Dynamic Control of a Macro–Mini Aerial Manipulator With Elastic Suspension

Arda Yigit, Loïc Cuvillon, Miguel Arpa Perozo, Sylvain Durand, Jacques Gangloff

Dynamic Control of a Macro-Mini Aerial Manipulator with Elastic Suspension

Arda Yiğit, Loïc Cuvillon, Miguel Arpa Perozo, Sylvain Durand and Jacques Gangloff

***Abstract*—In this work, a macro-mini aerial manipulator with elastic suspension is introduced. The mini is an omnidirectional aerial manipulator suspended from the macro by a spring. The macro is a Cartesian robot that moves the anchoring point of the spring. This design combines the advantages of the large workspace of the macro robot with the high dynamics of aerial vehicles, while reducing energy consumption thanks to gravity compensation.**

A partitioned control scheme is first implemented to regulate the aerial manipulator and its carrier separately. The redundancy resolution strategy positions the macro robot to minimize the energy consumption of the aerial manipulator at steady state. Then, a nonlinear model predictive controller replaces the partitioned controller to improve further the efficiency of the combined system, notably by anticipating the slow dynamics of the macro robot. A sufficient condition for offset-free tracking has been investigated theoretically. Experiments with a cable-driven parallel robot as macro are carried out to assess the added value of the carrier. Both controllers are validated and compared experimentally.

***Index Terms*—Aerial manipulation, macro-mini robot, dynamic control, nonlinear model predictive control, cable-driven parallel robot.**

I. Introduction

AERIAL manipulators (AMs) use unmanned aerial vehicles (UAVs) to perform robotic tasks within a very large workspace. Various configurations are possible in order to deal with the trade-off between energy efficiency and dexterity [1]–[3]. Simple tasks such as pick and place can be performed using flying hands, where a gripper is attached to a UAV (e.g., [4]). For tasks requiring more dexterity, the aerial vehicle is often equipped with one or more robotic arms (e.g., [5], [6]). Multirotor aerial vehicles can be classified according to the number, type and arrangement of the propellers [7]. On fully-actuated aerial vehicles, the resulting thrust and moment can be independently controlled along all directions. In addition, if they can hover regardless of the orientation, they are referred to as omnidirectional. Such aerial vehicles can be used as flying hands. They can achieve dexterous tasks without using an additional manipulator, thus simplifying the mechanical structure and improving the frugality of the design. This also reduces the inertia of the aerial vehicle, favoring applications requiring safe physical interaction. These vehicles can either use non-steerable thrusters (e.g., the omnidirectional multirotor vehicle from Brescianini and D'Andrea [8] or the ODAR from Park et al. [9]) or steerable thrusters (e.g., the FAST-Hex from Ryll et al. [10] or the OMAV from Brunner et al. [11]). Without steering mechanism, the vehicle usually needs more thrusters to achieve the same controllability, but with a simpler mechanical design. While six bidirectional non-steerable thrusters are sufficient to control all six degrees of freedom (DoFs) of an omnidirectional AM, some redundancy with at least seven thrusters is needed when using unidirectional thrusters [12].

Aerial manipulators consume a lot of energy to compensate for the gravity while hovering. This drastically reduces their flight time when running on batteries. With a tether [13], the flight time may theoretically be infinite but at the cost of some limitations in the workspace.

The suspension of an aerial vehicle reduces its energy consumption thanks to gravity compensation. SpiderMAV launches anchors on nearby ceiling and walls to perch and stabilize the quadrotor platform in a spider-inspired way [14]. The cable-suspended aerial manipulator SAM [15] is an example of an omnidirectional AM suspended by winch-actuated cables that carries a 7-DoF serial manipulator. On the SAM, the platform holding the thrusters is the end effector of a suspended cable-driven parallel robot (CDPR). The architecture of this CDPR is original since all the cables share a unique anchoring point: thus it can also be seen as a reconfigurable sling. On the SAM, cable actuation is used to regulate the orientation of the platform by compensating for the effects of the manipulator center of mass displacement. Since the propellers can generate any wrench on the platform, the design has actuation redundancy. This property can be used, for instance, to increase the stiffness of the robot as in [16].

For systems using actuators with different dynamics, specific control allocation algorithms have been developed. Frequency-apportioned control allocation uses a weighted pseudo-inverse of the Jacobian matrix to consider both the saturation and the frequency response of the actuators [17]. Model predictive control allocation takes into account actuator constraints, namely their dynamics and saturation, and solves an optimization problem in real time to distribute control inputs [18]. More recently, this last strategy was evaluated on a planar CDPR equipped with propellers [19].

Model predictive control has been applied successfully in the past to aerial vehicles, both in its linear and nonlinear forms [20]. To face the problem of steady-state offset, due

This work was supported by the e-VISER project under grant ANR-17-CE33-0008 by the French National Research Agency (ANR).

The authors are with the ICube Laboratory, University of Strasbourg, INSA Strasbourg, 67000, Strasbourg, France. E-mails: {arda.yigit, l.cuvillon, marpaperozo, sdurand, jacques.gangloff}@unistra.fr

for example to modelling errors, the prediction model can be augmented to include constant disturbances [21]–[23]. A state observer then estimates the constant disturbances. Offset-free steady state can also be achieved using the so-called delta-input formulation, in which the variation of the control input becomes the new input of the prediction model. Constant disturbances are then incorporated in the estimate of the control input by the state observer. Some omnidirectional aerial vehicles have already used the delta-input formulation [11], [24], [25]. Most recently, a similar approach combining MPC and incremental nonlinear dynamic inversion (INDI) has been proposed for quadrotors, reducing significantly the tracking error during challenging trajectories [26]. No proof was proposed to guarantee an offset-free steady state for the aerial vehicles.

The Aerial Manipulator with Elastic Suspension (AMES) is an alternative solution proposed by our group to the reduced flight time of AMs [27]. An AM is suspended by a single spring link from the supporting structure or robotic carrier. For a given carrier position, the spring compliance enables the AM to move in any space direction, contrary to a stiff cable that would restrict the AM DoFs. The AM can be driven away from its equilibrium pose with six DoFs by using the propeller thrusts. The AMES constitutes a macro-mini system: it combines the large workspace of the carrier with the fast dynamics, the low inertia and the high accuracy of the AM.

The development of the AMES was motivated by real-world applications involving robotic tasks covering a large workspace – larger than with industrial manipulators but smaller than with unmanned aerial manipulators – with millimetric repeatability and fast dynamics. These tasks include pick and place of parcels in logistic hubs, automated ship building, 3D printing for construction, washing, painting or inspection of large vertical surfaces. It is possible to find an appropriate static or mobile anchoring point for the spring in all these situations: a gantry crane in a logistic hub or on a shipyard, a tower crane for 3D construction or a cable-suspended platform for operations on building vertical surfaces. To widen even more the workspace of the AMES, it may also be possible to use an aerial robot with better autonomy for the carrier, using for example fuel cells, as proposed for the SAM [15]. Recently, the French National Research Agency funded two projects involving the AMES: TIR4sTREEt for urban climatology and STRAD for street art. In addition to better energy efficiency, another significant advantage of the AMES with respect to other aerial manipulators is that there are fewer legal restrictions for its use since it maintains a physical link with the ground (see [28] for France).

In a preliminary work using a static anchoring point, experimental results showed that the AMES is able to achieve millimetric repeatability and fast dynamics [27]. A computed torque control law was proposed, and its stability was proven using singular perturbation theory. Then, we added a suspended CDPR carrier to the AMES and implemented a partitioned controller to control the whole system [29]. The energy consumption of the combined system was reduced since the CDPR moves the equilibrium point synchronously with the AM such that, at steady state, the propulsion units generate thrust only to maintain a desired orientation.

The present work is an evolution of a previous one [29]. The partitioned controller introduced in [29] for the AMES system with an actuated carrier is briefly recalled before being compared to a newly proposed model predictive controller (NMPC). Actually, experiments revealed that the previously used approach yields high power consumption of the AM during transients. Indeed, since the CDPR had slower dynamics compared to the aerial vehicle, it is lagging behind during transients. As a consequence, the elastic restoring force of the spring creates a moment at the center of mass of the robot which yields high control inputs. Here, the model of the robot and the controller are improved to consider the dynamics of a generic macro-mini AM with elastic suspension, where the macro is a velocity controlled slow Cartesian robot and the mini is a generic omnidirectional aerial vehicle with non-steerable thrusters. A nonlinear model predictive controller (NMPC) is developed to reduce the energy consumption of the aerial vehicle (i) by anticipating the slow carrier dynamics and (ii) by generating smoother control inputs as highlighted in a previous work [25]. A sufficient condition to eliminate steady-state offset with the NMPC has been investigated theoretically. The sufficient condition extends the work of Morari and Maeder [22] to continuous plant models and the delta-input formulation. Hence, the AMES is robust to parameter errors and measurement errors on the carrier position. The performance of the AMES is assessed experimentally, showing the benefits of the NMPC. Thus, the main contributions of this paper are: (i) nonlinear model predictive control of a macro-mini AM with elastic suspension minimizing energy consumption while maximizing AM dynamics and accuracy, (ii) demonstration of a sufficient condition to achieve offset-free steady state with NMPC. Various experimental results are provided to validate and compare the proposed controllers.

The paper is organized as follows. The model of the AMES is first derived in Section II. Then, the partitioned controller and the offset-free NMPC are respectively presented in Sections III and IV. The experimental setup is shown in Section V. The results with both controllers are presented in Section VI. The conclusion and some perspectives are discussed in Section VII.

II. Modeling

A. Parametrization and Notations

The geometric parameters of the AMES are shown in Fig. 1.

Let $\mathcal{R}_f = (O, \boldsymbol{x}_f, \boldsymbol{y}_f, \boldsymbol{z}_f)$ be an inertial frame with O its origin. Let $\mathcal{R}_b = (G, \boldsymbol{x}_b, \boldsymbol{y}_b, \boldsymbol{z}_b)$ be a moving frame attached to the AM and positioned at its center of mass (CoM) G with $\boldsymbol{z}_b$ pointing toward the onboard anchoring point A of the spring. In the case where the CoM G coincides with the anchoring point A, vector $\boldsymbol{z}_b$ can be selected freely. The rotation matrix $\boldsymbol{R}_{fb} \in SO(3)$ describes the orientation of $\mathcal{R}_b$ with respect to $\mathcal{R}_f$. The AM has n_p identical propulsion units. The center of the i-th propulsion unit is B_i, and $\boldsymbol{u}_i$ is the unit thrust direction vector which is constant in the body frame $\mathcal{R}_b$. The thrust τ_i generated by the i-th propulsion

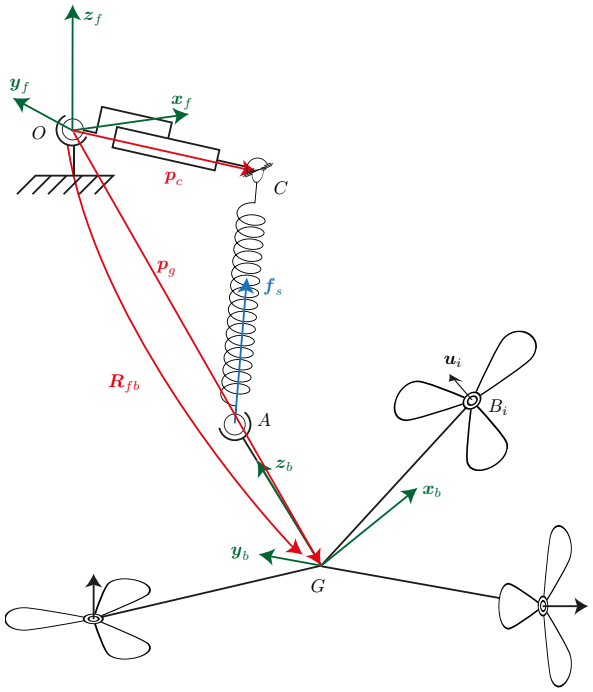


Fig. 1. Parameters of the AMES.

unit is proportional to the square of the rotational speed w_i: $\tau_i = a w_i |w_i|$, with $a > 0$ the thrust coefficient. The drag σ_i is proportional to the thrust τ_i: $\sigma_i = -\epsilon_i b \tau_i$, with $\epsilon_i \in \{-1,\ 1\}$ depending on the directionality (clockwise or counterclockwise) of the propeller and $b \geq 0$ [30]. The carrier is considered as a 3-DoF Cartesian robot. Its end effector coincides with the upper anchoring point C of the spring. The force applied by the spring on the AM is noted $\boldsymbol{f}_s$. The spring restoring force is the only coupling between the AM and the carrier dynamics, and it can be expressed as

$$\boldsymbol{f}_s = k\left(d_{AC} - l_0\right)\frac{\boldsymbol{p}_c - \boldsymbol{p}_a}{d_{AC}} \tag{1}$$

with l_0 the free length and k the stiffness of the spring. Let M, E and F be generic points. The vector pointing from the origin O of the inertial frame $\mathcal{R}_f$ to a point M is noted $\boldsymbol{p}_m$. The distance between two points E and F is noted d_{EF}.

Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be vectors and $\mathcal{R}_q$ a reference frame. The projection of $\boldsymbol{v}$ in $\mathcal{R}_q$ is written ${}^q\boldsymbol{v}$. The cross product of ${}^q\boldsymbol{u}$ with ${}^q\boldsymbol{v}$ is noted ${}^q\boldsymbol{u} \times {}^q\boldsymbol{v}$ and the cross-product matrix $[\cdot]_\times$ is defined such that ${}^q\boldsymbol{u} \times {}^q\boldsymbol{v} = [{}^q\boldsymbol{u}]_\times {}^q\boldsymbol{v}$.

To simplify the notations, unless otherwise stated, the position vectors $\boldsymbol{p}_g$ and $\boldsymbol{p}_c$ are expressed in the inertial frame $\mathcal{R}_f$, and, therefore, the superscripts are omitted: $\boldsymbol{p}_g = {}^f\boldsymbol{p}_g$ and $\boldsymbol{p}_c = {}^f\boldsymbol{p}_c$.

B. Rotation Representation

Roll (ψ), pitch (θ) and yaw (ϕ) angles describe the rotation of the body frame $\mathcal{R}_b$ with respect to the inertial frame $\mathcal{R}_f$: $\boldsymbol{\eta} = [\psi\ \theta\ \phi]^T$. The chosen representation of rotations, namely the roll-pitch-yaw angles, suffers from a gimbal lock like any other set of Euler angles, corresponding to a loss of one DoF when $\theta = +\frac{\pi}{2}$ and $\theta = -\frac{\pi}{2}$. This singularity is avoided if the range of the pitch angle θ is restricted to $|\theta| < \frac{\pi}{2}$, which will be the case in practice for the AM attitude. Indeed, a pitch angle θ superior or equal to $\frac{\pi}{2}$ is not relevant for the manipulation tasks considered here and prohibited due to possible interference of the AM body with the suspension spring. Therefore, the rotation matrix used in the dynamic model is

$$\boldsymbol{R}_{fb}(\boldsymbol{\eta}) = \boldsymbol{R}_{\boldsymbol{z},\phi}\boldsymbol{R}_{\boldsymbol{y},\theta}\boldsymbol{R}_{\boldsymbol{x},\psi} \tag{2}$$

where $\boldsymbol{R}_{\boldsymbol{u},\alpha}$ is the matrix corresponding to a rotation α about the $\boldsymbol{u}$ axis, and $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is the canonical basis of $\mathbb{R}^3$.

The analytical Jacobian matrix $\boldsymbol{S}(\boldsymbol{\eta}) \in \mathbb{R}^{3\times3}$ maps the time derivative of the set of Euler angles $\boldsymbol{\eta}$ to the angular velocity of the AM in the body frame $\mathcal{R}_b$:

$$\boldsymbol{S}(\boldsymbol{\eta}) = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos\psi & \sin\psi\cos\theta \\ 0 & -\sin\psi & \cos\psi\sin\theta \end{bmatrix}. \tag{3}$$

C. AM Dynamics

Let $\boldsymbol{p} = [\boldsymbol{p}_g^T\ \boldsymbol{\eta}^T]^T$ be the pose coordinates of the AM. The dynamic model of the AM is developed in a previous work [27] and is given as follows in the Cartesian-space canonical formulation:

$$\boldsymbol{M}_a(\boldsymbol{p})\ddot{\boldsymbol{p}} + \boldsymbol{C}_a(\boldsymbol{p},\dot{\boldsymbol{p}})\dot{\boldsymbol{p}} + \boldsymbol{g}_a(\boldsymbol{p},\boldsymbol{p}_c) = \boldsymbol{W}_a(\boldsymbol{p})\boldsymbol{\tau} \tag{4}$$

with

$$\begin{cases} \boldsymbol{M}_a(\boldsymbol{p}) &= \begin{bmatrix} m\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{S}^T {}^b\boldsymbol{I}_a \boldsymbol{S} \end{bmatrix} \\ \boldsymbol{C}_a(\boldsymbol{p},\dot{\boldsymbol{p}}) &= \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{S}^T({}^b\boldsymbol{I}_a\dot{\boldsymbol{S}} + [\boldsymbol{S}\dot{\boldsymbol{\eta}}]_\times {}^b\boldsymbol{I}_a\boldsymbol{S}) \end{bmatrix} \\ \boldsymbol{g}_a(\boldsymbol{p},\boldsymbol{p}_c) &= \begin{bmatrix} -m^f\boldsymbol{g} - {}^f\boldsymbol{f}_s \\ -\boldsymbol{S}^T {}^b\boldsymbol{n}_s \end{bmatrix} \\ \boldsymbol{W}_a(\boldsymbol{p}) &= \begin{bmatrix} \boldsymbol{R}_{fb} & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{S}^T \end{bmatrix} {}^b\boldsymbol{W} \end{cases} \tag{5}$$

where m is the total mass of the platform, ${}^b\boldsymbol{I}_a$ its inertia tensor at the CoM expressed in $\mathcal{R}_b$, $\boldsymbol{g} \in \mathbb{R}^3$ the gravity acceleration, $\boldsymbol{f}_s$ the force of the elastic link on the AM and $\boldsymbol{n}_s = (\boldsymbol{p}_a - \boldsymbol{p}_g)\times\boldsymbol{f}_s$ the associated moment at the CoM. The matrix ${}^b\boldsymbol{W} \in \mathbb{R}^{6\times n_p}$ maps the array $\boldsymbol{\tau} = [\cdots\ \tau_i\ \cdots]^T$ of the n_p propeller thrusts τ_i to the wrench they apply on the platform expressed in the moving frame $\mathcal{R}_b$.

The thrusts τ_i are supposed to be regulated with independent low-level velocity loops. The actuator dynamics, which relates the actual thrusts $\boldsymbol{\tau}$ to the reference thrusts $\boldsymbol{\tau}_{ref}$, is modelled as a first-order model with time constant t_τ:

$$\dot{\boldsymbol{\tau}} = \frac{1}{t_\tau}(\boldsymbol{\tau}_{ref} - \boldsymbol{\tau}). \tag{6}$$

D. Carrier Dynamics

If carrier and AM dynamics are in the same range, it is possible to move the carrier almost synchronously with the AM, and thus to save energy by ensuring an almost perfect gravity compensation along the whole trajectory. This work considers a more challenging situation, in which the carrier is

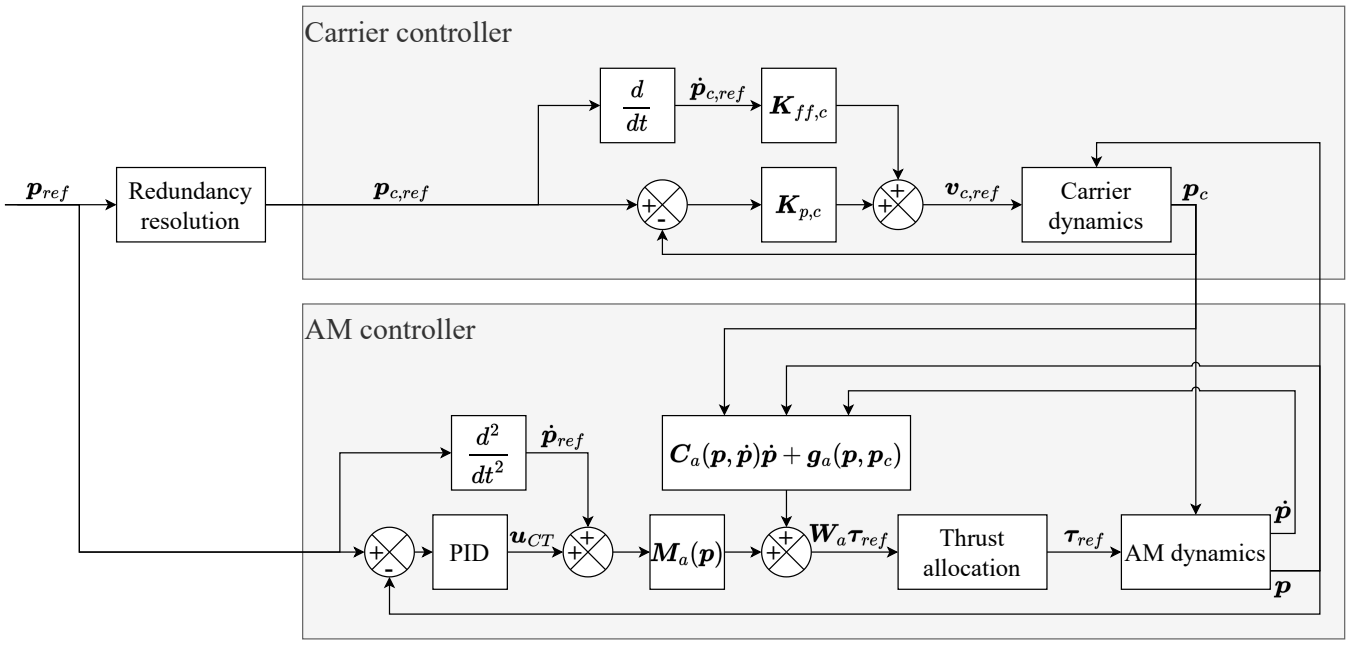


Fig. 2. Partitioned controller of the AMES. Redundancy resolution is explained in Section III-A. Thrust allocation has the form of (13).

slower than the AM. This hypothesis is more realistic when using heavy carriers such as a tower crane or a gantry crane.

The carrier is considered to be velocity controlled and is modelled as a first-order dynamic system:

$$\ddot{\boldsymbol{p}}_c = \frac{1}{t_c}(\boldsymbol{v}_{c,ref} - \dot{\boldsymbol{p}}_c) \tag{7}$$

with $\boldsymbol{v}_{c,ref}$ the reference velocity and t_c the time constant.

The considered carrier dynamics (7) are supposed independent of the spring restoring force $\boldsymbol{f}_s$. This can be achieved, for example, by using actuators with a velocity controller to reject the disturbance induced by $\boldsymbol{f}_s$. Section V-B shows how to obtain this model with a CDPR.

III. Partitioned Control

A partitioned control scheme is proposed to track a desired trajectory $\boldsymbol{p}_{ref} = [\boldsymbol{p}_{g,ref}^T \ \boldsymbol{\eta}_{ref}^T]^T$ with the end effector, which is here at the CoM of the AM. As illustrated in Fig. 2, this partitioned controller consists in using two independent controllers: one for the carrier position and one for the AM pose. The carrier is supposed to be much slower than the AM. Hence, the pose control loop of the AM is tuned to track small displacements with high accelerations while the carrier position loop tracks large amplitude motion with slow acceleration. With this strategy, the AMES has nine degrees of mobility: three for the carrier and six for the AM, yielding a redundant system. Here, the redundancy is solved with an energy criterion by allocating the reference input of the carrier position.

