eta}}$ is the equation that converts the Euler rates $\dot{\boldsymbol{\eta}}$ to the angular velocity $\boldsymbol{\omega}$ [46], where the transformation matrix ${}^{B}\boldsymbol{T}_I$ can be represented by:

$$
{}^{B}\boldsymbol{T}_I = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix}. \quad (2)
$$

The generated force and torques by the UAM are depicted in Fig. 1. The UAV's force vector and torque vector in the body frame are denoted by $\boldsymbol{f}_{uav} = [0, 0, f_z]^T$ and $\boldsymbol{\tau}_{uav} = [\tau_x, \tau_y, \tau_z]^T$, respectively. Moreover, $\boldsymbol{f}_i = [f_{i,x}, f_{i,y}, f_{i,z}]^T$ and $\boldsymbol{\tau}_i = [\tau_{i,x}, \tau_{i,y}, \tau_{i,z}]^T$ represent the torque and force vector exerted on link ith by link $i-1$th, respectively, where $i = 1, ..., n_l$. For control purposes, the first link's force and torque vector are considered as the robotic arm's reaction force and torque exerted on the UAV, i.e., $\boldsymbol{f}_{arm}^{rea} \triangleq -\boldsymbol{f}_1$ and $\boldsymbol{\tau}_{arm}^{rea} \triangleq -\boldsymbol{\tau}_1$.

A. UAV's Dynamic Modeling

The UAV's equations of motion, by including the exerted forces and torques by the manipulator, are given by:

$$
m\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times m\boldsymbol{V} = \boldsymbol{f}_{arm}^{rea} + \boldsymbol{f}_{uav} \quad (3a)
$$

$$
\boldsymbol{I}_t\dot{\boldsymbol{\omega}} \;\; + \boldsymbol{\omega} \times \boldsymbol{I}_t\boldsymbol{\omega} = \boldsymbol{\tau}_{arm}^{rea} + \boldsymbol{\tau}_{uav}, \quad (3b)
$$

where $\boldsymbol{I}_t \in \mathcal{R}^{3\times3}$ represents the UAV's inertia tensor matrix (subscript t denotes tensor), and m is UAM's mass. Considering that the centrifugal force $\boldsymbol{\omega} \times m\boldsymbol{V}$ is nullified in the inertial frame [47], the UAV's translational motion equation (3a) represented in the inertial frame is:

$$
\ddot{\boldsymbol{\zeta}} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{{}^{I}\boldsymbol{R}_B}{m}(\boldsymbol{f}_{arm}^{rea} + \boldsymbol{f}_{uav}). \quad (4)
$$

where $g = 9.81\frac{m}{s^2}$. Using translational and rotational dynamic equations (4) and (3b), the corresponding state-space models are:

$$
\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m}\left({}^{I}\boldsymbol{R}_B \begin{bmatrix} f_{1,x} \\ f_{1,y} \\ f_{1,z} + f_z \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \right), \quad (5a)
$$

$$
\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \boldsymbol{I}_t^{-1}\left(\boldsymbol{M}_c \begin{bmatrix} pq \\ qr \\ pr \end{bmatrix} + \boldsymbol{M}_s \begin{bmatrix} p^2 \\ q^2 \\ r^2 \end{bmatrix} + \begin{bmatrix} \tau_x + \tau_{1,x} \\ \tau_y + \tau_{1,y} \\ \tau_z + \tau_{1,z} \end{bmatrix} \right), \quad (5b)
$$

where equation (5b) is derived by expanding equation (3b) and factoring out the products and square terms of angular body rates, such that the elements of $\boldsymbol{M}_c \in \mathcal{R}^{3\times3}$ and $\boldsymbol{M}_s \in \mathcal{R}^{3\times3}$ consist of components from $\boldsymbol{I}_t$.

B. Manipulator's Dynamic Modeling

For the manipulator, we derive a state-space dynamic model through the iterative Newton-Euler dynamics algorithm [48]. In addition to using the dynamic model for manipulator control, we will utilize it to represent the cross-coupled dynamics between the manipulator and the UAV. These coupling dynamics will then be used to modify the UAV's dynamic model. It's important to note that achieving this objective may be facilitated by expressing the manipulator's equations of motion relative to the UAV's body frame, especially when we need to distinguish the UAV's dynamic states from the manipulator's dynamic states.

Assume that the origins of the first joint coordinate frame $\boldsymbol{M}_1$ and the UAV's body frame $\boldsymbol{B}$ coincide. Using the outward equations from link 1 to link n_l, the velocities and accelerations associated with each link are calculated, and then using the inward equations from link n_l to link 1, the torque and force vectors acting on link ith [48]:

-Newton-Euler outward equations $(i : 0 \rightarrow (n_l - 1))$:

$$
\boldsymbol{\omega}_{i+1} = {}^{i+1}_{\;\;i}\boldsymbol{R}\boldsymbol{\omega}_i + \dot{q}_{i+1}\hat{\boldsymbol{Z}}_{i+1}, \quad (6a)
$$

$$
\dot{\boldsymbol{\omega}}_{i+1} = {}^{i+1}_{\;\;i}\boldsymbol{R}\dot{\boldsymbol{\omega}}_i + {}^{i+1}_{\;\;i}\boldsymbol{R}\boldsymbol{\omega}_i \times \dot{q}_{i+1}\hat{\boldsymbol{Z}}_{i+1} + \ddot{q}_{i+1}\hat{\boldsymbol{Z}}_{i+1}, \quad (6b)
$$

$$
\dot{\boldsymbol{v}}_{i+1} = {}^{i+1}_{\;\;i}\boldsymbol{R}(\dot{\boldsymbol{\omega}}_i \times {}^{i}\boldsymbol{P}_{i+1} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times {}^{i}\boldsymbol{P}_{i+1}) + \dot{\boldsymbol{v}}_i), \quad (6c)
$$

$$
\dot{\boldsymbol{v}}_{C_{i+1}} = \dot{\boldsymbol{\omega}}_{i+1} \times \boldsymbol{P}_{C_{i+1}} + \boldsymbol{\omega}_{i+1} \times (\boldsymbol{\omega}_{i+1} \times \boldsymbol{P}_{C_{i+1}}) + \dot{\boldsymbol{v}}_{i+1}, \quad (6d)
$$

$$
\boldsymbol{F}_{i+1} = m_{i+1}\dot{\boldsymbol{v}}_{C_{i+1}}, \quad (6e)
$$

$$
\boldsymbol{N}_{i+1} = \boldsymbol{I}_{i+1}\dot{\boldsymbol{\omega}}_{i+1} + \boldsymbol{\omega}_{i+1} \times \boldsymbol{I}_{i+1}\boldsymbol{\omega}_{i+1}, \quad (6f)
$$

-Newton-Euler inward equations $(i : n_l \rightarrow 1)$:

$$
\boldsymbol{f}_i = {}_{i+1}^{\;\;i}\boldsymbol{R}\boldsymbol{f}_{i+1} + \boldsymbol{F}_i, \quad (7a)
$$

$$
\boldsymbol{\tau}_i = \boldsymbol{N}_i + {}_{i+1}^{\;\;i}\boldsymbol{R}\boldsymbol{\tau}_{i+1} + \boldsymbol{P}_{C_i} \times \boldsymbol{F}_i + {}^{i}\boldsymbol{P}_{i+1} \times {}_{i+1}^{\;\;i}\boldsymbol{R}\boldsymbol{f}_{i+1}, \quad (7b)
$$

$$
\tau_{i,y} = \boldsymbol{\tau}_i^t\hat{\boldsymbol{Z}}_i, \quad (7c)
$$

where $\boldsymbol{\omega}_i$ is the rotational velocity of the ith link's CoM and we assume that $\boldsymbol{\omega}_0 = \boldsymbol{\omega}$, $\dot{\boldsymbol{v}}_i$ is the linear acceleration of the ith frame origin and $\dot{\boldsymbol{v}}_0 = [0, 0, \frac{f_z}{m}]^T$, $\dot{\boldsymbol{v}}_{C_i}$ is the linear acceleration of the ith link's CoM. Also, ${}^{i+1}_{i}\boldsymbol{R}$ denotes the rotation matrix from the ith frame to the $(i+1)$th frame, $\hat{\boldsymbol{Z}}$ is the joint's axis of rotation of the associated frame, ${}^{i}\boldsymbol{P}_{i+1}$ denotes the $(i+1)$th frame's origin position relative to the ith frame's origin, $\boldsymbol{P}_{C_i}$ is the position of the ith link's CoM, and $\boldsymbol{I}_i$ is the inertia matrix associated with the ith link, and m_i is the ith link's mass. Moreover, $\boldsymbol{F}_i$ denotes the inertial force and $\boldsymbol{N}_i$ denotes the torque acting on the ith link's CoM. Also, the ith joint's torque $\tau_{i,y}$ can be found using the applied net force $\boldsymbol{f}_i$ and net torque $\boldsymbol{\tau}_i$ on the link.

Having obtained equation (7c) for each joint, the manipulator's state-space dynamic model can be represented as follows:

$$\mathcal{M}(\boldsymbol{\Theta})\ddot{\boldsymbol{\Theta}} + \mathcal{C}(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \boldsymbol{\omega})\dot{\boldsymbol{\Theta}} + \mathcal{G}(\boldsymbol{\Theta}, \boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \boldsymbol{f}_{uav}) = \boldsymbol{\tau}_{arm}, \tag{8}$$

where the torque vector is denoted by $\boldsymbol{\tau}_{arm} = [\tau_{1,y}, ..., \tau_{n_l,y}]^T$, the mass matrix by $\mathcal{M}$, the centrifugal and Coriolis terms matrix by $\mathcal{C}$, and the gravity terms vector by $\mathcal{G}$. It's important to note that the Python symbolic library Sympy [49] is used to derive the equation (8). So, the manipulator's state-space dynamic equations are:

$$\begin{bmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_{n_l} \end{bmatrix} = \mathcal{M}^{-1}(\boldsymbol{\Theta}) \left(-\mathcal{C}(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \boldsymbol{\omega}) \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_{n_l} \end{bmatrix} - \mathcal{G}(\boldsymbol{\Theta}, \boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \boldsymbol{f}_{uav}) + \begin{bmatrix} \tau_{1,y} \\ \vdots \\ \tau_{n_l,y} \end{bmatrix} \right). \tag{9}$$

C. Modified UAV's Dynamic Model by Including Cross-coupled Dynamics

As one can expect, the manipulator's dynamic equations include the UAV's rotational states based on equation (9). Therefore, it is expected that these cross-coupled dynamics will appear in the reaction torque, which depends on the UAV and manipulator's states $(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \ddot{\boldsymbol{\Theta}}, \boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \boldsymbol{f}_{uav})$. Consequently, by formulating $\boldsymbol{\tau}^{rea}_{arm}$ such that the states of the UAV and the manipulator are distinct from one another, we are able to include the coupling dynamics into the UAV's rotational dynamic model without losing coupling information. Hence, we derive and formulate $\boldsymbol{\tau}^{rea}_{arm}$ as follows:

$$\boldsymbol{\tau}^{rea}_{arm} = \begin{bmatrix} \tau_{1,x} \\ \tau_{1,y} \\ \tau_{1,z} \end{bmatrix} = \boldsymbol{M}^{rea}_d \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \boldsymbol{M}^{rea}_c \begin{bmatrix} pq \\ qr \\ pr \end{bmatrix} + \boldsymbol{M}^{rea}_s \begin{bmatrix} p^2 \\ q^2 \\ r^2 \end{bmatrix} + \boldsymbol{M}^{rea}_l \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \bar{\boldsymbol{\tau}}^{rea}_{arm} \tag{10}$$

where, $\bar{\boldsymbol{\tau}}^{rea}_{arm} = [\bar{\tau}_{1,x}, \bar{\tau}_{1,y}, \bar{\tau}_{1,z}]^T$ represents the residual of $\boldsymbol{\tau}^{rea}_{arm}$, depending solely on the manipulator's states $(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \ddot{\boldsymbol{\Theta}})$. Additionally, $\boldsymbol{M}^{rea}_d$, $\boldsymbol{M}^{rea}_c$, $\boldsymbol{M}^{rea}_s$, and $\boldsymbol{M}^{rea}_l$ are 3×3 matrices, each a function of the manipulator's states $(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \ddot{\boldsymbol{\Theta}})$. It must be noted that the Python symbolic library Sympy [49] is used to obtain these matrices. Also, the UAV's translational dynamic can be modified by expressing the reaction force vector $\boldsymbol{f}^{rea}_{arm}$ in terms of f_z. In this regard, using (7a), we have:

$$\boldsymbol{f}^{rea}_{arm} = \begin{bmatrix} f_{1,x} \\ f_{1,y} \\ f_{1,z} \end{bmatrix} = \boldsymbol{M}^{rea}_f f_z + \bar{\boldsymbol{f}}^{rea}_{arm} \tag{11}$$

in which $\bar{\boldsymbol{f}}^{rea}_{arm} = [\bar{f}_{1,x}, \bar{f}_{1,y}, \bar{f}_{1,z}]^T$ denotes the residual of $\boldsymbol{f}^{rea}_{arm}$, which means it does not include any terms of f_z, and $\boldsymbol{M}^{rea}_f = [m_{1,f}, m_{2,f}, m_{3,f}]^T$ are functions of manipulator's states and UAV's rotational dynamic $(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \ddot{\boldsymbol{\Theta}}, \boldsymbol{\omega}, \dot{\boldsymbol{\omega}})$. Therefore, the UAV's dynamic (5) is modified using equations (10) and (11) as follows:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \frac{1}{m} \left({}^{I}\boldsymbol{R}_B \begin{bmatrix} m_{1,f} f_z + \bar{f}_{1,x} \\ m_{2,f} f_z + \bar{f}_{1,y} \\ (1 + m_{3,f}) f_z + \bar{f}_{1,z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \right), \tag{12a}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = (\boldsymbol{I}_t - \boldsymbol{M}^{rea}_d)^{-1} \left((\boldsymbol{M}_c + \boldsymbol{M}^{rea}_c) \begin{bmatrix} pq \\ qr \\ pr \end{bmatrix} + (\boldsymbol{M}_s + \boldsymbol{M}^{rea}_s) \begin{bmatrix} p^2 \\ q^2 \\ r^2 \end{bmatrix} + \boldsymbol{M}^{rea}_l \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} \tau_x + \bar{\tau}_{1,x} \\ \tau_y + \bar{\tau}_{1,y} \\ \tau_z + \bar{\tau}_{1,z} \end{bmatrix} \right), \tag{12b}$$

III. CONTROLLER DESIGN

In this section, the goal is to design two decoupled optimal controllers, one for the UAV and one for the arm for fast trajectory tracking by the UAM. The decoupled dynamic model provided in Section (II) enables us to design a fast decoupled controller that can effectively take care of cross-coupled dynamics of the manipulator and the UAV. Fig. 2 demonstrates the suggested control scheme for the UAM. A cascaded MPC controller is considered, given that the UAV's dynamics are divided into translational and rotational components. In addressing translational dynamics, we develop a LMPC controller for trajectory tracking, accounting for system constraints and external disturbances.

For the rotational dynamic system, the nonlinear dynamics (12b) is formulated in a state-space LPV dynamic form. Then, by assuming that there is an upper bound on the time-varying parameters and that their rates are limited, an uncertain dynamic model is obtained. Using this, a tube-based LPV-MPC controller is developed to deal with the reaction torque imposed on the UAV. For the manipulator, the dynamic model (9) is converted to a LPV model in order to consider the coupling dynamics that the UAV imposes on the manipulator. Then, we design a tube-based LPV-MPC controller to drive the manipulator toward the desired reference trajectory.

Because the UAV is underactuated, a translational reference trajectory is given, and then the rotational reference trajectory is obtained based on the constrained LMPC

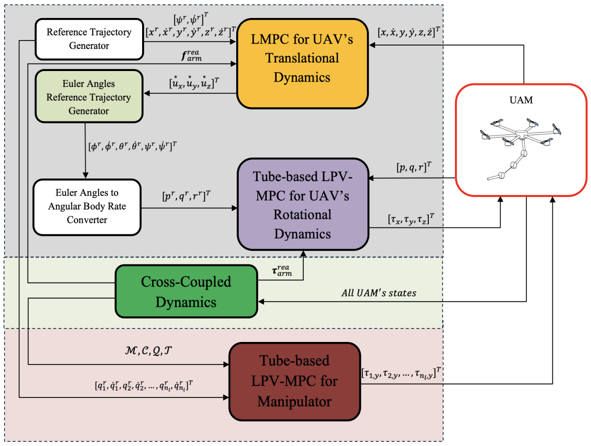
Reference Trajectory Generator
$[\psi^r, \dot{\psi}^r]^T$
$[x^r, \dot{x}^r, y^r, \dot{y}^r, z^r, \dot{z}^r]^T$
f_{arm}^{rea}
LMPC for UAV's Translational Dynamics
$[x, \dot{x}, y, \dot{y}, z, \dot{z}]^T$
Euler Angles Reference Trajectory Generator
$[\overset{*}{u}_x, \overset{*}{u}_y, \overset{*}{u}_z]^T$
UAM
$[\phi^r, \dot{\phi}^r, \theta^r, \dot{\theta}^r, \psi^r, \dot{\psi}^r]^T$
Euler Angles to Angular Body Rate Converter
$[p^r, q^r, r^r]^T$
Tube-based LPV-MPC for UAV's Rotational Dynamics
$[p, q, r]^T$
$[\tau_x, \tau_y, \tau_z]^T$
Cross-Coupled Dynamics
τ_{arm}^{rea}
All UAM's states
$\mathcal{M}, \mathcal{C}, \mathcal{Q}, \mathcal{T}$
$[q_1^r, \dot{q}_1^r, q_2^r, \dot{q}_2^r, \ldots, q_{n_l}^r, \dot{q}_{n_l}^r]^T$
Tube-based LPV-MPC for Manipulator
$[\tau_{1,y}, \tau_{2,y}, \ldots, \tau_{n_l,y}]^T$

Fig. 2: Cascaded MPC controller scheme

control results designed for the translational dynamic. This generated rotational reference trajectory will then be followed by a tube-based LPV-MPC controller designed for the rotational dynamic. In parallel, the manipulator's LPV-MPC control unit follows a given reference trajectory. Finally, the calculated control commands are applied to the UAM. The design of each control unit will be addressed in detail in the subsections that follow.

A. UAV's Translational Dynamics Controller Design

In order to develop an LMPC controller, we must formulate the nonlinear translational dynamic equation of the UAV, equation (5a), in discrete-time control affine form. we formulate a constrained optimization problem to address the trajectory tracking issue, aiming to determine the optimal force vector for the UAV. Afterwards, a desired reference trajectory is provided for the rotational controller using the optimal force vector obtained from the previous step.

1. *Constrained LMPC Controller*

We express the translational dynamics (5a) as an uncertain linear model in discrete-time, which is affected by the disturbance vector $\boldsymbol{w}_{\zeta,m} = [0, w_x, 0, w_y, 0, w_z]^T$, as follows:

$$\begin{aligned}&\boldsymbol{x}_{\zeta,m}(k+1) = \boldsymbol{A}_{\zeta,m}\boldsymbol{x}_{\zeta,m}(k) + \boldsymbol{B}_{\zeta,m}\boldsymbol{u}_{\zeta}(k) + \boldsymbol{B}^d_{\zeta,m}\boldsymbol{w}_{\zeta}(k),\\ &\boldsymbol{y}_{\zeta,m}(k) = \boldsymbol{C}_{\zeta,m}\boldsymbol{x}_{\zeta,m}(k),\end{aligned} \tag{13}$$

where the state vector is $\boldsymbol{x}_{\zeta,m} = [x, \dot{x}, y, \dot{y}, z, \dot{z}]^T$, the system output is $\boldsymbol{y}_{\zeta,m}$, and the control input vector is $\boldsymbol{u}_{\zeta} = [u_x, u_y, u_z + mg]^T$, such that:

$$\begin{aligned}
u_x &= (c_\psi c_\theta)(m_{1,f} f_z + \bar{f}_{1,x}) + (s_\phi s_\theta c_\psi - c_\phi s_\psi)(m_{2,f} f_z + \bar{f}_{1,y})\\
&+ (c_\phi s_\theta c_\psi + s_\phi s_\psi)(c_\phi c_\theta)((1 + m_{3,f}) f_z + \bar{f}_{1,z}),\\
u_y &= (c_\theta s_\psi)(m_{1,f} f_z + \bar{f}_{1,x}) + (s_\phi s_\theta s_\psi + c_\phi c_\psi)(m_{2,f} f_z + \bar{f}_{1,y})\\
&+ (c_\phi s_\theta s_\psi - s_\phi c_\psi)(c_\phi c_\theta)((1 + m_{3,f}) f_z + \bar{f}_{1,z}),\\
u_z + mg &= -(s_\theta)(m_{1,f} f_z + \bar{f}_{1,x})\\
&+ (s_\phi c_\theta)(m_{2,f} f_z + \bar{f}_{1,y}) + (c_\phi c_\theta)((1 + m_{3,f}) f_z + \bar{f}_{1,z}).
\end{aligned} \tag{14}$$

Also, $\boldsymbol{A}_{\zeta,m}$ and $\boldsymbol{B}^d_{\zeta,m}$ are 6×6 block diagonal matrices, where each diagonal element is $\boldsymbol{a}_{\zeta,m}$ and $\boldsymbol{b}^d_{\zeta,m}$, respectively. The rest matrices are defined as:

$$\begin{aligned}
\boldsymbol{a}_{\zeta,m} &= \begin{bmatrix} 1 & \Delta t_\zeta \\ 0 & 1 \end{bmatrix}, \boldsymbol{B}_{\zeta,m} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{\Delta t_\zeta}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{\Delta t_\zeta}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta t_\zeta}{m} \end{bmatrix},\\
\boldsymbol{b}^d_{\zeta,m} &= \begin{bmatrix} 0 & 0 \\ 0 & \Delta t_\zeta \end{bmatrix}, \boldsymbol{C}_{\zeta,m} = \boldsymbol{I}_6,
\end{aligned} \tag{15}$$

where $\boldsymbol{I}_6$ represents an identity matrix of size six. The following augmented state-space model is defined to reduce translational states' errors in the trajectory tracking problem:

$$\begin{aligned}&\boldsymbol{x}_{\zeta}(k+1) = \boldsymbol{A}_{\zeta}\boldsymbol{x}_{\zeta}(k) + \boldsymbol{B}_{\zeta}\Delta\boldsymbol{u}_{\zeta}(k) + \boldsymbol{B}^d_{\zeta}\boldsymbol{\epsilon}_{\zeta}(k),\\ &\boldsymbol{y}_{\zeta}(k) = \boldsymbol{C}_{\zeta}\boldsymbol{x}_{\zeta}(k),\end{aligned} \tag{16}$$

where,

$$\begin{aligned}
\boldsymbol{x}_{\zeta} &= \begin{bmatrix} \Delta\boldsymbol{x}_{\zeta,m}(k) \\ \boldsymbol{y}_{\zeta,m}(k) \end{bmatrix}, \boldsymbol{A}_{\zeta} = \begin{bmatrix} \boldsymbol{A}_{\zeta,m} & \boldsymbol{0}_{6\times 6} \\ \boldsymbol{C}_{\zeta,m}\boldsymbol{A}_{\zeta,m} & \boldsymbol{I}_6 \end{bmatrix},\\
\boldsymbol{B}_{\zeta} &= \begin{bmatrix} \boldsymbol{B}_{\zeta,m} \\ \boldsymbol{C}_{\zeta,m}\boldsymbol{B}_{\zeta,m} \end{bmatrix}, \boldsymbol{B}^d_{\zeta} = \begin{bmatrix} \boldsymbol{B}^d_{\zeta,m} \\ \boldsymbol{C}_{\zeta,m}\boldsymbol{B}^d_{\zeta,m} \end{bmatrix}, \boldsymbol{C}_{\zeta} = \begin{bmatrix} \boldsymbol{0}_{6\times 6} & \boldsymbol{C}_{\zeta,m} \end{bmatrix},
\end{aligned} \tag{17}$$

and $\boldsymbol{\epsilon}_{\zeta}(k) = \boldsymbol{w}_{\zeta}(k) - \boldsymbol{w}_{\zeta}(k-1)$.