A. Redundancy Resolution

Redundancy resolution often involves the choice of a secondary task for the controlled robot, the main task being the pose control. Here, the secondary task is the minimization of the energy consumption of the aerial vehicle.

The power consumption of a propulsion unit is proportional to the absolute value of its cubic rotational velocity [27]. Let us define $p_i(t) = c|w_i(t)^3|$ the power consumption of the i-th propulsion unit at time t, with c a constant to be identified experimentally. Then, for a given trajectory from time t_0 to time t_1, the total energy consumption E of the AM is

$$E = \int_{t_0}^{t_1} \sum_{i=1}^{i=n_p} p_i(t)dt \tag{8}$$

and can be rewritten as

$$E = c\int_{t_0}^{t_1} \|\boldsymbol{w}_3(t)\|_1 dt \tag{9}$$

with $\boldsymbol{w_3} = [\cdots \ w_i^3 \ \cdots]^T$.

Therefore, the reference input $\boldsymbol{p}_{c,ref}$ for the carrier position controller can be obtained by solving the constrained dynamic optimization problem

$$\min_{\boldsymbol{p}_{c,ref}} \int_{t_0}^{t_1} \|\boldsymbol{w}_3(t)\|_1 dt \tag{10}$$

subject to: (i) the dynamics of the AM (ii) the control allocation strategy on the AM if $n_p > 6$ (iii) the closed-loop dynamics of the carrier and (iv) the actuator limits. State-of-the-art software (e.g., [31]) may be able to solve this problem under some simplifying assumptions, but this is out of the scope of this paper.

Instead, a simpler (but suboptimal) solution is presented here, considering only the steady state. At steady state, the model of the AM (4) becomes $\boldsymbol{g}_a(\boldsymbol{p}_{ref}, \boldsymbol{p}_{c,ref}) = \boldsymbol{W}_a(\boldsymbol{p}_{ref})\boldsymbol{\tau}$. The reference position of the carrier $\boldsymbol{p}_{c,ref}$

is chosen to minimize the total power consumption of the propellers at steady state. Hence,

$$\boldsymbol{p}_{c,ref} = \arg\min_{\boldsymbol{p}_c} \|\boldsymbol{\tau}\|_{\frac{3}{2}}. \tag{11}$$

subject to the steady-state AM model $\boldsymbol{g}_a(\boldsymbol{p}_{ref}, \boldsymbol{p}_{c,ref}) = \boldsymbol{W}_a(\boldsymbol{p}_{ref})\boldsymbol{\tau}$.

B. AM Control

The AM follows a desired trajectory $\boldsymbol{p}_{ref}$ using a computed torque controller (see Fig. 2). The control input $\boldsymbol{\tau}_{ref}$ to be applied is the solution to the following equation:

$$\boldsymbol{W}_a\boldsymbol{\tau}_{ref} = \boldsymbol{M}_a(\ddot{\boldsymbol{p}}_{ref} + \boldsymbol{u}_{CT}) + \boldsymbol{C}_a\dot{\boldsymbol{p}} + \boldsymbol{g}_a \tag{12}$$

with $\boldsymbol{u}_{CT}$ the feedback control input. The existence of a solution is guaranteed since the AM is supposed to be omnidirectional. If $n_p > 6$, i.e., if the AM has more than six propulsion units, there are infinitely many solutions to (12), given by

$$\boldsymbol{\tau}_{ref} = \boldsymbol{W}_a^+\boldsymbol{\tau}_a + \boldsymbol{\tau}_{null} \tag{13}$$

with $\boldsymbol{\tau}_a$ the right-hand side of (12), $\boldsymbol{W}_a^+$ a generalized inverse of $\boldsymbol{W}_a$ (such as the Moore-Penrose inverse $\boldsymbol{W}_a^T(\boldsymbol{W}_a\boldsymbol{W}_a^T)^{-1}$) and $\boldsymbol{\tau}_{null}$ any vector in the nullspace of $\boldsymbol{W}_a$. Different methods exist to allocate the control signal, i.e., to choose a solution to (12), depending on the requirements (e.g., minimal power consumption [8] or unidirectional thrusters [12]).

Computed torque control (CTC) consists in inverting the dynamics of a system in order to obtain linear error dynamics. A linear control strategy can then be applied to the resulting error dynamics. A decoupled PID controller is considered here:

$$\boldsymbol{u}_{CT} = \boldsymbol{K}_p(\boldsymbol{p}_{ref}-\boldsymbol{p})+\boldsymbol{K}_d(\dot{\boldsymbol{p}}_{ref}-\dot{\boldsymbol{p}})+\boldsymbol{K}_i\int(\boldsymbol{p}_{ref}-\boldsymbol{p}) \tag{14}$$

where $\boldsymbol{K}_p$, $\boldsymbol{K}_d$ and $\boldsymbol{K}_i$ are diagonal matrices with positive coefficients. The stability of a computed torque control for the AM with a static anchoring point considering the actuator dynamics has been discussed in a previous work [27].

C. Carrier Control

The tracking of the carrier reference position $\boldsymbol{p}_{c,ref}$ is achieved using a proportional controller with a feedforward term (P+ff)

$$\boldsymbol{v}_{c,ref} = \boldsymbol{K}_{ff,c}\dot{\boldsymbol{p}}_{c,ref} + \boldsymbol{K}_{p,c}(\boldsymbol{p}_{c,ref} - \boldsymbol{p}_c) \tag{15}$$

with $\boldsymbol{K}_{ff,c} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{K}_{p,c} \in \mathbb{R}^{3\times3}$ positive definite gain matrices.

The exponential convergence toward $\boldsymbol{p}_{c,ref}$ of this control scheme using joint velocity inner loops has been first proven for serial robots [32]. Since then, this two-loop cascade control scheme has been successfully applied to redundant robots [33], parallel robots [34] and CDPRs [35].

IV. Nonlinear Model Predictive Control

The partitioned controller presented in previous section uses a basic proportional with feed-forward (P+ff) control loop for the carrier, yielding a significant potential tracking error even with a slowly varying $\boldsymbol{p}_{c,ref}$ as pointed out in a previous work [29]. In order to correctly handle this issue, to be more generic, and also to better take into account nonlinearities like saturations, a global nonlinear model predictive control approach is proposed.

Model predictive control (MPC) uses a model of the plant to predict its future behavior for various inputs. Among these input candidates, the chosen one is the one that minimizes a cost function under constraints on the system inputs and states. Using the model of the plant, MPC allows for compensating for the lag between the AM and the carrier by anticipating the slow carrier dynamics.

Unlike the linear-quadratic regulator (also known as LQR), a constrained optimization problem is solved online and periodically, at each time step. Initially intended for linear systems, it is now possible to deal with nonlinear plants thanks to increased computational power and state-of-the-art solvers.

Despite ongoing research to solve quadratic problems using analog circuits [36], most model predictive control implementations use a digital control strategy. As, in most cases, no explicit or analytical discrete-time model can be derived from a continuous-time nonlinear differential equation, continuous-time plant models for NMPC with numerical integration are used.

A. Standard NMPC Problem Formulation

The nonlinear state-space representation of the real plant to be controlled is not exactly known, but it can be written generically as

$$\begin{cases} \dot{\boldsymbol{x}}_p &= \boldsymbol{f}_p(\boldsymbol{x}_p, \boldsymbol{d}_p, \boldsymbol{u}) \\ \boldsymbol{y}_p &= \boldsymbol{g}_p(\boldsymbol{x}_p, \boldsymbol{d}_p, \boldsymbol{u}) \end{cases} \tag{16}$$

where $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ is the input of the system and $\boldsymbol{y}_p(t) \in \mathbb{R}^{n_y}$ the measured outputs. The functions $\boldsymbol{f}_p$ and $\boldsymbol{g}_p$, and the state $\boldsymbol{x}_p$ are unknown. The disturbance $\boldsymbol{d}_p(t)$ can be considered as an unknown additional input of the system.

Let

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{y} &= \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \end{cases} \tag{17}$$

be the nominal model of our system (including uncertainties and unmodelled dynamics), with $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ its state and $\boldsymbol{y}(t) \in \mathbb{R}^{n_y}$ its output.

The steady-state target $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}})$ for the NMPC controller is defined as an equilibrium of the nominal plant model for a desired reference $\boldsymbol{r}$:

$$\begin{cases} \boldsymbol{0} &= \boldsymbol{f}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}) \\ \boldsymbol{r} &= \boldsymbol{g}(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}) \end{cases}. \tag{18}$$

The NMPC problem is then formulated as

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{x}} & \left[\int_0^T l(\boldsymbol{x}(t)-\overline{\boldsymbol{x}},\boldsymbol{u}(t)-\overline{\boldsymbol{u}})dt\right] \\ \text{subject to } & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \\ & \boldsymbol{x}(0) = \boldsymbol{x_0} \\ & \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) \geq \boldsymbol{0} \end{aligned} \tag{19}$$

with $l \geq 0$ the running cost (often a quadratic form) and $\boldsymbol{h}$ a constraint function. Note that, in this work, the stability of the control is assessed experimentally. Grimm et al. show that a long enough finite horizon size T is sufficient to ensure the closed-loop stability using NMPC [37].

B. Steady-State Offset and Delta-Input Formulation

External disturbance and unmodelled or inaccurate dynamics can cause a steady-state error. There is no guarantee that the optimal control input will enforce a zero steady-state error if the NMPC problem is solved based on an inaccurate model of the plant dynamics. Indeed, in case of a mismatch between the implemented plant model (17) and the real system (16) at steady state, the equilibrium $(\overline{\boldsymbol{x}},\overline{\boldsymbol{u}})$ may not correspond to the minimum of the cost function, which, in turn, may yield a steady-state error.

Various proofs exist about the cancellation of the steady-state error by augmenting the nominal model with constant disturbances [22], [23]. This approach introduces additional degrees of freedom to match the steady states of the plant and of its model, avoiding an offset. To the best of our knowledge, all existing proofs consider a discrete-time model of the plant, which prevents testing sufficient conditions for many nonlinear systems since there is no closed-form discrete formulation for most nonlinear systems.

The model of the plant can be augmented with integrators on the system inputs in order to penalize the variation of the control input signal instead of its absolute value in the cost function. As a consequence, it is possible to obtain a smoother control input, which reduces the core losses in the electric motors and improves the energy efficiency of the AM [25]. Maeder et al. showed for discrete linear systems that these integrators can replace constant disturbances in the augmented model and named this approach "delta-input formulation" [21]. Other works exist using augmented models with or without delta-input formulation on aerial vehicles, however, no proof was proposed to guarantee offset-free steady state [11], [24], [25].

In this work, we adapt the proof proposed by Morari and Maeder [22] to plants whose models are written with continuous-time nonlinear differential equations and use the delta-input formulation.

Let

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \\ \dot{\boldsymbol{u}} &= \boldsymbol{u}_\Delta \\ \boldsymbol{y} &= \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}) \end{cases} \tag{20}$$

be the augmented plant model with the delta-input formulation, with $\boldsymbol{u}_\Delta$ the new input of the augmented system.

Let us introduce the following observer of the augmented system

$$\begin{bmatrix} \dot{\hat{\boldsymbol{x}}} \\ \dot{\hat{\boldsymbol{u}}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}) + \boldsymbol{l}_x(\boldsymbol{y}_p - \boldsymbol{g}(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}})) \\ \boldsymbol{u}_\Delta + \boldsymbol{l}_u(\boldsymbol{y}_p - \boldsymbol{g}(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}})) \end{bmatrix} \tag{21}$$

with $\boldsymbol{l}_x$ and $\boldsymbol{l}_u$ functions designed to ensure the stability of the observer. Note that this observer estimates both the state $\boldsymbol{x}$ of the nominal model (17) and the control signal $\boldsymbol{u}$. As it will be shown below, the estimate $\hat{\boldsymbol{u}}$ contains not only the control signal $\boldsymbol{u}$ but also the disturbances induced by the model uncertainties, which is source of a steady-state error.