Let $\boldsymbol{x}_{\zeta}(i|k)$ denote the predicted state vector, and $\boldsymbol{u}_{\zeta}(i|k)$ the future control signal vector, where N_ζ is the prediction. For the UAM's translational movement to track trajectory $\boldsymbol{x}^r_{\zeta}(k)$, a constrained optimization problem is formulated as below:

$$\begin{aligned}
&\min_{\Delta\boldsymbol{u}_{\zeta}} \quad J_{\zeta}(\boldsymbol{x}_{\zeta}, \boldsymbol{u}_{\zeta}) =\\
&\sum_{i=1}^{N_\zeta} \Big(\boldsymbol{x}_{\zeta}(i|k) - \boldsymbol{x}^r_{\zeta}(k)\Big)^t \boldsymbol{Q}_{\zeta} \Big(\boldsymbol{x}_{\zeta}(i|k) - \boldsymbol{x}^r_{\zeta}(k)\Big)\\
&+ \sum_{i=0}^{N_\zeta - 1} \Delta\boldsymbol{u}_{\zeta}(i|k)^t \boldsymbol{R}_{\zeta} \Delta\boldsymbol{u}_{\zeta}(i|k),\\
&\text{s.t.}: \boldsymbol{u}_{\zeta}(i|k) \in \mathbb{U}_{\zeta},\\
&\qquad \boldsymbol{x}_{\zeta}(i+1|k) \in \mathbb{X}_{\zeta}, \quad \forall i = 1, ..., N_\zeta - 1,
\end{aligned} \tag{18}$$

where $\mathbb{U}_{\zeta}$ and $\mathbb{X}_{\zeta}$ denote the control input and the state constraints sets, respectively, $\boldsymbol{Q}_{\zeta}$ and weight matrices are denoted by $\boldsymbol{R}_{\zeta}$, which are positive semidefinite.

2. *Reference Trajectory Calculation for Rotational Dynamics*

Since the UAV's dynamics are underactuated, we determine the rotational reference trajectory based on the controller's outcomes for the translational dynamic [50]. Upon optimizing the quadratic problem (18), we denote the optimal control input vector as $\overset{*}{\boldsymbol{u}}_{\zeta} = [\overset{*}{u}_x, \overset{*}{u}_y, \overset{*}{u}_z +$

$mg]^T$. By utilizing (14), the force component f_z is determined by solving the subsequent quadratic equation [51]:

$$F_x^2 + F_y^2 + F_z^2 = \overset{*}{u}_x^{\,2} + \overset{*}{u}_y^{\,2} + (\overset{*}{u}_z + mg)^2, \tag{19}$$

where $F_x = m_{1,f} f_z + \bar{f}_{1,x}$, $F_y = m_{2,f} f_z + \bar{f}_{1,y}$, and $F_z = (1 + m_{3,f}) f_z + \bar{f}_{1,z}$. After obtaining f_z, by considering equation (14), the reference Euler angle ϕ^r is given by [51]:

$$\phi^r = atan2\left(\frac{ac + b\sqrt{a^2 + b^2 - c^2}}{a^2 + b^2}, \frac{bc + a\sqrt{a^2 + b^2 - c^2}}{a^2 + b^2}\right), \tag{20}$$

where $a = F_z$, $b = F_y$, and $c = s_{\psi^r}\overset{*}{u}_x - c_{\psi^r}\overset{*}{u}_y$. Then, θ^r can be found using:

$$\theta^r = atan2(di - eh, de + hi), \quad de \neq -hi \tag{21}$$

where $d = c_{\psi^r}\overset{*}{u}_x + s_{\psi^r}\overset{*}{u}_y$, $e = F_x$, $h = \overset{*}{u}_z + mg$ and $i = F_y s_{\phi^r} + F_z s_{\phi^r}$. Then, using the obtained reference Euler rates, the angular body rate $\boldsymbol{\omega}$ should be calculated through equation (2). Finally, the UAV's rotational dynamics follows the reference trajectory $\boldsymbol{x}_\eta^r = [p^r, q^r, r^r]^T$.

B. UAV's Rotational Dynamics Controller Design

Here, we design a tube-based LPV-MPC controller for the UAV's rotational dynamics to track trajectory $\boldsymbol{x}_\eta^r$ provided by the UAV's translational dynamics controller. Initially, a parameter-dependent state-space model [52], [53] is used to develop a discrete-time LPV dynamic equations for the continuous nonlinear dynamics (5b). After formulating the uncertain system due to the uncertainty of varying parameters, a tube-based LPV-MPC controller will be formulated for the proposed model, ensuring the robustness and stability for the UAV's rotational dynamics.

1. LPV Dynamic Model for UAV's Rotational Dynamics

We represent the nonlinear dynamic model (5b) as a discrete-time LPV system as follows:

$$\boldsymbol{x}_\eta(k+1) = \boldsymbol{A}_\eta(\boldsymbol{\vartheta}(k))\boldsymbol{x}_\eta(k) + \boldsymbol{B}_\eta(\boldsymbol{\vartheta}(k))\boldsymbol{u}_\eta(k), \tag{22}$$

where $\boldsymbol{\vartheta} = [\vartheta_1, \vartheta_2, ..., \vartheta_{n_\vartheta}]$ is a time-varying parameter vector with n_ϑ components that belongs to the compact set $\mathbb{P}_\eta$, and depends on a scheduling signal vector $\boldsymbol{\rho} = [\rho_1, \rho_2, \rho_3, \rho_4, ..., \rho_{n_l+3}] = [p, q, r, q_1, ..., q_{n_l}]$, i.e., $\boldsymbol{\vartheta}(k) = \mathbf{P}_\eta(\boldsymbol{\rho}(k))$. Moreover, the corresponding state and control vectors are $\boldsymbol{x}_\eta = [p, q, r]^T$, $\boldsymbol{x}_\eta = [\tau_x + \bar{\tau}_{1,x}, \tau_y + \bar{\tau}_{1,y}, \tau_z + \bar{\tau}_{1,z}]^T$, respectively. Also, the system matrices $\boldsymbol{A}_\eta(\boldsymbol{\vartheta})$ and $\boldsymbol{B}_\eta(\boldsymbol{\vartheta})$ will be provided after considering the subsequent assumptions.

Assumption 1.

(i) The matrices $\boldsymbol{A}_\eta(\boldsymbol{\vartheta})$ and $\boldsymbol{B}_\eta(\boldsymbol{\vartheta})$ depend affinely on $\boldsymbol{\vartheta}$ as follows:

$$\begin{aligned} \boldsymbol{A}_\eta(\boldsymbol{\vartheta}) &= \boldsymbol{A}_{\eta,0} + \sum_{j=1}^{n_\vartheta} \vartheta_j \boldsymbol{A}_{\eta,j}, \\ \boldsymbol{B}_\eta(\boldsymbol{\vartheta}) &= \boldsymbol{B}_{\eta,0} + \sum_{j=1}^{n_\vartheta} \vartheta_j \boldsymbol{B}_{\eta,j}, \end{aligned} \tag{23}$$

where $\boldsymbol{A}_{\eta,j}$ and $\boldsymbol{B}_{\eta,j}$, $j = 1, ..., n_\vartheta$, are constant matrices.

(ii) The parameter compact set $\mathbb{P}_\eta$ can be expressed as $\mathbb{P}_\eta \subset \mathcal{R}^{n_\vartheta} : \boldsymbol{\vartheta}(k) \in \mathbb{P}_\eta \ \forall k > 0, \vartheta_j^{min} < \vartheta_j(k) < \vartheta_j^{max}, j = 1, ..., n_\vartheta$.

(iii) $\Delta\boldsymbol{\vartheta}(k) = \boldsymbol{\vartheta}(k) - \boldsymbol{\vartheta}(k-1)$ denotes the rate of variation for ϑ, that the corresponding compact set $\Delta\mathbb{P}_\eta$ can be defined as $\Delta\mathbb{P}_\eta \subset \mathcal{R}^{n_\vartheta} : \Delta\boldsymbol{\vartheta}(k) \in \Delta\mathbb{P}_\eta \ \forall k > 0, \Delta\vartheta_j^{min} < \Delta\vartheta_j(k) < \Delta\vartheta_j^{max}, j = 1, ..., n_\vartheta$.

Considering **Assumption 1**, system matrices $\boldsymbol{A}_\eta(\boldsymbol{\vartheta})$ and $\boldsymbol{B}_\eta(\boldsymbol{\vartheta})$ are defined below:

$$\begin{aligned} \boldsymbol{A}_\eta(\boldsymbol{\vartheta}) &= (\Delta t_\eta)\begin{bmatrix} \frac{1}{\Delta t_\eta} + \vartheta_1 & \vartheta_2 & \vartheta_3 \\ \vartheta_4 & \frac{1}{\Delta t_\eta} + \vartheta_5 & \vartheta_6 \\ \vartheta_7 & \vartheta_8 & \frac{1}{\Delta t_\eta} + \vartheta_9 \end{bmatrix}, \\ \boldsymbol{B}_\eta(\boldsymbol{\vartheta}) &= (\Delta t_\eta)\begin{bmatrix} \vartheta_{10} & \vartheta_{11} & \vartheta_{12} \\ \vartheta_{13} & \vartheta_{14} & \vartheta_{15} \\ \vartheta_{16} & \vartheta_{17} & \vartheta_{18} \end{bmatrix}, \end{aligned} \tag{24}$$

where $\vartheta_1, \vartheta_2, ..., \vartheta_9$ corresponds to each component of $\boldsymbol{P}_{\eta,1}(\boldsymbol{\rho}(k)) \in \mathcal{R}^{3\times3}$ respectively, i.e., $\vartheta_1 = (\boldsymbol{P}_{\eta,1})_{(1,1)}, \vartheta_2 = (\boldsymbol{P}_{\eta,1})_{(1,2)}, ..., \vartheta_9 = (\boldsymbol{P}_{\eta,1})_{(3,3)}$, and $\vartheta_{10}, \vartheta_{11}, ..., \vartheta_{18}$ corresponds to each component of $\boldsymbol{P}_{\eta,2}(\boldsymbol{\rho}(k)) \in \mathcal{R}^{3\times3}$ respectively, i.e., $\vartheta_{10} = (\boldsymbol{P}_{\eta,2})_{(1,1)}, \vartheta_{11} = (\boldsymbol{P}_{\eta,2})_{(1,2)}, ..., \vartheta_{18} = (\boldsymbol{P}_{\eta,2})_{(3,3)}$. Using system matrices (12b), $\boldsymbol{P}_{\eta,1}(\boldsymbol{\rho})$ and $\boldsymbol{P}_{\eta,2}(\boldsymbol{\rho})$ can be obtained as:

$$\begin{aligned} \boldsymbol{P}_{\eta,1}(\boldsymbol{\rho}) &= (\boldsymbol{I}_t - \bar{\boldsymbol{M}}_d^{rea})^{-1}\Bigg((\boldsymbol{M}_c + \bar{\boldsymbol{M}}_c^{rea})\begin{bmatrix} \rho_2 & 0 & 0 \\ 0 & \rho_3 & 0 \\ 0 & 0 & \rho_1 \end{bmatrix} \\ &\quad + (\boldsymbol{M}_s + \bar{\boldsymbol{M}}_s^{rea})\begin{bmatrix} \rho_1 & 0 & 0 \\ 0 & \rho_2 & 0 \\ 0 & 0 & \rho_3 \end{bmatrix} + \bar{\boldsymbol{M}}_l^{rea}\Bigg), \\ \boldsymbol{P}_{\eta,2}(\boldsymbol{\rho}) &= (\boldsymbol{I}_t - \bar{\boldsymbol{M}}_d^{rea})^{-1}, \end{aligned} \tag{25}$$

where matrices $\bar{\boldsymbol{M}}_d^{rea}$, $\bar{\boldsymbol{M}}_c^{rea}$, $\bar{\boldsymbol{M}}_s^{rea}$, and $\bar{\boldsymbol{M}}_l^{rea}$ are similar to the matrices $\boldsymbol{M}_d^{rea}$, $\boldsymbol{M}_c^{rea}$, $\boldsymbol{M}_s^{rea}$, and $\boldsymbol{M}_l^{rea}$, respectively, and just the UAM's states vector is replaced with scheduling signal vector $\boldsymbol{\rho}$.

For MPC controller design, the system dynamic matrices (24) indicate that the future evolutions of the system states depend on the future values for the varying parameter $\boldsymbol{\vartheta}$. Assuming that at time instant k we have only the value of $\boldsymbol{\vartheta}$ and the future trajectory evolutions are undetermined, a tube-based MPC controller will be designed, taking into account uncertain future evolutions for $\boldsymbol{\vartheta}$. By knowing the current value $\boldsymbol{\vartheta}(k)$ and the bounded rate of variation $\Delta\boldsymbol{\vartheta}$, the uncertain predicted trajectory evolutions for $\boldsymbol{\vartheta}$ at time instant $k+i$, $i = 1, ..., (N_\eta - 1)$, and prediction horizon N_η are obtained by:

$$\boldsymbol{\vartheta}(i|k) = \tilde{\boldsymbol{\vartheta}}(i|k) + \sum_{p=1}^{i} \Delta\boldsymbol{\vartheta}_{p|k}, \tag{26}$$

where $\tilde{\boldsymbol{\vartheta}}(i|k)$ is the nominal future trajectory evaluations for the time-varying parameter. Now, using (26), the system matrices $\boldsymbol{A}_\eta(\boldsymbol{\rho})$ and $\boldsymbol{B}_\eta(\boldsymbol{\rho})$ in (23) can be rewritten as:

$$\begin{aligned} \boldsymbol{A}_\eta(\boldsymbol{\vartheta}(i|k)) &= \boldsymbol{A}(i|k) + \sum_{j=1}^{n_\vartheta} \boldsymbol{A}_{\eta,j} \sum_{p=1}^{i} \Delta\vartheta_j(p|k), \\ \boldsymbol{B}_\eta(\boldsymbol{\vartheta}(i|k)) &= \boldsymbol{B}(i|k) + \sum_{j=1}^{n_\vartheta} \boldsymbol{B}_{\eta,j} \sum_{p=1}^{i} \Delta\vartheta_j(p|k), \end{aligned} \tag{27}$$

Hence, the predicted uncertain LPV dynamic model for the LPV system (22) is given by:

$$\boldsymbol{x}_\eta(i+1|k) = \boldsymbol{A}_\eta(\boldsymbol{\vartheta}(i|k))\boldsymbol{x}_\eta(i|k) + \boldsymbol{B}_\eta(\boldsymbol{\vartheta}(i|k))\boldsymbol{u}_\eta(i|k), \tag{28}$$

in which $\boldsymbol{x}_\eta(i|k)$ and $\boldsymbol{u}_\eta(i|k)$ denote the predicted states and control input vectors, where $i = 0, ..., N_\eta - 1$. Also, the uncertain LPV model (27) by including the nominal system matrices $\boldsymbol{A}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))$ and $\boldsymbol{B}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))$ is given by:

$$\begin{aligned} \boldsymbol{x}_\eta(i+1|k) =& \boldsymbol{A}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))\boldsymbol{x}_\eta(i|k) \\ &+ \boldsymbol{B}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))\boldsymbol{u}_\eta(i|k) + \boldsymbol{w}_\eta(i|k). \end{aligned} \tag{29}$$

Consequently, using (27), (28), and (29), the $\boldsymbol{w}_\eta(i|k)$ vector can be represented as follows:

$$\begin{aligned} \boldsymbol{w}_\eta(i|k) =& \left(\sum_{j=1}^{n_\vartheta} \boldsymbol{A}_{\eta,j} \sum_{p=1}^{i} \Delta\vartheta_j(p|k)\right)\boldsymbol{x}_\eta(i|k) \\ &+ \left(\sum_{j=1}^{n_\vartheta} \boldsymbol{B}_{\eta,j} \sum_{p=1}^{i} \Delta\vartheta_j(p|k)\right)\boldsymbol{u}_\eta(i|k). \end{aligned} \tag{30}$$

2. *Tube-based LPV-MPC Controller for UAV's Rotational Dynamics System*

Here, we design a tube-based LPV-MPC controller for the UAV's rotational dynamics system. Considering the LPV system (29), a nominal LPV system is defined as follows:

$$\tilde{\boldsymbol{x}}_\eta(i+1|k) = \boldsymbol{A}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))\tilde{\boldsymbol{x}}_\eta(i|k) + \boldsymbol{B}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))\tilde{\boldsymbol{u}}_\eta(i|k), \tag{31}$$

here, $\tilde{\boldsymbol{x}}\eta$ represents the nominal state, and $\tilde{\boldsymbol{u}}\eta$ denotes the nominal control input. The following control input is considered to offset any discrepancies between the actual and the nominal dynamic systems (29) and (31), respectively, [54], [55]:

$$\boldsymbol{u}_\eta(i|k) = \boldsymbol{K}_\eta\boldsymbol{e}_\eta(i|k) + \tilde{\boldsymbol{u}}_\eta(i|k), \tag{32}$$

where $\boldsymbol{e}_\eta(k) = \boldsymbol{x}_\eta(k) - \tilde{\boldsymbol{x}}_\eta(k)$, and the feedback control gain $\boldsymbol{K}_\eta$ stabilize the system matrix $\boldsymbol{A}_{\boldsymbol{K},\eta}(\tilde{\boldsymbol{\vartheta}}(i|k)) \triangleq \boldsymbol{A}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k)) + \boldsymbol{B}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))\boldsymbol{K}_\eta$ ($\boldsymbol{K}_\eta \triangleq \boldsymbol{K}^{LQR}$) [56]. Hence, a system for mismatch error is given by:

$$\boldsymbol{e}_\eta(i+1|k) = \boldsymbol{A}_{\boldsymbol{K},\eta}(\tilde{\boldsymbol{\vartheta}}(i|k))\boldsymbol{e}_\eta(i|k) + \boldsymbol{w}_\eta(i|k). \tag{33}$$

Let $\mathbb{W}_\eta$ denote a polytope disturbance set, which can be calculated using (30) such that:

$$\begin{aligned} &\mathbb{W}_\eta(i|k) = \boldsymbol{Conv}\{\boldsymbol{w}_\eta(i|k)|\Delta\boldsymbol{\vartheta}(i|k) \in \Delta\mathbb{P}_\eta, \\ &\boldsymbol{x}_\eta(i|k) \in \mathbb{X}_\eta,\ \boldsymbol{u}_\eta(i|k) \in \mathbb{U}_\eta, \forall i = 0, ..., N_\eta - 1\}, \end{aligned} \tag{34}$$

where $\boldsymbol{Conv}\{\cdot\}$ represents the convex hull of a set, $\mathbb{X}_\eta$ and $\mathbb{U}_\eta$ are polytopes and denote the state and control input constraints sets. Next, we need to determine a boundary for the evolutions of the uncertain system (33) over the N_η around the nominal system trajectory by defining the reachable sets as follows:

$$\mathscr{R}_\eta(i+1|k) := \boldsymbol{A}_{\boldsymbol{K},\eta}(\tilde{\boldsymbol{\vartheta}}(i|k))\mathscr{R}_\eta(i|k) \oplus \mathbb{W}_\eta(i|k), \tag{35}$$

where $\mathscr{R}_\eta(0|k) = \{\boldsymbol{0}\}$ is the initial condition and $\oplus$ represents the Minkowski sum operation. In addition, to compensate for the mismatch error system effectively, the state constraints set $\mathbb{X}_\eta$ and input constraints set $\mathbb{U}_\eta$ should be restricted further using the obtained reachable sets this way:

$$\begin{aligned} \tilde{\mathbb{X}}_\eta(i) &= \mathbb{X}_\eta \ominus \mathscr{R}_\eta(i|k), \quad \forall i = 1, ..., N_\eta - 1 \\ \tilde{\mathbb{U}}_\eta(i) &= \mathbb{U}_\eta \ominus \boldsymbol{K}_\eta\mathscr{R}_\eta(i|k), \quad \forall i = 0, ..., N_\eta - 1 \end{aligned} \tag{36}$$

where $\ominus$ represents the Minkowski difference operation.

To track the trajectory $\boldsymbol{x}_\eta^r$, we define the subsequent optimization problem:

$$\begin{aligned} &\min_{\tilde{\boldsymbol{u}}_\eta} J_\eta(\tilde{\boldsymbol{x}}_\eta, \tilde{\boldsymbol{u}}_\eta) \\ &\text{subject to}: \tilde{\boldsymbol{u}}_\eta(i|k) \in \tilde{\mathbb{U}}_\eta(i), \quad \forall i = 1, ..., N_\eta - 1, \\ &\qquad \tilde{\boldsymbol{x}}_\eta(i+1|k) \in \tilde{\mathbb{X}}_\eta(i), \quad \forall i = 0, ..., N_\eta - 1 \\ &\qquad \tilde{\boldsymbol{x}}_\eta(N_\eta|k) \in \mathcal{X}_\eta, \end{aligned} \tag{37}$$

where $\mathcal{X}_\eta$ is terminal set, $J_\eta(\tilde{\boldsymbol{x}}_\eta, \tilde{\boldsymbol{u}}_\eta) = \mathcal{L}_\eta(\tilde{\boldsymbol{x}}_\eta, \tilde{\boldsymbol{u}}_\eta) + \Upsilon_\eta(N_\eta)$, in which $\mathcal{L}_\eta$ denotes the objective function and Υ_η denotes the terminal cost and given by:

$$\begin{aligned} &\mathcal{L}_\eta(\tilde{\boldsymbol{x}}_\eta, \tilde{\boldsymbol{u}}_\eta) = \\ &\sum_{i=0}^{N_\eta-1} \Big(\tilde{\boldsymbol{x}}_\eta(i|k) - \boldsymbol{x}_\eta^r(k)\Big)^t \boldsymbol{Q}_\eta \Big(\tilde{\boldsymbol{x}}_\eta(i|k) - \boldsymbol{x}_\eta^r(k)\Big) \\ &+ \tilde{\boldsymbol{u}}_\eta(i|k)^t \boldsymbol{R}_\eta \tilde{\boldsymbol{u}}_\eta(i|k), \end{aligned} \tag{38}$$

$$\Upsilon_\eta(\tilde{\boldsymbol{x}}_\eta(N_\eta|k)) = \tilde{\boldsymbol{x}}_\eta(N_\eta|k)^T \boldsymbol{P}_\eta \tilde{\boldsymbol{x}}_\eta(N_\eta|k).$$

C. Manipulator's Controller Design

Here, we formulate a tube-based LPV-MPC controller for the manipulator to follow a desired reference trajectory despite the coupling effects imposed by the UAV. First, the continuous nonlinear dynamics (5b) will be written in a discrete-time control affine form to obtain a LPV model formulation. Consequently, a tube-based LPV-MPC controller will be formulated for the proposed model.

1. LPV Dynamic Model for Manipulator's Dynamic

We rewrite the state-space dynamic equations (9) by modifying the gravity vector as $\mathcal{G}(\cdot) = \mathcal{Q}(\cdot)\boldsymbol{\Theta} + \mathcal{T}(\cdot)$, in which $\sin(q_i)$ terms, $i = 1, ..., n_l$ are substituted with $\operatorname{sinc}(q_i)q_i = (\frac{\sin(q_i)}{q_i})q_i$. After defining $\mathcal{T}(\cdot) = [T_{1,y}, ..., T_{n_l,y}]^T$, we have:

$$\begin{bmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_{n_l} \end{bmatrix} = \mathcal{M}^{-1}(\boldsymbol{\Theta}) \left(-\mathcal{C}(\boldsymbol{\Theta}, \dot{\boldsymbol{\Theta}}, \boldsymbol{\omega}) \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_{n_l} \end{bmatrix} \right. \\ \left. -\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\omega}, \dot{\boldsymbol{\omega}}, \boldsymbol{f}_{uav}) \begin{bmatrix} q_1 \\ \vdots \\ q_{n_l} \end{bmatrix} + \begin{bmatrix} \tau_{1,y} + T_{1,y} \\ \vdots \\ \tau_{n_l,y} + T_{n_l,y} \end{bmatrix} \right). \tag{39}$$