The optimal control problem (OCP) is modified as

$$\begin{aligned} \min_{\boldsymbol{u}_\Delta,\boldsymbol{x}} & \left[\int_0^T l(\boldsymbol{x}(t)-\overline{\boldsymbol{x}},\boldsymbol{u}(t)-\overline{\boldsymbol{u}},\boldsymbol{u}_\Delta(t))dt\right] \\ \text{subject to } & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \\ & \boldsymbol{x}(0) = \hat{\boldsymbol{x}}(0) \\ & \boldsymbol{u}(0) = \hat{\boldsymbol{u}}(0) \\ & \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) \geq \boldsymbol{0} \end{aligned} . \tag{22}$$

The closed-loop system is illustrated by the block diagram in Fig. 3.

Assuming the stability of the closed-loop system, the following assumption is made:

Assumption 1. *For an asymptotically constant feasible reference $\boldsymbol{r}(t\to\infty) = \boldsymbol{r}_\infty$ and disturbance $\boldsymbol{d}_p(t\to\infty) = \boldsymbol{d}_{p,\infty}$, all the states converge. Therefore, the measured outputs $\boldsymbol{y}_p$ and the input $\boldsymbol{u}$ reach steady-state values, respectively $\boldsymbol{y}_{p,\infty}$ and $\boldsymbol{u}_\infty$.*

In the linear case, the observability and the controllability of the nominal system are required for offset-free control. In the nonlinear case, both conditions are replaced as follows:

Assumption 2. *For all $\boldsymbol{y} = \boldsymbol{r}$, there exists a unique $(\boldsymbol{x}_\infty,\boldsymbol{u}_\infty)$ such that*

$$\begin{cases} \boldsymbol{0} &= \boldsymbol{f}(\boldsymbol{x}_\infty,\boldsymbol{u}_\infty) \\ \boldsymbol{y} &= \boldsymbol{g}(\boldsymbol{x}_\infty,\boldsymbol{u}_\infty) \end{cases} . \tag{23}$$

Using the implicit function theorem, it is possible to define a condition on the linearized model to test Assumption 2 locally in the neighborhood of the steady state. Let us introduce the following matrices:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}_\infty,\boldsymbol{u}_\infty) &= \boldsymbol{A} \\ \frac{\partial}{\partial \boldsymbol{u}}\boldsymbol{f}(\boldsymbol{x}_\infty,\boldsymbol{u}_\infty) &= \boldsymbol{B} \\ \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{g}(\boldsymbol{x}_\infty,\boldsymbol{u}_\infty) &= \boldsymbol{C} \\ \frac{\partial}{\partial \boldsymbol{u}}\boldsymbol{g}(\boldsymbol{x}_\infty,\boldsymbol{u}_\infty) &= \boldsymbol{D}. \end{aligned}$$

Proposition 1. *Assumption 2 holds locally if $n_y = n_u$ and*

$$\mathit{rank}\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} = n_x + n_u. \tag{24}$$

Proof. Let us introduce the function

$$\boldsymbol{\mathcal{I}}(\boldsymbol{y},\boldsymbol{z}) = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{z}_1,\boldsymbol{z}_2) \\ \boldsymbol{g}(\boldsymbol{z}_1,\boldsymbol{z}_2) - \boldsymbol{y} \end{bmatrix} \tag{25}$$

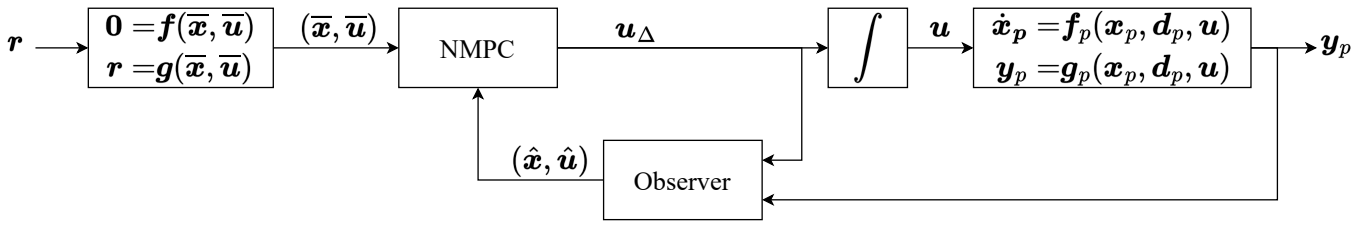


Fig. 3. Delta-input NMPC scheme. The NMPC and the observer respectively have the form of (22) and (21).

where $\boldsymbol{z} = [\boldsymbol{z}_1^T \ \boldsymbol{z}_2^T]^T$, $\boldsymbol{z}_1 = \boldsymbol{x}$ and $\boldsymbol{z}_2 = \boldsymbol{u}$, such that $\boldsymbol{\mathcal{I}}(\boldsymbol{y}_\infty, \boldsymbol{z}_\infty) = \boldsymbol{0}$ (23) and

$$\frac{\partial}{\partial \boldsymbol{z}} \boldsymbol{\mathcal{I}}(\boldsymbol{y}_\infty, \boldsymbol{z}_\infty) = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}. \tag{26}$$

According to the implicit function theorem, if the Jacobian $\frac{\partial}{\partial \boldsymbol{z}} \boldsymbol{\mathcal{I}}(\boldsymbol{y}_\infty, \boldsymbol{z}_\infty)$ is invertible, then there exists a unique solution $\boldsymbol{z}_\infty = [\boldsymbol{x}_\infty^T \ \boldsymbol{u}_\infty^T]^T$ to (23). Assuming $n_y = n_u$, the Jacobian is a square matrix of dimension $n_x + n_u$. As the Jacobian is invertible if and only if the matrix rank is $n_x + n_u$, it ends the proof of Proposition 1. □

In the case of a linear and stable observer, the function $\boldsymbol{l}_u$ (21) becomes a full row rank $n_u \times n_y$ matrix $\boldsymbol{L}_u$ [21], with $\dot{\hat{\boldsymbol{u}}} = \boldsymbol{u}_\Delta + \boldsymbol{L}_u(\boldsymbol{y}_p - \boldsymbol{g}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}))$. If $n_u = n_y$, then $\boldsymbol{L}_u(\boldsymbol{y}_p - \boldsymbol{g}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})) = 0$ at steady state implies $\boldsymbol{g}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) = \boldsymbol{y}_p$, since $\boldsymbol{L}_u$ is square and non-singular. Thus, the observer is said to be nominally error free (since its output matches the plant output at steady state). However, this may not be the case with a nonlinear observer, hence the following assumption:

Assumption 3. *At steady state, the observer* (21) *is nominally error-free:*

$$\boldsymbol{l}_u(\boldsymbol{y}_{p,\infty} - \boldsymbol{g}(\hat{\boldsymbol{x}}_\infty, \hat{\boldsymbol{u}}_\infty)) = \boldsymbol{0} \implies \boldsymbol{y}_{p,\infty} - \boldsymbol{g}(\hat{\boldsymbol{x}}_\infty, \hat{\boldsymbol{u}}_\infty) = \boldsymbol{0}. \tag{27}$$

Let the control law be defined by the function $\boldsymbol{c}_0$: $\boldsymbol{u}_{\Delta 0}(t) = \boldsymbol{c}_0(\hat{\boldsymbol{x}}(t), \hat{\boldsymbol{u}}(t), \boldsymbol{r}(t))$, where $\boldsymbol{u}_{\Delta 0}$ is the solution of the NMPC problem (22).

Assumption 4. *The NMPC is designed to be nominally error-free at steady state, i.e., for all feasible reference* $\boldsymbol{r}_\infty$*,*

$$\boldsymbol{f}(\boldsymbol{x}_\infty, \boldsymbol{u}_\infty) = \boldsymbol{0},\ \boldsymbol{c}_0(\boldsymbol{x}_\infty, \boldsymbol{u}_\infty, \boldsymbol{r}_\infty) = \boldsymbol{0} \implies \boldsymbol{g}(\boldsymbol{x}_\infty, \boldsymbol{u}_\infty) = \boldsymbol{r}_\infty \tag{28}$$

holds for all $\boldsymbol{x}_\infty$ *and for all* $\boldsymbol{u}_\infty$*.*

Finally, the theorem giving a sufficient condition for offset-free control is the following:

Theorem 1. *If Assumptions 1-4 hold and* $n_u = n_y$*, then* $\boldsymbol{y}_p(t \to \infty) = \boldsymbol{r}(t \to \infty)$*.*

Proof. Assumption 1 enforces asymptotic closed-loop stability. Therefore, only $\boldsymbol{y}_{p,\infty} = \boldsymbol{r}_\infty$ needs to be proven.

Combining Assumptions 2 and 3, at steady state, the observer satisfies

$$\boldsymbol{0} = \boldsymbol{f}(\hat{\boldsymbol{x}}_\infty, \hat{\boldsymbol{u}}_\infty) \tag{29}$$

and

$$\boldsymbol{y}_{p,\infty} = \boldsymbol{g}(\hat{\boldsymbol{x}}_\infty, \hat{\boldsymbol{u}}_\infty). \tag{30}$$

From (29) and Assumption 4, it follows that

$$\boldsymbol{g}(\hat{\boldsymbol{x}}_\infty, \hat{\boldsymbol{u}}_\infty) = \boldsymbol{r}_\infty \tag{31}$$

since at steady state $\boldsymbol{c}_0(\hat{\boldsymbol{x}}_\infty, \hat{\boldsymbol{u}}_\infty, \boldsymbol{r}_\infty) = \boldsymbol{0}$.

Finally, combining these last two equations yields

$$\boldsymbol{y}_{\boldsymbol{p},\infty} = \boldsymbol{r}_\infty. \tag{32}$$

Thereby, the control is offset free, i.e., with no steady-state error. □

C. AMES Control

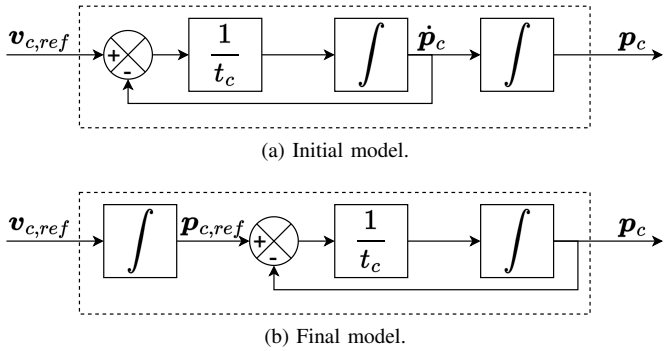
Theorem 1 requires as many independent inputs as independent outputs. The AMES has $n_p + 3$ inputs, corresponding to n_p thrusts for the AM and three Cartesian reference velocity components for the carrier. If the only measured outputs are the AM pose $\boldsymbol{p}$ and the carrier position $\boldsymbol{p}_c$, then Theorem 1 can be tested only if $n_p = 6$. Indeed, let us assume that $n_p > 6$. Then, the nullity of matrix $\boldsymbol{W}_a$ is non-zero. As a consequence, the control signal $\boldsymbol{\tau}_{ref}$, which becomes a state of the system in delta-input formulation, is not observable. Indeed, different thrusts can produce the same wrench on the platform (as already mentioned by Maeder et al. [21] at Section 5.3).

This limitation could be circumvented by considering a model of the AM where the total wrench $\boldsymbol{W}_a \boldsymbol{\tau}_{ref} \in \mathbb{R}^6$ is the control signal rather than individual thrusts of the propulsion units [11]. However, such a model implies that the total wrench computed by the NMPC has to be distributed between the propulsion units by solving online a separate control allocation problem. Moreover, this approach would be incompatible with the consideration of saturations at the motor level.

This work investigates another approach that measures and constrains the projection of the control input $\boldsymbol{\tau}_{ref}$ in the nullspace of matrix $\boldsymbol{W}_a$. This enables the observability of the control input as explained in the remaining of this section.

1) State-Space Model: Theorem 1 requires the model of the AMES to be in the delta-input form (20). The carrier dynamics can be considered in a delta-input formulation without introducing additional integrators. As illustrated in Fig. 4, moving the integrator from the outputs of the carrier model (see Fig. 4a) to the inputs yields a model (see Fig. 4b) with the same input-output behavior. This equivalent model is used as a delta-input formulation of the carrier dynamics, where the control signal is a variation of the anchoring point position, i.e., a velocity reference. Therefore, the nominal carrier model, equivalent to (7), is

$$\dot{\boldsymbol{p}}_c = \frac{1}{t_c}(\boldsymbol{p}_{c,ref} - \boldsymbol{p}_c). \tag{33}$$



(a) Initial model.

(b) Final model.

Fig. 4. Equivalent delta-input formulation of the carrier dynamics.

Let us define the input $\boldsymbol{u} = [\boldsymbol{\tau}_{ref}\ \boldsymbol{p}_{c,ref}]^T$ of the nominal model and the measured variables $\boldsymbol{y} = [\boldsymbol{p}^T\ \boldsymbol{\nu}^T\ \boldsymbol{p}_c^T]^T$. Variable $\boldsymbol{\nu} \in \mathbb{R}^{n_p-6}$ is a projection of the control input $\boldsymbol{\tau}_{ref}$ within the nullspace $\mathrm{Null}(\boldsymbol{W}_a)$ of matrix $\boldsymbol{W}_a$. This can be written as $\boldsymbol{\nu} = \boldsymbol{N}_w\boldsymbol{\tau}_{ref}$ with $\boldsymbol{N}_w^T$ a matrix whose columns form a basis of $\mathrm{Null}(\boldsymbol{W}_a)$.

The nominal plant model of the AMES is then obtained by combining (4), (6) and (33):

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{f}_{nom}(\boldsymbol{x},\boldsymbol{u}) \\ \boldsymbol{y} &= \boldsymbol{g}_{nom}(\boldsymbol{x},\boldsymbol{u}) \end{cases} \tag{34}$$

with $\boldsymbol{x} = [\dot{\boldsymbol{p}}^T\ \boldsymbol{p}^T\ \boldsymbol{\tau}^T\ \boldsymbol{p}_c^T]^T$ the state variable.

The nominal plant model (34) is augmented with integrators before the control inputs $\boldsymbol{u}$ such that the plant model is transformed into the delta-input form with input $\boldsymbol{u}_\Delta = [\dot{\boldsymbol{\tau}}_{ref}^T\ \boldsymbol{v}_{c,ref}^T]^T$

$$\begin{cases} \dot{\boldsymbol{x}} &= \boldsymbol{f}_{nom}(\boldsymbol{x},\boldsymbol{u}) \\ \dot{\boldsymbol{u}} &= \boldsymbol{u}_\Delta \\ \boldsymbol{y} &= \boldsymbol{g}_{nom}(\boldsymbol{x},\boldsymbol{u}) \end{cases}. \tag{35}$$

At the expense of an increased dimension of the state vector, this formulation enforces an optimal control input $\boldsymbol{u}_\Delta = \boldsymbol{0}$ for any desired feasible steady state. Furthermore, this formulation allows for penalizing the derivative of the AM nominal control input $\boldsymbol{\tau}_{ref}$ in the cost function, which results in a smoother control signal and consequently a smoother variation of the current in the brushless DC motors.

2) Cost Function: A quadratic cost function is preferred since efficient solvers exist for quadratic problems. Then, according to (22), the cost function J is of the form

$$\begin{aligned} J = \int_0^T \big(&\|\boldsymbol{p} - \boldsymbol{p}_{ref}\|^2_{\boldsymbol{Q_a}} + \|\boldsymbol{p}_c - \boldsymbol{p}_{c,ref}\|^2_{\boldsymbol{Q_c}} \\ &+ \|\boldsymbol{\nu} - \boldsymbol{\nu}_{ref}\|^2_{\boldsymbol{Q_\nu}} + \|\dot{\boldsymbol{\tau}}_{ref}\|^2_{\boldsymbol{R_a}} + \|\boldsymbol{v}_{c,ref}\|^2_{\boldsymbol{R_c}} \big)\, dt \end{aligned} \tag{36}$$

with $\|\boldsymbol{v}\|^2_{\boldsymbol{P}} = \boldsymbol{v}^T\boldsymbol{P}\boldsymbol{v}$.

The cost function (36) is minimal when the measured variables $\boldsymbol{y}$ reach their desired steady state. Furthermore, the last two terms in this cost function tend to reduce the energy consumption and limit the control signal chattering.

The proposed framework for the control requires two control allocation strategies, (i) one to obtain the reference trajectory of the carrier $\boldsymbol{p}_{c,ref}$ knowing the reference trajectory of the AM $\boldsymbol{p}_{ref}$ and (ii) one to choose the desired value $\boldsymbol{\nu}_{ref}$ of the projection of the reference AM thrusts $\boldsymbol{\tau}_{ref}$ in the nullspace of matrix $\boldsymbol{W}_a$. This limitation is the same as with the partitioned controller from Section III and the allocation can be carried out the same way, i.e., through (11) and (13).

3) Actuator Saturations: They are handled by adding a constraint on the thrust reference $\boldsymbol{\tau}_{ref}$ in the OCP (22): $|\tau_i| \leq \tau_M$, with τ_M the maximum thrust propulsion units can generate.

4) Theoretical Validation of the Offset-Free Steady State: With the whole AMES (carrier and AM) model in the delta-input formulation, offset-free property of the NMPC can be inferred from Theorem 1.

Assuming that only the pose $\boldsymbol{p}$ of the AM ($\dim(\boldsymbol{p}) = 6$), the position $\boldsymbol{p}_c$ of the carrier ($\dim(\boldsymbol{p}_c) = 3$) and the projection $\boldsymbol{\nu}$ of the AM control inputs into $\mathrm{Null}(\boldsymbol{W}_a)$ ($\dim(\boldsymbol{\nu}) = n_p - 6$) are measured, the number of measurements is $n_y = n_p + 3$. From the previous modeling, the AMES model (35) has $n_u = \dim(\boldsymbol{\tau}_{ref}) + \dim(\boldsymbol{p}_{c,ref}) = n_p + 3$ delta inputs. Thereby, $n_u = n_y$ and the controller is offset free at steady state if Assumptions 1-4 can be verified.

Proposition 1 is formulated as a condition on the nominal model linearized at the desired steady state. Let us linearize the AM dynamics around the steady state $(\boldsymbol{p}_\infty, \boldsymbol{p}_{c,\infty}, \boldsymbol{\tau}_\infty)$:

$$\begin{aligned} \boldsymbol{M}_a(\boldsymbol{p}_\infty)\delta\ddot{\boldsymbol{p}}_\infty + \boldsymbol{G}_p(\boldsymbol{p}_\infty, \boldsymbol{p}_{c,\infty})\delta\boldsymbol{p} \\ + \boldsymbol{G}_c(\boldsymbol{p}_\infty, \boldsymbol{p}_{c,\infty})\delta\boldsymbol{p}_c = \boldsymbol{W}_a(\boldsymbol{p}_\infty)\delta\boldsymbol{\tau} \end{aligned} \tag{37}$$

with $\delta\boldsymbol{p} = \boldsymbol{p} - \boldsymbol{p}_\infty$, $\delta\boldsymbol{p}_c = \boldsymbol{p}_c - \boldsymbol{p}_{c,\infty}$ and $\delta\boldsymbol{\tau} = \boldsymbol{\tau} - \boldsymbol{\tau}_\infty$. The nominal AMES model linearized at the steady state $\boldsymbol{x_\infty}$ is then obtained by combining (33) and (37).

To test Proposition 1 (in order to verify Assumption 2), the following matrices of the AMES state-space model are introduced based on the linearized model:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{M}_a^{-1}\boldsymbol{G}_p & \boldsymbol{M}_a^{-1}\boldsymbol{W}_a & -\boldsymbol{M}_a^{-1}\boldsymbol{G}_c \\ \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\frac{1}{t_\tau}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\frac{1}{t_c}\boldsymbol{I} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \frac{1}{t_\tau}\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{t_c}\boldsymbol{I} \end{bmatrix} \boldsymbol{C} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{D} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{N}_w & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

Let us evaluate the Jacobian (26)

$$\boldsymbol{J}_\infty = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}. \tag{38}$$

Using row and column operations, the determinant of the Jacobian verifies

$$|\det(\boldsymbol{J}_\infty)| = \frac{1}{t_c^3}\frac{1}{t_\tau^{n_p}} \left| \det\left(\begin{bmatrix} \boldsymbol{M}_a^{-1}\boldsymbol{W}_a \\ \boldsymbol{N}_w \end{bmatrix} \right) \right| \tag{39}$$

Therefore, $\boldsymbol{J}_\infty$ is invertible since the row space of $\boldsymbol{W}_a$ is orthogonal to the nullspace of $\boldsymbol{W}_a$, and columns of $\boldsymbol{N}_w^T$ form a basis of this nullspace.

To reconstruct the full state of the augmented system, an extended Kalman filter (EKF) is used as an observer. Based on a linearization of the smooth nonlinear dynamics of the system (35), an estimate of the state is provided. As for a basic

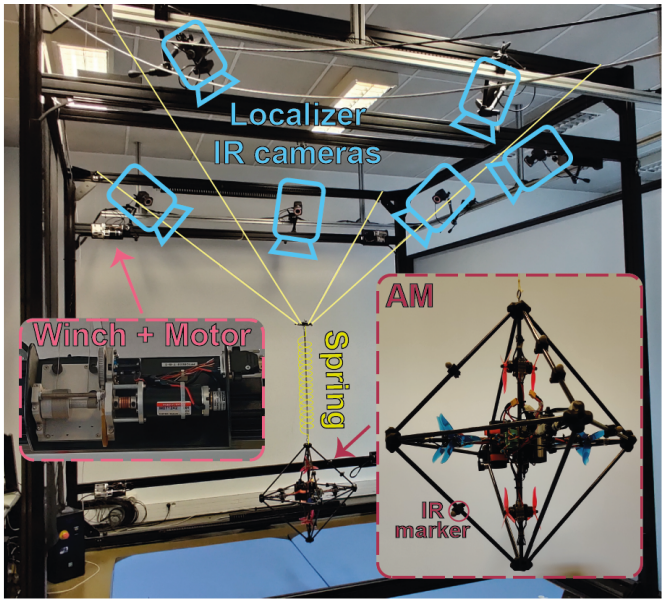


Fig. 5. Experimental setup.

Kalman filter, the correction term on the state estimate is the product of the gain matrix $\boldsymbol{l}$ and the error between estimated and measured outputs. As the augmented system model uses as many measured variables as inputs, the submatrix $\boldsymbol{l}_u$ of $\boldsymbol{l}$, associated with the state $\boldsymbol{u}$ (see (21)), is square. Thereby, as explained previously (before Assumption 3) for the linear case, the output estimate is error-free at steady state. Thus, Assumption 3 holds.

In conclusion, offset-free control can be reached under the Assumptions 1 and 4.

V. Experimental Setup

A. Hardware Description

We designed and implemented the robot shown in Fig. 5, where an omnidirectional aerial manipulator is suspended from the platform of a four-cable CDPR by a spring.

The AM was described in detail in a previous paper [27]. It has the same structure as an omnidirectional multirotor vehicle described by Brescianini and D'Andrea (Fig. 3a in [8]) and can generate a 6-DoF wrench thanks to six bidirectional propulsion units (or thrusters). The positioning of propulsion units is the result of an optimization problem that maximizes the agility, i.e., the highest thrust and torque that can be generated in all directions [8]. With this configuration, the wrench matrix ${}^b\boldsymbol{W}$ mapping the thrusts $\boldsymbol{\tau}$ to the total wrench generated by the propellers is

$$ {}^b\boldsymbol{W} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ b & -b & 0 & 0 & d_{B_iG} & -d_{B_iG} \\ d_{B_iG} & -d_{B_iG} & b & -b & 0 & 0 \\ 0 & 0 & d_{B_iG} & -d_{B_iG} & b & -b \end{bmatrix}. \tag{40} $$

The determinant of the wrench matrix verifies $|\det({}^b\boldsymbol{W})| = 8d_{B_iG}^3 + 8b^3$. As a consequence, since the propulsion units are bidirectional, the AM can generate any wrench (within saturation limits). This is in agreement with the definition of the omnidirectional AM proposed by Hamandi et al. [7].