Therefore, a discrete-time LPV model for the dynamic model (39) is given by:

$$\begin{aligned} \boldsymbol{x}_\gamma(k+1) &= \boldsymbol{A}_\gamma(\boldsymbol{\varphi}(k))\boldsymbol{x}_\gamma(k) + \boldsymbol{B}_\gamma(\boldsymbol{\varphi}(k))\boldsymbol{u}_\gamma(k), \\ \boldsymbol{y}_\gamma(k) &= \boldsymbol{C}_\gamma \boldsymbol{x}_\gamma(k), \end{aligned} \tag{40}$$

where $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, ..., \varphi_{n_\varphi}]$ is the time-varying parameter vector with n_φ components that belongs to the compact set $\mathbb{P}_\varphi$. Also, $\boldsymbol{\varphi}$ is a function of scheduling signal vector $\boldsymbol{\rho}$, i.e., $\boldsymbol{\varphi}(k) = \mathbf{P}_\gamma(\boldsymbol{\rho}(k))$. In addition, $\boldsymbol{x}_\gamma = [q_1, ..., q_{n_l}, \dot{q}_1, ..., \dot{q}_{n_l}]^T$ is the state vector, and $\boldsymbol{u}_\gamma = [\tau_{1,y} + T_{1,y}, ..., \tau_{n_l,y} + T_{n_l,y}]^T$ is the control vector. Additionally, $\boldsymbol{C}_\gamma = \boldsymbol{I}_{n_l}$, and after taking into account similar assumptions like **Assumption 1**, system matrices $\boldsymbol{A}_\gamma(\boldsymbol{\vartheta})$ and $\boldsymbol{B}_\gamma(\boldsymbol{\vartheta})$ for $n_l = 3$ are expressed as:

$$\boldsymbol{A}_\gamma(\boldsymbol{\varphi}) = (\Delta t_\gamma) \\ \begin{bmatrix} \frac{1}{\Delta t_\gamma} & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{\Delta t_\gamma} & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\Delta t_\gamma} & 0 & 0 & 1 \\ \varphi_1 & \varphi_2 & \varphi_3 & \frac{1}{\Delta t_\gamma} + \varphi_{10} & \varphi_{11} & \varphi_{12} \\ \varphi_4 & \varphi_5 & \varphi_6 & \varphi_{13} & \frac{1}{\Delta t_\gamma} + \varphi_{14} & \varphi_{15} \\ \varphi_7 & \varphi_8 & \varphi_9 & \varphi_{16} & \varphi_{17} & \frac{1}{\Delta t_\gamma} + \varphi_{18} \end{bmatrix}, \\ \boldsymbol{B}_\gamma(\boldsymbol{\varphi}) = (\Delta t_\gamma) \begin{bmatrix} & \mathbf{0}_3 & \\ \varphi_{19} & \varphi_{20} & \varphi_{21} \\ \varphi_{22} & \varphi_{23} & \varphi_{24} \\ \varphi_{25} & \varphi_{26} & \varphi_{27} \end{bmatrix}, \tag{41}$$

where $\mathbf{0}_3$ is a zero matrix of size 3, $\varphi_1, \varphi_2, ..., \varphi_9$ are associated with each component of $\boldsymbol{P}_{\gamma,1}(\boldsymbol{\rho}(k)) \in \mathcal{R}^{3\times3}$, i.e., $\varphi_1 = (\boldsymbol{P}_{\gamma,1})_{(1,1)}, \varphi_2 = (\boldsymbol{P}_{\gamma,1})_{(1,2)}, ..., \varphi_9 = (\boldsymbol{P}_{\gamma,1})_{(3,3)}$. Similarly, $\varphi_{10}, \varphi_{11}, ..., \varphi_{18}$ and $\varphi_{19}, \varphi_{12}, ..., \varphi_{27}$ are associated with each component of $\boldsymbol{P}_{\gamma,2}(\boldsymbol{\rho}(k)) \in \mathcal{R}^{3\times3}$ and $\boldsymbol{P}_{\gamma,3}(\boldsymbol{\rho}(k)) \in \mathcal{R}^{3\times3}$, respectively. Also, the matrices $\boldsymbol{P}_{\gamma,1}(\boldsymbol{\rho})$, $\boldsymbol{P}_{\gamma,2}(\boldsymbol{\rho})$, and $\boldsymbol{P}_{\gamma,3}(\boldsymbol{\rho})$ can be expressed based on (39) as:

$$\begin{aligned} \boldsymbol{P}_{\gamma,1}(\boldsymbol{\rho}) &= -\bar{\mathcal{M}}^{-1}\bar{\mathcal{C}}, \\ \boldsymbol{P}_{\gamma,2}(\boldsymbol{\rho}) &= -\bar{\mathcal{M}}^{-1}\bar{\mathcal{Q}}, \\ \boldsymbol{P}_{\gamma,3}(\boldsymbol{\rho}) &= \bar{\mathcal{M}}^{-1}, \end{aligned} \tag{42}$$

where $\bar{\mathcal{M}}$, $\bar{\mathcal{C}}$, and $\bar{\mathcal{Q}}$ matrices are the same as $\mathcal{M}$, $\mathcal{C}$, and $\mathcal{Q}$ matrices, respectively, and just the scheduling signal vector $\boldsymbol{\rho}$ is used instead of UAM's states vector. In order to design the MPC controller, by knowing the bounded rate of variation $\Delta\boldsymbol{\varphi}$ for the varying parameter $\boldsymbol{\varphi}$ and the nominal future trajectory evolution $\tilde{\boldsymbol{\varphi}}(i|k)$, the uncertain predicted trajectory evolutions $\boldsymbol{\varphi}(i|k)$ can be defined similar to equation (26) at time instant $k+i$, $i = 1, ..., (N_\gamma - 1)$. Also, similar to equations (27), we define the system matrices $\boldsymbol{A}_\gamma(\tilde{\boldsymbol{\varphi}}(i|k))$ and $\boldsymbol{A}_\gamma(\tilde{\boldsymbol{\varphi}}(i|k))$ to obtain the following uncertain LPV dynamic model:

$$\boldsymbol{x}_\gamma(i+1|k) = \boldsymbol{A}_\gamma(\boldsymbol{\varphi}(i|k))\boldsymbol{x}_\gamma(i|k) + \boldsymbol{B}_\gamma(\boldsymbol{\varphi}(i|k))\boldsymbol{u}_\gamma(i|k), \tag{43}$$

By defining the nominal system matrices $\boldsymbol{A}_\gamma(\tilde{\boldsymbol{\varphi}}(i|k))$ and $\boldsymbol{B}_\gamma(\tilde{\boldsymbol{\varphi}}(i|k))$, the uncertain LPV model (43) can be written as:

$$\begin{aligned} \boldsymbol{x}_\gamma(i+1|k) =& \boldsymbol{A}_\gamma(\tilde{\boldsymbol{\varphi}}(i|k))\boldsymbol{x}_\gamma(i|k) \\ &+ \boldsymbol{B}_\gamma(\tilde{\boldsymbol{\varphi}}(i|k))\boldsymbol{u}_\gamma(i|k) + \boldsymbol{w}_\gamma(i|k). \end{aligned} \tag{44}$$

where the $\boldsymbol{w}_\gamma(i|k)$ vector can be defined similar to equation (30).

2. Tube-based LPV-MPC controller for manipulator

After defining the predicted nominal LPV system $\tilde{\boldsymbol{x}}_\gamma(i+1|k)$, feedback control gain $\boldsymbol{K}_\gamma$, mismatch error system $\boldsymbol{e}_\gamma(i+1|k)$, and the disturbance set $\mathbb{W}_\gamma$ similar to equations (31-34), the reachable sets over the N_γ horizon are given by:

$$\mathscr{R}_\gamma(i+1|k) := \boldsymbol{A}_{\boldsymbol{K},\gamma}(\tilde{\boldsymbol{\varphi}}(i|k))\mathscr{R}_\gamma(i|k) \oplus \mathbb{W}_\gamma(i|k), \tag{45}$$

where $\mathscr{R}_\gamma(0|k) = \{\mathbf{0}\}$. Now, the restricted set of state constraints $\mathbb{X}_\eta$ and input constraints $\mathbb{U}_\eta$ are given by:

$$\begin{aligned} \tilde{\mathbb{X}}_\gamma(i) &= \mathbb{X}_\gamma \ominus \mathscr{R}_\gamma(i|k), \quad \forall i = 1, ..., N_\gamma - 1 \\ \tilde{\mathbb{U}}_\gamma(i) &= \mathbb{U}_\gamma \ominus \boldsymbol{K}_\gamma\mathscr{R}_\gamma(i|k), \quad \forall i = 0, ..., N_\gamma - 1 \end{aligned} \tag{46}$$

Consequently, the following optimization problem should be solved to track the reference trajectory $\boldsymbol{x}_\gamma^r$:

$$\begin{aligned} &\min_{\tilde{\boldsymbol{u}}_\gamma} J_\gamma(\tilde{\boldsymbol{x}}_\gamma, \tilde{\boldsymbol{u}}_\gamma) \\ &\text{s.t. : } \tilde{\boldsymbol{u}}_\gamma(i|k) \in \tilde{\mathbb{U}}_\gamma(i), \quad \forall i = 1, ..., N_\gamma - 1, \\ &\qquad \tilde{\boldsymbol{x}}_\gamma(i+1|k) \in \tilde{\mathbb{X}}_\gamma(i), \quad \forall i = 0, ..., N_\gamma - 1 \\ &\qquad \tilde{\boldsymbol{x}}_\gamma(N_\gamma|k) \in \mathcal{X}_\gamma, \end{aligned} \tag{47}$$

where $\mathcal{X}_\gamma$ represents the terminal set, $J_\gamma(\tilde{\boldsymbol{x}}_\gamma, \tilde{\boldsymbol{u}}_\gamma) = \mathcal{L}_\gamma(\tilde{\boldsymbol{x}}_\gamma, \tilde{\boldsymbol{u}}_\gamma) + \Upsilon_\gamma(N_\gamma)$, in which:

$$\begin{aligned} &\mathcal{L}_\gamma(\tilde{\boldsymbol{x}}_\gamma, \tilde{\boldsymbol{u}}_\gamma) = \\ &\sum_{i=0}^{N_\gamma-1} \Big(\tilde{\boldsymbol{x}}_\gamma(i|k) - \boldsymbol{x}_\gamma^r(k)\Big)^t \boldsymbol{Q}_\gamma \Big(\tilde{\boldsymbol{x}}_\gamma(i|k) - \boldsymbol{x}_\gamma^r(k)\Big) \\ &+ \tilde{\boldsymbol{u}}_\gamma(i|k)^t \boldsymbol{R}_\gamma \tilde{\boldsymbol{u}}_\gamma(i|k), \\ &\Upsilon_\gamma(\tilde{\boldsymbol{x}}_\gamma(N_\gamma|k)) = \tilde{\boldsymbol{x}}_\gamma(N_\gamma|k)^T \boldsymbol{P}_\gamma \tilde{\boldsymbol{x}}_\gamma(N_\gamma|k). \end{aligned} \tag{48}$$

IV. SYSTEM'S FEASIBILITY, AND STABILITY, AND ALGORITHM IMPLEMENTATION

We proposed a decoupled control scheme for a UAM in the previous section, consisting of an LMPC controller for the UAV's translational dynamics, tube-based MPC controllers for the UAV's rotational dynamics, and the manipulator. As proposed in [51], [57], [58], It's common practice to update the controller associated with the rotational dynamics more frequently than the controller of the translational dynamics to maintain UAV stability. Therefore, it's logical to decouple these dynamics, allowing for separate stability assessments of each dynamic system. The stability analysis from [50] is adapted to the translational dynamics of the UAV in order to ensure the feasibility of the closed-loop system and its stability. This analysis employs a pole placement method using the LQR control strategy, aiming to achieve a specified level of stability where closed-loop system eigenvalues fit inside a smaller circle and matches to a set of values. As the detailed explanation of this approach can be found in [50], it will not be reiterated here. Instead, we will focus on discussing the stability and feasibility concerning both the rotational dynamics and the manipulator.

In the following, the necessary definitions and theorems under which the UAV's closed-loop rotational dynamics system and the manipulator's closed-loop system are feasible and stable are provided. For the brevity of the paper, because tube-based MPC is employed for both systems, we use the $(\cdot)$ subscript instead of η and γ for those systems, respectively.

Definition 1 [59], [60].
Considering the optimization problem (38), the system (22) is quadratically stabilizable if the provided positive definite function $\Upsilon_{(\cdot)}(\boldsymbol{x}_{(\cdot)})$ and the control gain $\boldsymbol{u}_{(\cdot)} = \boldsymbol{K}_{(\cdot)}\boldsymbol{x}_{(\cdot)}$ satisfy:

$$\begin{aligned}&\Upsilon_{(\cdot)}(\boldsymbol{A}_{\boldsymbol{k},(\cdot)}\boldsymbol{x}_{(\cdot)}) - \Upsilon_{(\cdot)}(\boldsymbol{x}_{(\cdot)}) \leq \\ &-\boldsymbol{x}^T_{(\cdot)}(\boldsymbol{Q}_{(\cdot)} + \boldsymbol{K}^T_{(\cdot)}\boldsymbol{R}_{(\cdot)}\boldsymbol{K}_{(\cdot)})\boldsymbol{x}_{(\cdot)},\end{aligned} \tag{49}$$

hence the origin of the system $\boldsymbol{x}_{(\cdot)}(k+1) = \boldsymbol{A}_{\boldsymbol{k},(\cdot)}\boldsymbol{x}_{(\cdot)}(k)$ is exponentially stable.

It must be noted that the terminal set $\mathcal{X}_{(\cdot)}$ and the control gain $\boldsymbol{K}_{(\cdot)}$ are computed offline. In fact, $\mathcal{X}_{(\cdot)}$ is calculated as a robust positively invariant (RPI) set using MPT toolbox [61]. Now, using the following proposition, we address the optimization problem's feasibility (38).