Most commercial electronic speed controllers (ESCs) for bidirectional propellers are sensorless, using only the back EMF due to the rotor motion to prevent the slip. The back EMF being hard to measure at low speeds, the motor control input suffers from a dead-zone when changing the direction of rotation, increasing significantly its response time. This issue is documented as "ESC-induced singularity" by Park et al. [9]. Non-suspended aerial vehicles often require a constant thrust on all motors to hover. Hence, they usually do not require to change the thrust direction. For the AMES, motor thrusts are around zero at steady state. This would yield very frequent changes in the direction of rotation if bidirectional propellers are used. To avoid this problem, it was decided to use the double unidirectional propeller configuration. Thereby, each propulsion unit is made of a pair of propellers (DALPROP 5045), mounted on two coaxial brushless DC motors (T-Motor F-40 Pro III Kv2400). According to the sign and magnitude of the force to be generated, the corresponding propeller rotates at the desired speed while the second one is idling, i.e., rotating at its lowest speed ($1500\,\mathrm{rpm}$, equivalent to $0.03\,\mathrm{N}$). To control the rotational speed of the propellers, we developed the open-source firmware called Teensyshot (https://github.com/jacqu/teensyshot) that implements a fast PID (proportional-integral-derivative) speed regulation loop on a Teensy 4.0 embedded system using real-time telemetry data acquired from KISS 32A ESCs.

The AM carries its own energy source, a $2300\,\mathrm{mAh}$, $11.1\,\mathrm{V}$ / 3S lithium polymer battery pack (TATTU 3S1P). It also has an onboard CPU (Raspberry Pi 4B) running high-level control algorithms and communicating with other remote devices (e.g., CDPR and motion capture system) through Wi-Fi TCP/IP sockets thanks to the open-source Simulink toolbox RPIt developed in our lab [38]. While the partitioned controller runs on the onboard CPU, the NMPC runs on a remote computer (Intel i5-9500) to increase the prediction horizon and the sampling rate. Existing similar works show that NMPC could also be solved on board with more powerful embedded computers [39], [40].

A CDPR is selected as the carrier. Like AMs, CDPRs are particularly lightweight with respect to the size of their workspace and reduce raw material consumption. Indeed, some commercially available CDPRs like the Skycam or the Spidercam [https://www.spidercam.tv] may cover large areas of up to 250m x 250m using only four winches as actuators. Rotational DoFs might also be controlled by cables like with the eight-cable suspended CDPR CoGiRo [41], or by a more conventional robot mounted on the CDPR platform [42]. Suspended CDPRs also consume some energy to carry the platform and fully-constrained CDPRs might even consume more energy to maintain some desired tension in the cables (e.g., to increase the stiffness of the platform). But for a given static payload, AMs consume far more energy to compensate for gravity than CDPRs. Indeed, on a CDPR, when the platform is static, no mechanical energy is consumed, only electrical energy, mainly in the motor windings due to resistive

losses. On the other hand, propellers of an AM rotate at high speed to generate thrust, which requires high mechanical power and even higher electrical power.

The spring between the AM and the CDPR compensates for gravity. Thereby, the AM floats almost freely around its equilibrium position, and is therefore very energy efficient. For large displacements, the carrier is used to slowly move the equilibrium point of the AM to the average position of the current task in order to further reduce the energy consumption. The AMES combines the large workspace capabilities of the carrier with the high dynamics of AMs while avoiding their flight-time limitations. Since the rotations are exclusively handled by the AM, only a limited number of cables are required (a minimum of three).

The CDPR carrier is a commercial product called Inca™ provided by Haption, [https://www.haption.com/en/products-en/inca-en.html]. The CDPR platform is small enough to assume that the cables are attached to a unique point, which is also the spring anchoring point. Each cable is equipped with a balancing spring at the winch side to maintain tension in the cable and also to create a passive equilibrium point, reducing the energy consumption even more. The CDPR is actuated by four Maxon RE40 (48V) DC motors with encoders. Each motor drives two winches with helical grooves, one for the balancing spring side and the other one for the end-effector side. A digital PI (proportional-integral) controller running on an FPGA regulates the current of each motor at $25\,\mathrm{kHz}$.

An industrial PC/104 rack drives the CDPR motors. It is equipped with a Pentium M $1.6\,\mathrm{GHz}$ CPU, power supplies and FPGAs. A Linux operating system with a Xenomai real-time core extension runs on the CPU. A TCP/IP socket server sends the current reference periodically to the FPGAs, at $500\,\mathrm{Hz}$. The current reference is updated asynchronously at a slower refresh rate ($400\,\mathrm{Hz}$) by the onboard CPU of the AM. At each reference update request, the TCP/IP client receives back (i) the measured currents, (ii) the position and velocity of the winches from the incremental encoder and (iii) the $\mathrm{I^2t}$ integral estimations to prevent overheating of the motors. If the socket server does not receive a request for $500\,\mathrm{ms}$, a software watchdog disables the motors for security purpose.

The higher-level controller runs on the onboard CPU of the AM. It handles the digital regulation of the angular velocity of cable winches and the position control of the anchoring point. Measurements are available for the controllers from two types of sensors: winch motor encoders on the CDPR and 3D measurements from a Vicon Bonita motion-capture system ($240\,\mathrm{Hz}$, infrared markers on the anchoring point and on the AM).

Numerical values of model parameters used for experiments are listed in Table I. The drag is neglected, hence $b = 0$.

TABLE I
MODEL PARAMETERS

Parameter	Value	Unit
m	1.95	kg
${}^b\boldsymbol{I}_a$	$\mathrm{diag}(35.3, 34.4, 34.0)$	$10^{-3}\mathrm{kg\,m^2}$
k	51	$\mathrm{N/m}$
l_0	0.35	m
d_{B_iG}	0.17	m
d_{AG}	0.345	m
a	1.36×10^{-6}	$\mathrm{N/(rad/s)^2}$
τ_M	7.2	N
n_p	6	
k_c	49	$\mathrm{N/m}$
$\boldsymbol{I_c}$	$29.1 \times 10^{-6}\boldsymbol{I}$	$\mathrm{kg\,m^2}$
r_1	6×10^{-3}	m
r_2	17×10^{-3}	m
n_c	4	

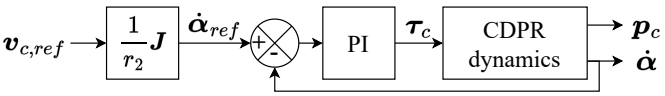


Fig. 6. CDPR velocity controller.

With these assumptions, the dynamic model of the CDPR in its canonical form becomes [43], [44]

$$\boldsymbol{M}_c(\boldsymbol{p}_c)\ddot{\boldsymbol{p}}_c + \boldsymbol{C}_c(\boldsymbol{p}_c, \dot{\boldsymbol{p}}_c)\dot{\boldsymbol{p}}_c + \boldsymbol{g}_c(\boldsymbol{p}_c, \boldsymbol{p}) = -\boldsymbol{J}^T\boldsymbol{\tau}_c \quad (41)$$

with

$$\begin{cases} \boldsymbol{M}_c(\boldsymbol{p}_c) &= \frac{1}{r_2}\boldsymbol{J}^T\boldsymbol{I}_c\boldsymbol{J} \\ \boldsymbol{C}_c(\boldsymbol{p}_c, \dot{\boldsymbol{p}}_c) &= \frac{1}{r_2}\boldsymbol{J}^T\boldsymbol{I}_c\dot{\boldsymbol{J}} \\ \boldsymbol{g}_c(\boldsymbol{p}_c, \boldsymbol{p}) &= k_c\frac{r_1^2}{r_2}\boldsymbol{J}^T(\boldsymbol{l}_c - \boldsymbol{l}_{c0}) - {r_2}^f\boldsymbol{f}_s \end{cases}$$

where k_c is the balancing-spring stiffness, $\boldsymbol{I}_c \in \mathbb{R}^{n_c \times n_c}$ the diagonal matrix of actuator inertiae, $\boldsymbol{\tau}_c \in \mathbb{R}^{n_c}$ the motor torques, $\boldsymbol{l}_{c0} \in \mathbb{R}^{n_c}$ the cable lengths at equilibrium (when $\boldsymbol{\tau} = \boldsymbol{0}$), $\boldsymbol{J} \in \mathbb{R}^{n_c \times 3}$ the Jacobian matrix of the CDPR relating the anchoring point velocity $\dot{\boldsymbol{p}}_c$ to $\dot{\boldsymbol{l}}_c$, that is the time derivative of the cable lengths $\boldsymbol{l}_c \in \mathbb{R}^{n_c}$.

To regulate the velocity $\dot{\boldsymbol{\alpha}}$ of the winches, a PI velocity controller is implemented (see Fig. 6). This rejects significant nonlinearities like dry friction. Furthermore, assuming that the spring elasticity is high enough, its restoring force can be considered as a constant disturbance for this velocity loop. The controller runs at $400\,\mathrm{Hz}$. The coefficients of the controller are $k_{p,\dot{\alpha}} = 2.55 \times 10^{-2}$ and $k_{i,\dot{\alpha}} = 1.53$.

With the velocity controller, winches are assumed to have first-order dynamics with time constant $t_c = 0.5\,\mathrm{s}$

$$\ddot{\boldsymbol{\alpha}} = \frac{1}{t_c}(\dot{\boldsymbol{\alpha}}_{ref} - \dot{\boldsymbol{\alpha}}). \quad (42)$$

Since the motion of the carrier is supposed to be slow within its large workspace, the Jacobian matrix $\boldsymbol{J}(\boldsymbol{p}_c)$ (41) evolves slowly. So, the time derivative of the Jacobian matrix can be neglected, yielding

$$\boldsymbol{J}\ddot{\boldsymbol{p}}_c = r_2\ddot{\boldsymbol{\alpha}}. \quad (43)$$

B. CDPR Velocity Controller

The CDPR platform, at the intersection of the n_c cables, is assumed to be a massless point particle. Indeed, since only the translational motion is required for the CDPR, the platform might be very small, almost punctual. The cables are considered as straight lines and their elasticity is neglected compared to the elasticity of the balancing springs.

Combining (43) and (42) with the first-order CDPR kinematic model

$$\boldsymbol{J}\dot{\boldsymbol{p}}_c = r_2\dot{\boldsymbol{\alpha}} \tag{44}$$

results in a first-order system with the same time constant t_c that models the dynamics of the anchoring point C Cartesian motion

$$\ddot{\boldsymbol{p}}_c = \frac{1}{t_c}(\boldsymbol{v}_{c,ref} - \dot{\boldsymbol{p}}_c). \tag{45}$$

C. Partitioned Controller Implementation and Tuning

Both controllers, for the carrier and for the AM, run on the onboard computer at the same frequency as the NMPC, which is $100\,\mathrm{Hz}$.

1) Carrier Position Controller: The loop that tracks the position reference $\boldsymbol{p}_{c,ref}$ of the carrier implements the proportional controller with the feedforward term (15), where $\boldsymbol{K}_{ff,c} = 0.5\boldsymbol{I}$ and $\boldsymbol{k}_{p,c} = 1.3\boldsymbol{I}$.

To guarantee positive tension of the cables, a tension distribution algorithm can be included if required [35]. However, with the present suspended configuration and the balancing springs that guarantee a minimal tension, all the cable tensions remained positive during the experiments.

2) AM Controller: The PID gain matrices of the computed torque controller are diagonal, therefore the control of all DoFs is decoupled:

$$\begin{aligned}\boldsymbol{K}_p &= \mathrm{diag}(150, 150, 150, 150, 150, 120)\\ \boldsymbol{K}_d &= \mathrm{diag}(20, 20, 20, 20, 20, 20)\\ \boldsymbol{K}_i &= \mathrm{diag}(500, 500, 500, 500, 500, 500).\end{aligned}$$

The controller is tuned experimentally. First, proportional and derivative coefficients are selected to minimize the 5% settling time with no overshoots. Then, the integral coefficient is increased to eliminate the steady-state error while keeping the 5% settling time as small as possible. As it can be seen in Section VI, adding the integral term induces overshoots.

D. NMPC Implementation and Tuning

The estimate of the augmented system state is required by the NMPC. This estimate is provided by an EKF observer.

The measurement of the AM velocity $\dot{\boldsymbol{p}}$ is considered in the observer along with the measurement of the AM pose $\boldsymbol{p}$ and the carrier position $\boldsymbol{p}_c$ in order to improve the state estimate. In this context, the measured variables $\boldsymbol{y}$ have a dependency at steady state: the velocity measurement is necessarily zero if the pose measurement is constant. In this situation, the condition $n_u = n_y$ can be relaxed, as highlighted by Morari and Maeder [22]. Indeed, if the measured variables $\boldsymbol{y}$ are degenerate, i.e., there is a linear dependency between measurements, Assumption 2 can be readily modified to eliminate redundant equations. The same reasoning can also be applied to Assumption 3. As a consequence, Theorem 1 is still valid.

The NMPC runs at $100\,\mathrm{Hz}$. In order to have a $1\,\mathrm{s}$ prediction horizon, the discretization sampling period must be chosen such that the OCP (22) stays solvable in strictly less than $10\,\mathrm{ms}$. Indeed, increasing the sampling period for the discretization of the prediction horizon reduces the computational complexity, which is proportional to the number of discretization steps. Here, the prediction model is discretized with a $0.02\,\mathrm{s}$ sampling period, reducing the computational burden as proposed by Bicego et al. [24].

Let us define the weighting coefficients q_{p_g}, $q_{\eta_{rp}}$, $q_{\eta_{yaw}}$, q_{p_c}, $q_{\dot{\tau}_{ref}}$ and $q_{v_{c,ref}}$. The weighting matrices of the cost function (36) are selected diagonal:

$$\begin{aligned}\boldsymbol{Q}_a &= \mathrm{diag}(q_{p_g}, q_{p_g}, q_{p_g}, q_{\eta_{rp}}, q_{\eta_{rp}}, q_{\eta_{yaw}})\\ \boldsymbol{Q}_c &= q_{p_c}\boldsymbol{I}\\ \boldsymbol{R}_a &= q_{\dot{\tau}_{ref}}\boldsymbol{I}\\ \boldsymbol{R}_c &= q_{v_{c,ref}}\boldsymbol{I}.\end{aligned}$$

An initial approximate value for the weights is obtained by normalization, considering the maximum expected value of each variable in the cost function. Then, the weighting coefficients are tuned experimentally to minimize the 5% settling time while limiting the amplitude of oscillations at steady state. The weighting coefficients are shown in Table II.

TABLE II
NMPC PARAMETERS

Parameter	Value
q_{p_g}	25.00
$q_{\eta_{rp}}$	1.64
$q_{\eta_{yaw}}$	2.46
q_{p_c}	12.50
$q_{\dot{\tau}_{ref}}$	$4.86 \cdot 10^{-6}$
$q_{v_{c,ref}}$	1.80

The NMPC is implemented using acados [31], an open-source framework that generates efficient low-level code for optimization-based control. The continuous OCP (22) is discretized with a multiple shooting method and then solved with the HPIPM solver [45]. For real-time compliance, acados performs the Real-Time Iteration scheme that enables a reasonably fast computation of the NMPC output, but at the cost of a suboptimal solution [46].

The distant computer running the solver hosts a TCP/IP server. The server receives the input data of the controller (setpoint and estimated state) from the embedded computer, and sends back the control input $\boldsymbol{u}_\Delta$. More than 99% of the time, the NMPC solver takes at most $4.7\,\mathrm{ms}$. Similarly, the round-trip time (ping) between the computer solving the NMPC problem and the onboard CPU is at most $1.82\,\mathrm{ms}$. No packet losses are observed.

VI. RESULTS

This section presents experimental results obtained with both controllers, the partitioned controller and the NMPC.

Fifth-order polynomial trajectories are used during the experiments, with zero velocity and zero acceleration at each waypoint. The list of waypoints is shown in Table III, where l_p is a positive length. The time t_p between two consecutive waypoints is the same for each waypoint during an experiment.

TABLE III
POSE AND TIMING OF THE TRAJECTORY WAYPOINTS

	x	y	z	ψ [°]	θ [°]	ϕ [°]	t [s]
0	0	0	0	0	0	0	0
1	$2l_p$	$2l_p$	l_p	5	0	0	t_p
2	$-2l_p$	$2l_p$	l_p	5	0	90	$2\,t_p$
3	$-2l_p$	$-2l_p$	$-l_p$	-5	0	0	$3\,t_p$
4	$2l_p$	$-2l_p$	$-l_p$	5	0	90	$4\,t_p$

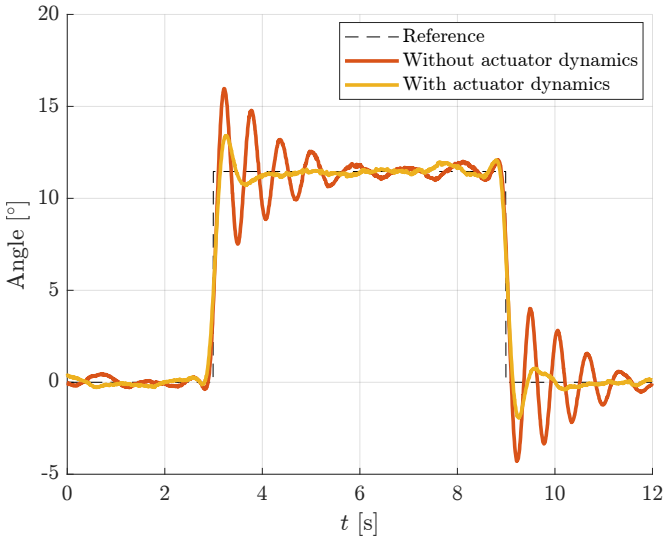


Fig. 7. Step response for yaw control with and without actuator dynamics.

Its numerical value is given for each experiment (see Table VII).

The analysis of step responses use the percentage overshoot, the rise time from 10% to 90% and the settling time at 5%. They are abbreviated respectively as PO, RT and ST.

A. Impact of Actuator Dynamics in the NMPC Prediction Model

An accurate model of plant dynamics is required to achieve high dynamic performances with NMPC. However, improving the accuracy of the model requires additional state variables, increasing the computational burden. The question then arises of whether to take into account the model of the actuators (6).

To illustrate the impact of actuator dynamics in the prediction model of the NMPC, a simple experiment is done during which only the yaw angle is controlled. This degree of freedom is identified both experimentally and in simulation as the most sensitive one to actuator dynamics, since it is not subject to any force or torque other than that of the actuators. Indeed, the suspension spring has no impact on the yaw.

Figure 7 compares step responses with and without actuator dynamics in the prediction model, using the same weights in the cost function. In the latter case, the prediction model assumes that the desired thrust is the reference value.

Table IV sums up overshoots, rise times and settling times in both cases. Removing actuator dynamics from the model doubles the overshoot and triples the settling time, but reduces slightly the rise time.

TABLE IV
STEP RESPONSES FOR YAW CONTROL WITH AND WITHOUT ACTUATOR DYNAMICS

	PO [%]	RT [s]	ST [s]
Without	39.3	0.15	2.48
With	17.0 (-57%)	0.22 (+47%)	0.77 (-69%)

So, considering actuator dynamics improves significantly the transients. Therefore, this makes possible to tune the cost function in a more aggressive way.

B. Step Responses

Step responses are used to assess the dynamic performance of the AMES using both controllers, namely the partitioned controller and the NMPC. Figure 8 shows step responses for each DoF of the AM. The step amplitude is $0.05\,\mathrm{m}$ for translations and 4° for orientations. Overshoots, rise times and 5% settling times are summed up in Table V.

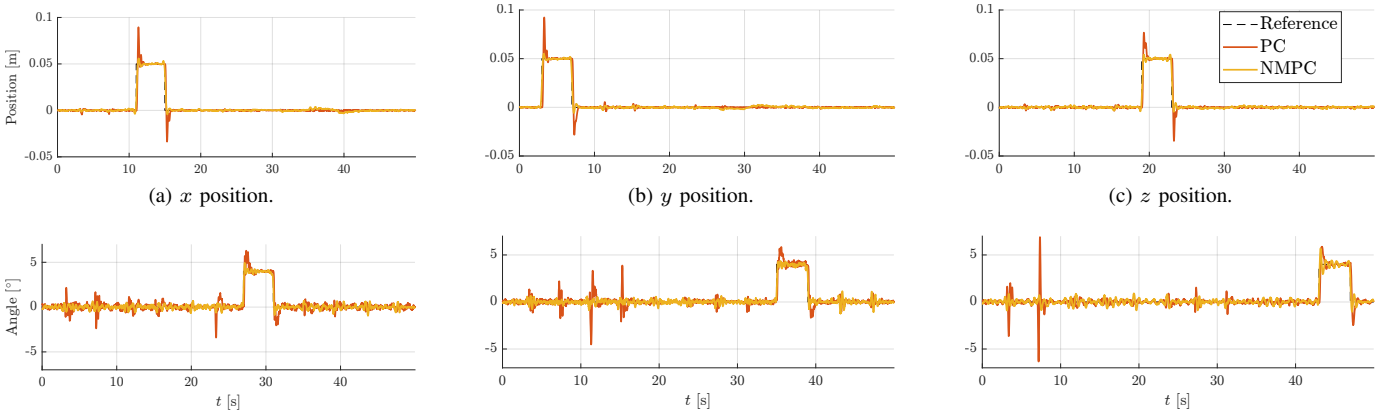
With the selected tuning for both controllers, the settling times are comparable. The superiority of partitioned controller or NMPC depends on the degrees of freedom that are considered. The settling time is shorter with NMPC for x, y, z and pitch, and longer for roll and yaw. However, the rise time is shorter with partitioned controller for all DoFs. The partitioned controller does not handle saturations. Thus, the control signal may be limited during sharp transients, yielding integral term windup. This windup usually yields higher overshoots. Since the control input signal merges the output of the PID with the feedback linearization terms, simple anti-windup strategies cannot be used.

Translational DoFs have shorter settling times with both controllers compared to orientational DoFs.

TABLE V
STEP RESPONSES WITH PARTITIONED CONTROLLER AND NMPC

	PO [%]		RT [s]		ST [s]	
	PC	NMPC	PC	NMPC	PC	NMPC
x	79.0	11.7	0.10	0.22	0.76	0.73
y	84.6	9.6	0.09	0.24	0.85	0.34
z	53.4	7.6	0.09	0.21	0.83	0.60
ψ	57.4	22.6	0.06	0.33	1.68	1.73
θ	46.0	8.5	0.08	0.32	1.99	1.19
ϕ	46.9	42.5	0.09	0.13	1.85	1.98

For a better analysis, the step along x is presented with further details in Fig. 9. During the step, the reference for the relative displacement of the AM CoM x_g and the carrier x_c is the same. Nonlinear model predictive controller anticipates the step, resulting in a smoother movement compared to partitioned controller. As a consequence, thrusts stay low,



(a) x position. (b) y position. (c) z position.

(d) Roll angle. (e) Pitch angle. (f) Yaw angle.