Proposition 1 [60].
Given the terminal cost $\Upsilon_{(\cdot)}$ and the terminal set $\mathcal{X}_{(\cdot)}$ established as a RPI set under the control action of $\boldsymbol{K}_{(\cdot)}$, and assuming the fulfillment of **Assumption 1**, it follows that the optimization problem (38) exhibits recursive feasibility.

A detailed proof of **Proposition 1** is provided in [60]. This proposition implies that, for each time step, a feasible solution exists, ensuring the continued feasibility of the optimization problem in a recursive manner under the specified constraints. Now, using the following theorem, we can ensure that the provided LPV-MPC approach stabilizes the rotational dynamics system.

Theorem 1 [60].
Suppose that the terminal cost $\Upsilon_{(\cdot)}$ meets the conditions outlined in **Definition 1**, and considering the fulfillment of the conditions specified in **Proposition 1**, with the given optimization problem (38) in the provided LPV-MPC controller, the system (22) in closed-loop is asymptotically stable.

A proof of **Theorem 1** is presented in [60]. This theorem implies that stability can be achieved by the provided LPV-MPC controller, which confines the UAV's rotational dynamics system trajectories within the tube despite uncertainties, ensuring that the actual trajectories converge towards the nominal trajectories.

Algorithm 1 represents the step-by-step implementation the provided control method. According this algorithm, the terminal costs $\Upsilon_{(\eta)}$ and $\Upsilon_{(\gamma)}$, the terminal sets $\mathcal{X}_{(\eta)}$ and $\mathcal{X}_{(\gamma)}$, and the control gains $\boldsymbol{K}_{(\eta)}$ and $\boldsymbol{K}_{(\gamma)}$ are computed offline. For the online part of the algorithm, after measuring the scheduling and computing reachable sets, states, and control input constraints sets, the optimization problems (17), (37), and (47) will be solved to find control inputs $\boldsymbol{u}_\zeta$, $\tilde{\boldsymbol{u}}_\eta$, and $\tilde{\boldsymbol{u}}_\gamma$, and then applied to the UAM.

Algorithm 1 The proposed control algorithm

```
Require: $\Upsilon_\eta$, $\Upsilon_\gamma$, $\mathcal{X}_\eta$, $\mathcal{X}_\gamma$, $\boldsymbol{K}_\eta$, $\boldsymbol{K}_\gamma$, and control parameters ▷
    Offline
    Initialization $\boldsymbol{x}_\zeta$, $\boldsymbol{x}_\eta$, $\boldsymbol{x}_\gamma$, $\boldsymbol{\rho}$, $\boldsymbol{\vartheta}$, $\boldsymbol{\varphi}$ at the time of $k = 0$
while the system is running do                                        ▷ Online
    Measure $\boldsymbol{x}_\zeta(k)$ and get $\boldsymbol{x}^r_\zeta(k)$ then solve optimization problem (17) to find $\boldsymbol{u}_\zeta$
    Calculate $\phi^r$ and $\theta^r$ using (19-21)
    Calculate $\boldsymbol{x}^r_\eta$ using (2)
    Measure $\boldsymbol{\rho}(k)$ and $\boldsymbol{\vartheta}(k)$ then compute $\boldsymbol{A}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))$, $\boldsymbol{B}_\eta(\tilde{\boldsymbol{\vartheta}}(i|k))$, $\boldsymbol{A}_{\boldsymbol{K},\eta}(\tilde{\boldsymbol{\vartheta}}(i|k))$, $\mathbb{W}_\eta(i|k)$, $\mathscr{R}_\eta(i|k)$, $\tilde{\mathbb{X}}_\eta(i)$, $\tilde{\mathbb{U}}_\eta(i)$, $\forall i = 0, ..., N_\eta - 1$
    Measure $\boldsymbol{x}_\eta(k)$ and get $\boldsymbol{x}^r_\eta(k)$ then solve optimization problem (37) to find $\tilde{\boldsymbol{u}}_\eta$
    Measure $\boldsymbol{\rho}(k)$ and $\boldsymbol{\varphi}(k)$ then compute $\boldsymbol{A}_\gamma(\tilde{\boldsymbol{\varphi}}(i|k))$, $\boldsymbol{B}_\gamma(\tilde{\boldsymbol{\varphi}}(i|k))$, $\boldsymbol{A}_{\boldsymbol{K},\gamma}(\tilde{\boldsymbol{\varphi}}(i|k))$, $\mathbb{W}_\gamma(i|k)$, $\mathscr{R}_\gamma(i|k)$, $\tilde{\mathbb{X}}_\gamma(i)$, $\tilde{\mathbb{U}}_\gamma(i)$, $\forall i = 0, ..., N_\gamma - 1$
    Measure $\boldsymbol{x}_\gamma(k)$ and get $\boldsymbol{x}^r_\gamma(k)$ then solve optimization problem (47) to find $\tilde{\boldsymbol{u}}_\gamma$
    apply the control inputs $\boldsymbol{u}_\zeta$, $\tilde{\boldsymbol{u}}_\eta$ and $\tilde{\boldsymbol{u}}_\gamma$ to the UAM
    $k \leftarrow k + 1$
end while
```

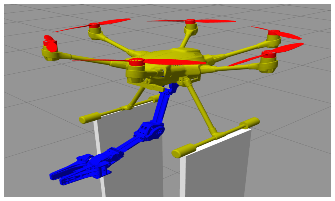

Fig. 3: UAM's scheme in Gazebo

V. Simulation results

We assess the effectiveness of the suggested LPV dynamic modelling and control design for the UAM in the trajectory tracking problem in this section. Initially, we conducted a comparison between our dynamic modelling approach and previous studies that aimed to characterize the cross-coupled dynamics of UAM by ERTF approach. Specifically, similar to [24], [25], [31], we use RNE equations to measure the reaction torque and force imposed on the UAV. This comparison is conducted while attempting to navigate a demanding trajectory for the UAM. Subsequently, we explore an even more challenging reference trajectory to further assess the capabilities of our proposed approach.

In this simulation, the UAM used is made of a hexarotor and a 3-link manipulator attached to it. The UAM is modeled and simulated in a ROS/Gazebo environment, and a schematic of the UAM is shown in Fig. 3. The UAV is expected to follow a predefined sinusoidal reference trajectory (RT) along the x and y axes, while following a ramp trajectory along the z axis. Simultaneously, the manipulator's joints are required to follow a predefined periodic step RT characterized by positive and negative amplitude. Table I includes the model parameters for the UAM, and Table II contains the control parameters utilised for the simulation.

TABLE I: UAM model parameters

Symbol	Description	Value
m_{uav}	Hexarotor mass	$5\ (kg)$
m_i	ith arm link mass	$0.166\ (kg)$
l_i	ith arm link length	$0.33\ (m)$
g	Gravity	$9.8\ (m \cdot s^{-2})$
i_{xx}^{uav}	UAV's MOI about x_b	$0.18\ (kg \cdot m^2)$
i_{yy}^{uav}	UAV's MOI about y_b	$0.18\ (kg \cdot m^2)$
i_{zz}^{uav}	UAV's MOI about z_b	$0.3\ (kg \cdot m^2)$
i_{xx}^{i}	ith link's MOI about x_{M_i}	$15 \times 10^{-4}\ (kg \cdot m^2)$
i_{yy}^{i}	ith link's MOI about y_{M_i}	$15 \times 10^{-4}\ (kg \cdot m^2)$
i_{zz}^{i}	ith link's MOI about z_{M_i}	$55 \times 10^{-6}\ (kg \cdot m^2)$

Figure 4 depicts the UAM's performance to track a designated reference trajectory in three-dimensional space for the two controllers,LPV-MPC (ours) and ERTF (baseline). LPV-MPC approach demonstrates better performance than ERTF approach. Specifically, the trajectory associated with the ERTF method deviates from the RT, leading the UAM to be on the verge of instability. Sub-figures 5 plot the UAV's motion along the x, y, and z axes. The sub-figures highlight that the LPV-MPC method consistently maintains accurate trajectory tracking, whereas the ERTF approach exhibits a growing trajectory following error over time (which becomes more noticeable after $t = 20s$), and this discrepancy finally results in the UAM significantly deviating from the RT. Table (III) shows the integral absolute value of error (IAE) performance index.

Time plots of UAV's Euler angles and the manipulator's joint angles trajectory tracking responses are illustrated in Figs. 6 and 7, respectively. LPV-MPC approach shows better performance in following the trajectory provided by the UAV's translational dynamics compared to

TABLE II: Controller's parameters

Parameter	N_ζ	N_η	N_γ	Δt_ζ	Δt_η
Value	8	4	5	0.05	0.025
Parameter	Δt_γ	$\boldsymbol{Q}_\zeta$	$\boldsymbol{Q}_\eta$	$\boldsymbol{Q}_\gamma$	$\boldsymbol{R}_\zeta$
Value	0.025	$10\boldsymbol{I}_{12}$	$15\boldsymbol{I}_3$	$100\boldsymbol{I}_6$	$0.1\boldsymbol{I}_3$
Parameter	$\boldsymbol{R}_\eta$	$\boldsymbol{R}_\gamma$	$\boldsymbol{P}_\eta$	$\boldsymbol{P}_\gamma$	
Value	$0.01\boldsymbol{I}_3$	$0.01\boldsymbol{I}_3$	$15\boldsymbol{I}_3$	$100\boldsymbol{I}_3$	

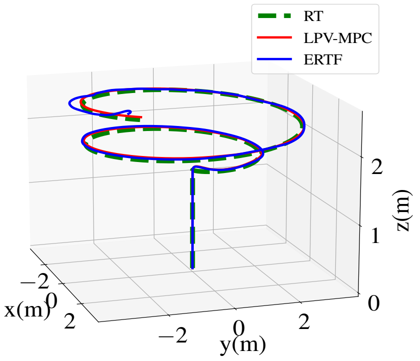

Fig. 4: 3D trajectory tracking

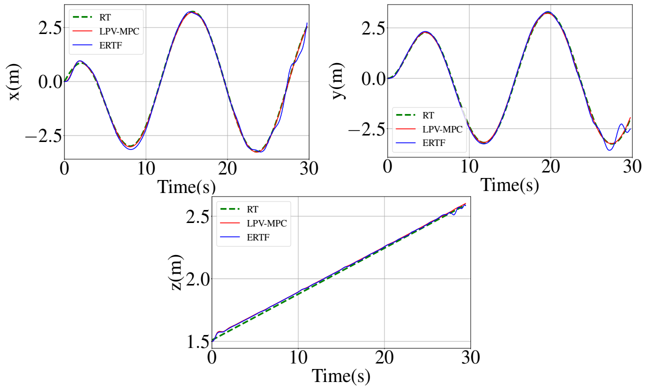

Fig. 5: Position response of the UAV

TABLE III: IAE performance index

States	LPV+MPC	ERTF
x	2.35 m	3.39 m
y	1.32 m	2.83 m
z	0.52 m	0.79 m
ϕ	0.041 rad	0.64 rad
θ	0.088 rad	1.67 rad
ψ	0.0 rad	0.0 rad
q_1	0.22 rad	0.32 rad
q_2	0.21 rad	0.41 rad
q_3	0.23 rad	0.29 rad

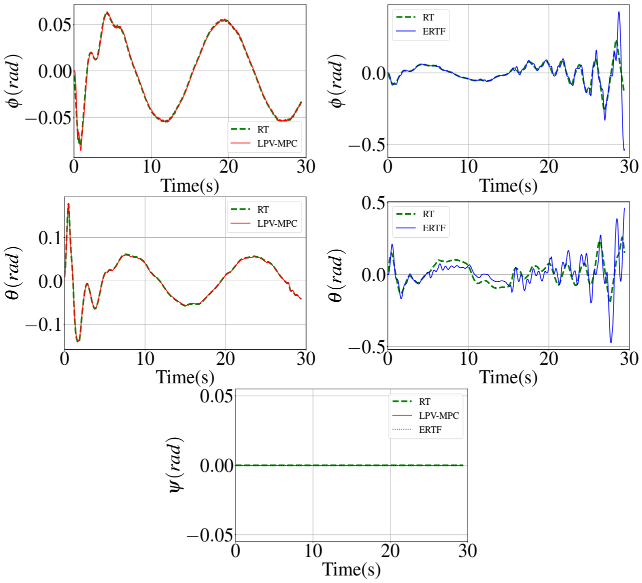

Fig. 6: Attitude responses of the UAV

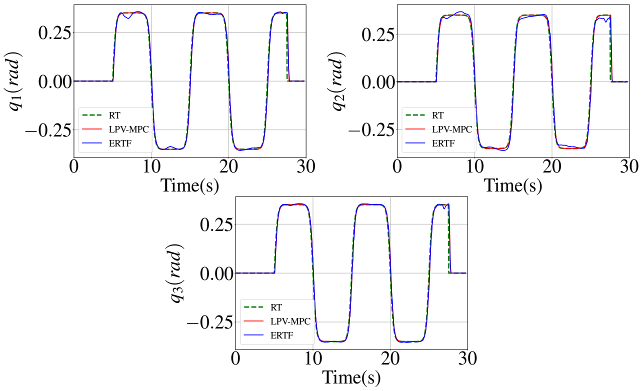

Fig. 7: Position response of the manipulator

the ERTF approach. As an example, in figures illustrating the ϕ motion, the ERTF response begins to significantly diverge from the reference trajectory at about 15s, coinciding with the initiation of a new arm movement (see Fig. 7). This suggests that the ERTF approach struggles to accurately consider the cross-coupled dynamics in the calculation of reaction torques, particularly in scenarios requiring rapid trajectory tracking. Additionally, since the initial joint movement occurs about the y axis, it significantly impacts the UAV's θ, posing a greater challenge for the ERTF approach to offset the corresponding reaction torque applied to the UAV. This challenge becomes particularly evident during the initial movement of the manipulator at about 5-second mark (refer to Fig. 7).

In contrast, the LPV-MPC approach demonstrates a more successful adherence to the reference trajectory (RT), showing a rapid reduction in tracking error, especially during intervals such as $[5\ 10]s$ and $[20\ 25]s$ when there is a slight deviation from the RT. This highlights the LPV-MPC approach's robustness to the varying parameters and capability to effectively respond to deviations and promptly mitigate tracking errors by updating the UAV's model based on the manipulator's movement. As reported in table (III), the LPV-MPC approach provides a significantly lower IAE performance index for the UAV's attitude states than the ERTF approach, indicating a more stable attitude dynamic for the UAV that results in a lower tracking error for the UAV's translational and manipulator dynamics as well.

The sub-figures presented in Fig. 7 illustrate the movements of the manipulator's joints as they are tracking the prescribed trajectories. Each joint is anticipated to execute rapid and periodic movements, featuring an amplitude of 0.33 radians and a period of 5 seconds. The comparative results between LPV-MPC and ERTF reveal a better performance of the former, suggesting that the manipulator's motion is less adversely influenced by cross-coupled dynamics within the LPV-MPC approach. The better performance is because of the enhanced stability of the UAV's rotational dynamics in the LPV-MPC approach, enabling the manipulator to more effectively track the reference trajectory (RT) compared to the ERTF approach. Furthermore, the IAE index for each joint angle, as detailed in table (III), indicates a moderately lower error for the LPV-MPC approach.

The state trajectories evolution for the UAV's rotational dynamics (p and q) and the manipulator ($\dot{q}_1$, $\dot{q}_2$, and $\dot{q}_3$) within a 0 to 15 second time interval are depicted in Fig. 8 and Fig. 9, respectively. The nominal system trajectory is represented by the black line, while the actual system trajectory is shown by the blue line. Also, the bounded robust tube is denoted by the blue area confined between the green and brown lines. Both the actual and nominal system trajectories demonstrate robustness to uncertainties as they remain successfully enclosed within the robust tubes. Particularly as the manipulator movement starts at the 5-second mark, the evolution of the robust tubes is evident in Fig. 8. The UAV's rotational dynamics' controller effectively guides the actual trajectory along the nominal trajectory within the provided tubes, ensuring the robustness of the system without any violations. Similarly, Fig. 9 indicates that the manipulator's controller successfully steers the actual

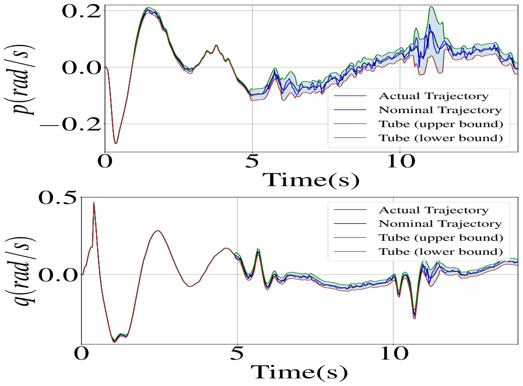

Fig. 8: Actual and nominal trajectories evolutions through the bounded robust tube for p and q

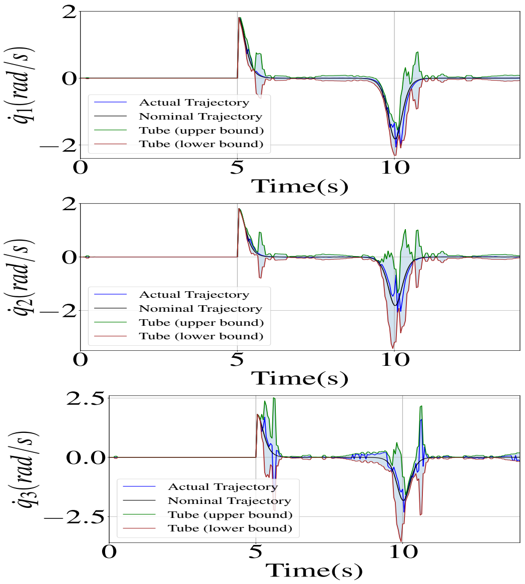

Fig. 9: Actual and nominal trajectories evolutions through the bounded robust tube for $\dot{q}_1$, $\dot{q}_2$, and $\dot{q}_3$

and nominal trajectory within the robust tube despite the UAV's maneuvers.

To further demonstrate the significant performance boost of the presented LPV-MPC approach, a more demanding RT is assumed for the UAM. The amplitude of this new trajectory is nearly doubled, and the frequency is increased by 50 percent compared to the RT used in the previous scenario. Additionally, the manipulator is tasked with tracking a swifter trajectory, featuring an amplitude of 0.55 radians (see sub-figures 13). The three-dimensional UAM's motion is visualized in Fig. 10. Furthermore, the UAM's movements along the x, y, and z axes are depicted in sub-figures 11.

Despite the challenging RT provided, the LPV-MPC approach adeptly follows the trajectory, maintaining an acceptable tracking error throughout the duration. Handling cross-coupled dynamic effects becomes notably more challenging in the context of rapid trajectory tracking, especially during manipulator movements. An illustrative example is evident in the sub-figure related to the z axis in 11, where discernible bumps occur in the trajectory at 5-second intervals. These bumps signify the reaction force associated with the manipulator's movement on the UAV. Remarkably, the LPV-MPC approach effectively addresses and mitigates these dynamic effects, showing its capability to handle challenges in fast-paced trajectory tracking scenarios.

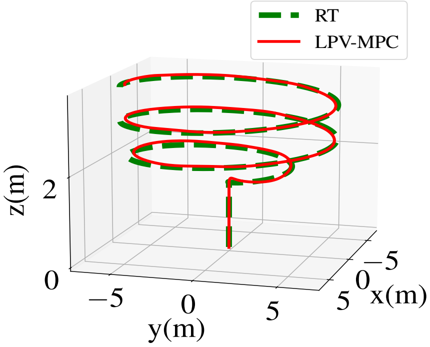

Fig. 10: 3D trajectory tracking

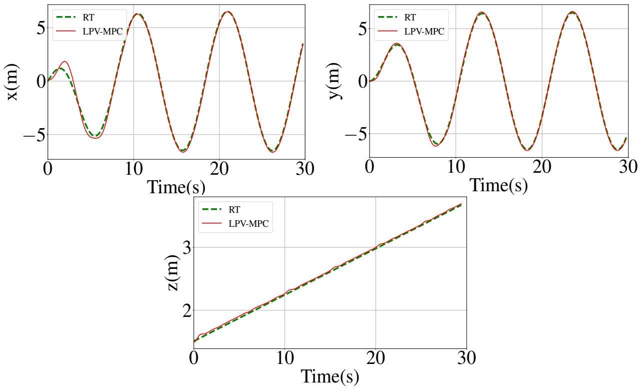

Fig. 11: Position response of the UAV

Additionally, the significant instances occurring at 5-second intervals are noticeable in the UAV's attitude

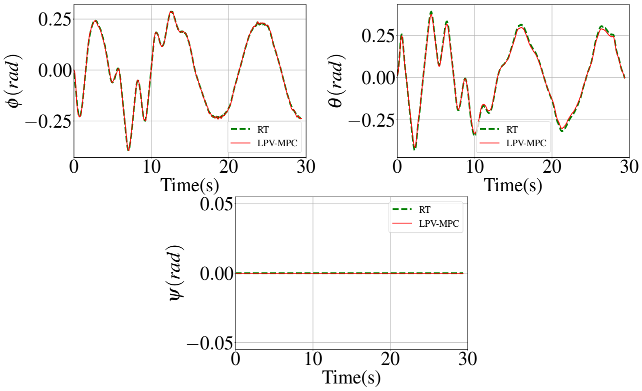

Fig. 12: Attitude responses of the UAV

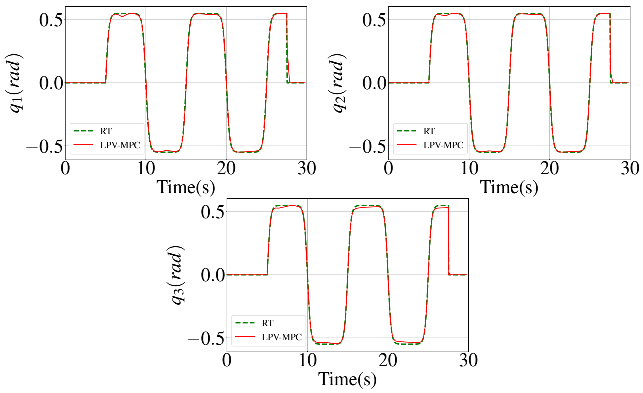

Fig. 13: Position response of the manipulator

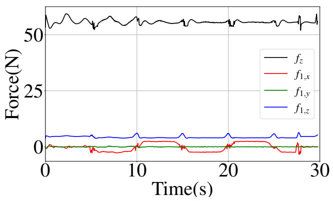

Fig. 14: UAV's force vector and estimated reaction force vector

changes, as depicted in sub-figures 12. These figures reveal an increased magnitude compared to the earlier scenario, yet the LPV-MPC approach consistently maintains its ability to track the desired reference trajectory (RT) with favourable performance. Specifically, in the case of the UAV's θ motion, which is notably influenced by the manipulator's joint movements, the LPV-MPC approach actively works to reduce the tracking error, especially during these critical moments. Also, the tracking performance observed in the manipulator joint movements, as depicted in sub-figures 13, indicates that the LPV model formulated for the manipulator effectively incorporates all cross-coupled dynamics exerted by the UAV, and the LPV-MPC controller guides the manipulator joints along the RT with an acceptable tracking error.

Lastly, the responses of the UAV's force and torque vectors are depicted in Figs. 14 and 15, respectively, alongside the corresponding reaction force and torque vectors imposed on the UAV by the manipulator. As illustrated in Fig. 15, since the first manipulator joint rotates about the y axis, the reaction torque about the y axis exhibits greater variation compared to the other components of the reaction torque vector. Moreover,the reaction force along the x axis varies significantly in response to the manipulator's movement, as depicted in Fig. 14. The UAV's force component response along the z axis in Fig. 14 effectively compensates for the reaction forces. Similarly, the UAV's torque vector responses in Fig. 15 vary in accordance with the trend of reaction torque response and manipulator movement, demonstrating that the suggested method performs efficiently for dealing with the system's cross-coupled dynamic effects.

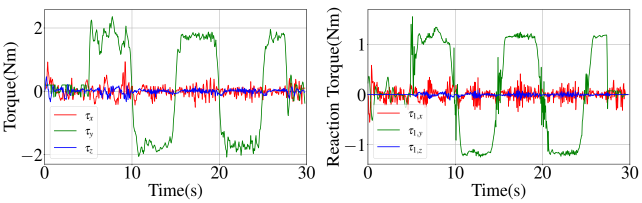

Fig. 15: UAV's torque vector and estimated reaction torque vector

VI. Conclusion

A decoupled approach for the trajectory tracking problem of UAMs is provided in this paper, addressing challenges associated with dynamic modeling and controller design. We proposed a systematic decomposition of reaction torque and force in terms of UAM states, facilitating effective modification of the dynamic model for both the UAV and manipulator. Then we formulated the resulting decoupled dynamic model in the form of LPV systems for both the UAV's rotational dynamics and manipulator. Because the system is parameter-varying, a tube-based MPC controller was employed to take care of the cross-coupled effects as well as the bounded uncertainties in the system. Through simulations, we validated the efficacy of the proposed LPV-MPC methodology, comparing its performance against the ERTF approach. The LPV-MPC approach exhibited improved trajectory tracking accuracy, particularly excelling in stabilizing the UAV's rotational dynamics and achieving significantly lower Integral of Absolute Error (IAE) performance for UAM states compared to the ERTF approach. Furthermore, our results demonstrated the LPV-MPC approach's capability to successfully follow more challenging trajectories with desirable performance. In the future, we would like to implement the LPV-MPC approach on a real UAM in our lab.

REFERENCES

[1] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2668–2673, Sept. 2011. ISSN: 2153-0866.

[2] S. Shimahara, S. Leewiwatwong, R. Ladig, and K. Shimonomura, "Aerial torsional manipulation employing multi-rotor flying robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1595–1600, Oct. 2016. ISSN: 2153-0866.

[3] L. Tianyu, L. Yongzhe, Q. Juntong, M. Xiangdong, and H. Jianda, "Modeling and controller design of hydraulic rotorcraft aerial manipulator," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pp. 5446–5452, May 2015. ISSN: 1948-9447.

[4] D. R. McArthur, A. B. Chowdhury, and D. J. Cappelleri, "Autonomous Control of the Interacting-BoomCopter UAV for Remote Sensor Mounting," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5219–5224, May 2018. ISSN: 2577-087X.

[5] N. Staub, D. Bicego, Q. Sablé, V. Arellano, S. Mishra, and A. Franchi, "Towards a Flying Assistant Paradigm: the OTHex," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6997–7002, May 2018. ISSN: 2577-087X.

[6] A. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *2013 IEEE International Conference on Robotics and Automation*, pp. 4916–4921, May 2013. ISSN: 1050-4729.

[7] H.-N. Nguyen and D. Lee, "Hybrid force/motion control and internal dynamics of quadrotors for tool operation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3458–3464, Nov. 2013. ISSN: 2153-0866.

[8] V. Ghadiok, J. Goldin, and W. Ren, "Autonomous indoor aerial gripping using a quadrotor," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4645–4651, Sept. 2011. ISSN: 2153-0866.

[9] F. Augugliaro and R. D'Andrea, "Admittance control for physical human-quadrocopter interaction," in *2013 European Control Conference (ECC)*, pp. 1805–1810, July 2013.

[10] S. Kannan, M. A. Olivares-Mendez, and H. Voos, "Modeling and Control of Aerial Manipulation Vehicle with Visual sensor*," *IFAC Proceedings Volumes*, vol. 46, pp. 303–309, Jan. 2013.

[11] M. Kobilarov, "Nonlinear Trajectory Control of Multi-body Aerial Manipulators," *Journal of Intelligent & Robotic Systems*, vol. 73, pp. 679–692, Jan. 2014.

[12] P. Ji, C. Li, and F. Ma, "Sliding Mode Control of Manipulator Based on Improved Reaching Law and Sliding Surface," *Mathematics*, vol. 10, p. 1935, Jan. 2022. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[13] B. Yang, Y. Q. He, J. Han, and G. Liu, "Rotor-Flying Manipulator: Modeling, Analysis, and Control," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–13, May 2014.

[14] H. B. Khamseh and F. Janabi-Sharifi, "UKF–Based LQR Control of a Manipulating Unmanned Aerial Vehicle," *Unmanned Systems*, vol. 05, pp. 131–139, July 2017. Publisher: World Scientific Publishing Co.

[15] G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, and A. Ollero, "Control of a multirotor outdoor aerial manipulator," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3417–3422, Sept. 2014. ISSN: 2153-0866.

[16] S. Kim, H. Seo, S. Choi, and H. J. Kim, "Vision-Guided Aerial Manipulation Using a Multirotor With a Robotic Arm," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 1912–1923, Aug. 2016.

[17] G. Garimella and M. Kobilarov, "Towards model-predictive control for aerial pick-and-place," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4692–4697, May 2015. ISSN: 1050-4729.

[18] H. Bonyan Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation—A literature survey," *Robotics and Autonomous Systems*, vol. 107, pp. 221–235, Sept. 2018.

[19] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, Present, and Future of Aerial Robotic Manipulators," *IEEE Transactions on Robotics*, vol. 38, pp. 626–645, Feb. 2022. Conference Name: IEEE Transactions on Robotics.

[20] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, "Impedance control of VToL UAVs with a momentum-based external generalized forces estimator," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2093–2099, May 2014. ISSN: 1050-4729.

[21] F. Ruggiero, M. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor UAVs equipped with a servo robot arm," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4014–4020, May 2015. ISSN: 1050-4729.

[22] D. Wuthier, D. Kominiak, E. Fresk, and G. Nikolakopoulos, "A Geometric Pulling Force Controller for Aerial Robotic Workers*," *IFAC-PapersOnLine*, vol. 50, pp. 10287–10292, July 2017.

[23] 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, pp. 331–338, Apr. 2020. Publisher: IEEE.

[24] A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismail, "Modeling and control of a new quadrotor manipulation system," in *2012 First International Conference on Innovative Engineering Systems*, pp. 109–114, Dec. 2012.

[25] M. Yavari, K. Gupta, and M. Mehrandezh, "Interleaved Predictive Control and Planning for an Unmanned Aerial Manipulator With On-The-Fly Rapid Re-Planning in Unknown Environments," *IEEE Transactions on Automation Science and Engineering*, pp. 1–16, 2022.

[26] S. Kim, S. Choi, H. Kim, J. Shin, H. Shim, and H. J. Kim, "Robust Control of an Equipment-Added Multirotor Using Disturbance Observer," *IEEE Transactions on Control Systems Technology*, vol. 26, pp. 1524–1531, July 2018.

[27] M. Orsag, C. Korpela, M. Pekala, and P. Oh, "Stability control in aerial manipulation," in *2013 American Control Conference*, pp. 5581–5586, June 2013. ISSN: 2378-5861.

[28] M. Kamel, K. Alexis, and R. Siegwart, "Design and modeling of dexterous aerial manipulator," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4870–4876, Oct. 2016. ISSN: 2153-0866.

[29] S. Di Lucia, G. D. Tipaldi, and W. Burgard, "Attitude stabilization control of an aerial manipulator using a quaternion-based backstepping approach," in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6, Sept. 2015.

[30] M. Fanni and A. Khalifa, "A New 6-DOF Quadrotor Manipulation System: Design, Kinematics, Dynamics, and Control," *IEEE/ASME Transactions on Mechatronics*, vol. 22, pp. 1315–1326, June 2017.

[31] A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismail, "Adaptive Intelligent Controller Design for a New Quadrotor Manipulation System," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1666–1671, Oct. 2013. ISSN: 1062-922X.

[32] L. Fang, H. Chen, Y. Lou, Y. Li, and Y. Liu, "Visual Grasping for a Lightweight Aerial Manipulator Based on NSGA-II and Kinematic Compensation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3488–3493, May 2018. ISSN: 2577-087X.

[33] S. A. Emami and A. Banazadeh, "Simultaneous trajectory tracking and aerial manipulation using a multi-stage model predictive control," *Aerospace Science and Technology*, vol. 112, p. 106573, May 2021.

[34] H. Cao, Y. Wu, and L. Wang, "Adaptive NN motion control and predictive coordinate planning for aerial manipulators," *Aerospace Science and Technology*, vol. 126, p. 107607, July 2022.

[35] M. Yavari, K. Gupta, M. Mehrandezh, and A. Ramirez-Serrano, "Optimal Real-Time Trajectory Control of a Pitch-Hover UAV with a Two Link Manipulator," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 930–938, June 2018. ISSN: 2575-7296.

[36] M. Yavari, K. Gupta, and M. Mehrandezh, *Lazy Steering RRT*: An Optimal Constrained Kinodynamic Neural Network Based Planner with no In-Exploration Steering*. Dec. 2019. Pages: 407.

[37] M. Orsag, C. M. Korpela, S. Bogdan, and P. Y. Oh, "Hybrid Adaptive Control for Aerial Manipulation," *Journal of Intelligent & Robotic Systems*, vol. 73, pp. 693–707, Jan. 2014.

[38] X. Meng, Y. He, F. Gu, Q. Li, and J. Han, "Dynamics modeling and simulation analysis for rotorcraft aerial manipulator system," in *2017 36th Chinese Control Conference (CCC)*, pp. 1156–1161, July 2017. ISSN: 1934-1768.

[39] G. Zhang, Y. He, F. Gu, J. Han, and G. Liu, "Varying inertial parameters model based robust control for an aerial manipulator," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 696–701, Dec. 2016.

[40] G. Zhang, Y. He, B. Dai, F. Gu, L. Yang, J. Han, G. Liu, and J. Qi, "Grasp a Moving Target from the Air: System & Control of

an Aerial Manipulator," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1681–1687, May 2018. ISSN: 2577-087X.

[41] 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)*, pp. 87–93, June 2017.

[42] E. Fresk, D. Wuthier, and G. Nikolakopoulos, "Generalized center of gravity compensation for multirotors with application to aerial manipulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4424–4429, Sept. 2017. ISSN: 2153-0866.

[43] G. Zhang, Y. He, B. Dai, F. Gu, J. Han, and G. Liu, "Robust Control of an Aerial Manipulator Based on a Variable Inertia Parameters Model," *IEEE Transactions on Industrial Electronics*, vol. 67, pp. 9515–9525, Nov. 2020.

[44] Y. Becerra-Mora and S. Soto-Gaona, "Tracking and Grasping of Moving Objects Using Aerial Robotic Manipulators: A Brief Survey," *Revista UIS Ingenierías*, vol. 22, Nov. 2023.

[45] R. Beard, "Quadrotor Dynamics and Control Rev 0.1," *Faculty Publications*, May 2008.

[46] S. Bouabdallah, *Design and control of quadrotors with application to autonomous flying*. PhD thesis, EPFL, Lausanne, 2007.

[47] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, p. 22, 2011.

[48] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Pearson/Prentice Hall, 2005. Google-Books-ID: MqMeAQAAIAAJ.

[49] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, "Sympy: symbolic computing in python," *PeerJ Computer Science*, vol. 3, p. e103, Jan. 2017.

[50] A. Eskandarpour and I. Sharf, "A constrained error-based MPC for path following of quadrotor with stability analysis," *Nonlinear Dynamics*, vol. 99, Jan. 2020.

[51] A. Eskandarpour, M. Mehrandezh, K. Gupta, A. Ramirez-Serrano, and M. Soltanshah, "A constrained robust switching MPC structure for tilt-rotor UAV trajectory tracking problem," *Nonlinear Dynamics*, vol. 111, pp. 17247–17275, Sept. 2023.

[52] A. Kwiatkowski, M.-T. Boll, and H. Werner, "Automated Generation and Assessment of Affine LPV Models," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 6690–6695, Dec. 2006. ISSN: 0191-2216.

[53] S. M. Hashemi, H. S. Abbas, and H. Werner, "Low-complexity linear parameter-varying modeling and control of a robotic manipulator," *Control Engineering Practice*, vol. 20, pp. 248–257, Mar. 2012.

[54] R. Gonzalez, M. Fiacchini, T. Alamo, J. Guzman, and F. Rodriguez, "Online robust tube-based MPC for time-varying systems: a practical approach," *International Journal of Control*, vol. 84, pp. 1157–1170, June 2011.

[55] P. Goulart, E. Kerrigan, and J. Maciejowski, "Optimization Over State Feedback Policies for Robust Control with Constraints," *Automatica*, vol. 42, pp. 523–533, Apr. 2006.

[56] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust tube-based MPC for tracking of constrained linear systems with additive disturbances," *Journal of Process Control*, vol. 20, pp. 248–260, Mar. 2010.

[57] R. Scattolini and P. Colaneri, "Hierarchical model predictive control," in *2007 46th IEEE Conference on Decision and Control*, pp. 4803–4808, Dec. 2007. ISSN: 0191-2216.

[58] A. Bemporad, C. A. Pascucci, and C. Rocchi, "Hierarchical and Hybrid Model Predictive Control of Quadcopter Air Vehicles," *IFAC Proceedings Volumes*, vol. 42, pp. 14–19, Jan. 2009.

[59] J. Rawlings and D. Mayne, "Postface to " Model Predictive Control : Theory and Design "," 2012.

[60] H. S. Abbas, G. Männel, C. Herzog né Hoffmann, and P. Rostalski, "Tube-based model predictive control for linear parameter-varying systems with bounded rate of parameter variation," *Automatica*, vol. 107, pp. 21–28, Sept. 2019.

[61] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," *URL: http://control.ee.ethz.ch/ mpt/*, 2004.