Fig. 8. Step responses for each DoF with partitioned controller and NMPC.

while partitioned controller yields saturation and overshoots. Furthermore, large thrust peaks during transients yield coupling between DoFs due to unmodeled phenomena (such as drag and reaction torque). This explains rotation error spikes in Fig. 8 with partitioned controller. It can also be observed in Fig. 9e and Fig. 9f that the anticipation reduces the total thrust and moment at the CoM G. Moreover, the penalty on the variation of the thrust reference reduces the chattering on the NMPC control signal.

It is important to note that the step responses highly depend on the tuning of the controllers. As a consequence, the selected performance characteristics, namely overshoot, rise time and settling time, do not allow alone for concluding about the superiority of one controller with respect to the other one. However, the analysis of the step responses highlights two advantages of NMPC over partitioned control, regardless of the tuning of the controllers. First, the penalty on the variation of the control input in the cost function reduces chattering and yields smoother signals. Second, the anticipation capabilities of the NMPC when the reference trajectories are known in advance allow for reducing the mean tracking error.

Although the saturation of the thrusters with the actual tuning of the NMPC are not reached, they are handled. Saturation handling is a well-known benefit of model predictive control and is well documented in the literature [24].

C. Offset-free NMPC

1) Rejection of Step Disturbance: A constant disturbance is emulated using the actuators. A constant disturbance signal $\boldsymbol{\tau}_d$ is added to the control signal $\boldsymbol{\tau}_{ref}$ sent to the motors at time $t = 0\,\mathrm{s}$ (see Fig. 10). The disturbance is $3\,\mathrm{N}$ for translations with a duration of $10\,\mathrm{s}$ and $1\,\mathrm{N}$ with a $0.3\,\mathrm{m}$ lever arm, so $0.3\,\mathrm{Nm}$, for orientations with a duration of $5\,\mathrm{s}$.

With the observer, constant disturbances are rejected for all DoFs. For translations, the rejection takes $5\,\mathrm{s}$, for roll and pitch $0.7\,\mathrm{s}$, and, for yaw, $1.2\,\mathrm{s}$. Without the observer, there is a constant static error in the presence of the disturbance, and even before introducing the disturbance. It shows that the equilibrium position is misestimated by the EKF, mainly due to errors on robot mass and suspension spring stiffness.

2) Rejection of Modeling Errors: To validate the offset-free behavior of NMPC in presence of modeling errors, a $0.15\,\mathrm{kg}$ mass is attached at the bottom of the robot, increasing by 7.7% the mass of the AM and moving its CoM by $2.5\,\mathrm{cm}$. Figure 11 shows the AM pose in three scenarios: (i) without observer and without load, (ii) without observer and with load, (iii) with observer and with load. Despite the load, the observer allows for an offset-free tracking of the trajectory. Root-mean-square errors are presented in Table VI and compared to those for the nominal case, i.e., with observer and without load. Despite the load, the observer designed with the delta-input model of the AMES improves significantly the trajectory tracking. In the absence of an observer, introducing the load has the most significant impact on the position (mostly vertical) rather than the orientations. Indeed, the tilt angle of the robot is low during the trajectory, and so is the moment the load generates.

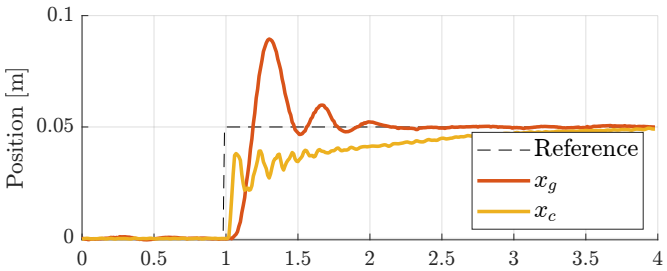
TABLE VI
ROOT-MEAN-SQUARE ERRORS

	$\boldsymbol{p}_g$ [mm]	ψ [°]	θ [°]	ϕ [°]
EKF + no load	2.8	0.14	0.16	0.21
EKF + load	3.4	0.25	0.19	0.29
No EKF + no load	19.1	0.55	0.26	0.40
No EKF + load	34.8	1.03	0.41	0.48

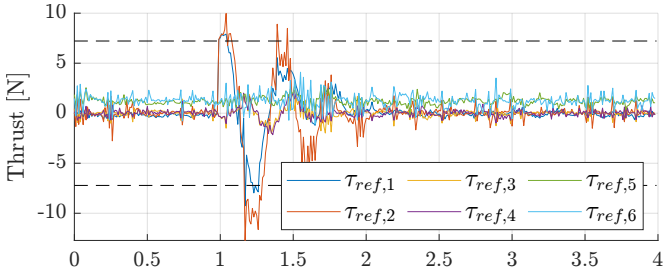
External disturbances $\boldsymbol{d}_{ext}$ can be evaluated at steady state by comparing the reference thrust $\hat{\boldsymbol{\tau}}_{ref}$ estimated by the observer and the actual reference thrust $\boldsymbol{\tau}_{ref}$ sent to the actuators:

$$\boldsymbol{d}_{ext} = \boldsymbol{W}_a(\hat{\boldsymbol{\tau}}_{ref} - \boldsymbol{\tau}_{ref}) \tag{46}$$

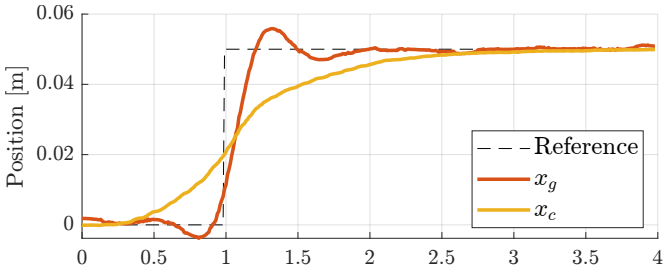
Therefore, evaluating external disturbances before and after the introduction of the load gives an estimate of the latter. With this technique, the estimate of the load is $0.17\,\mathrm{kg}$, which is close. The 13% difference with respect to the original value results from identification errors on matrix $\boldsymbol{W}_a$ and thrust coefficient a.



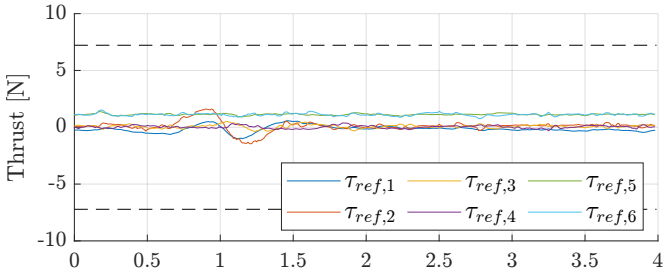
(a) Step along x with PC.



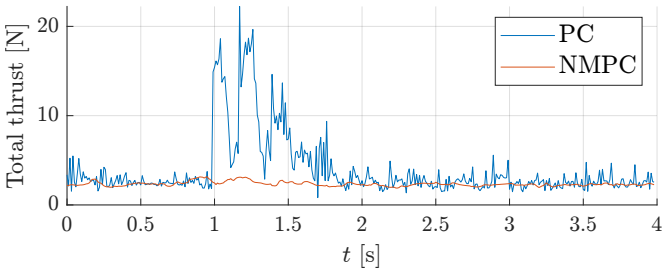
(b) Thrust reference with PC. Dashed lines are saturation values.



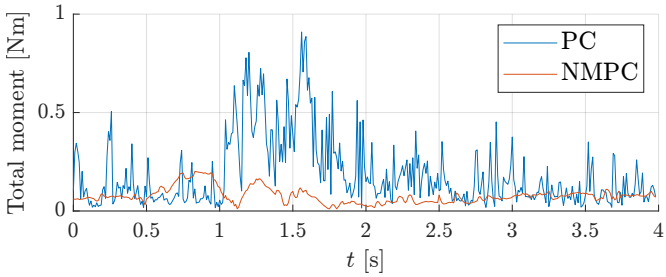
(c) Step along x with NMPC.



(d) Thrust reference with NMPC. Dashed lines are saturation values.

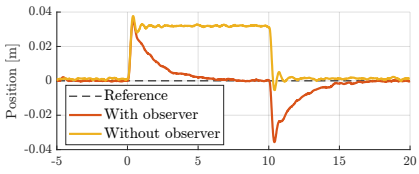


(e) Norm of the total thrust generated by propellers with PC and NMPC.

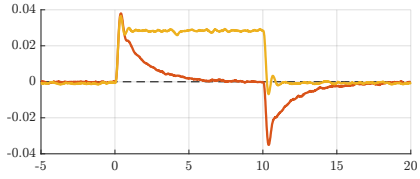


(f) Norm of the total moment at the CoM G generated by propellers with PC and NMPC.

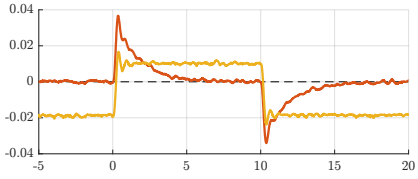
Fig. 9. Step response for x only.



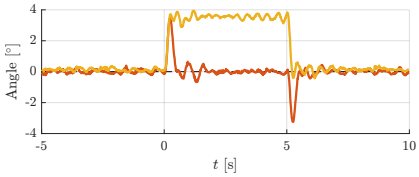
(a) Disturbance on x.



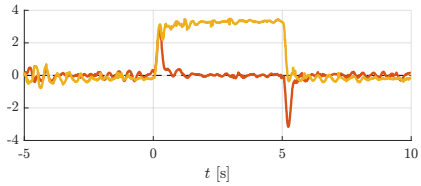
(b) Disturbance on y.



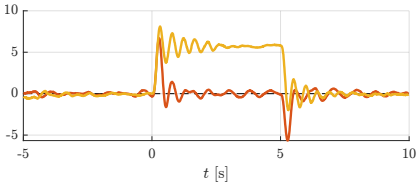
(c) Disturbance on z.



(d) Disturbance on roll.

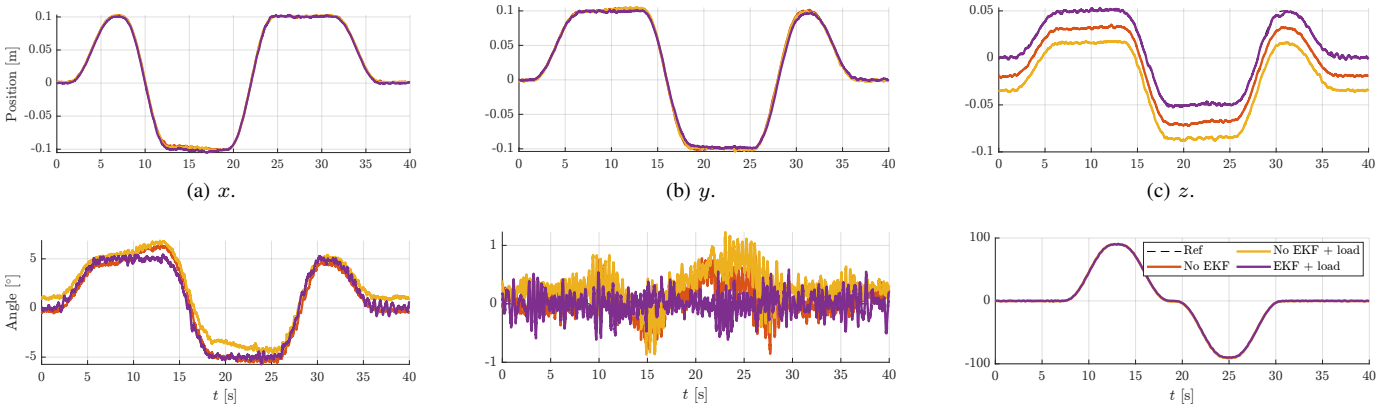


(e) Disturbance on pitch.



(f) Disturbance on yaw.

Fig. 10. Constant disturbance rejection.



(a) x. (b) y. (c) z.

(d) Roll. (e) Pitch. (f) Yaw.

Fig. 11. Trajectory disturbance rejection. Load: 0.15 kg attached at the bottom of the robot.

D. Energy Efficiency

To assess the improvement in power consumption related to the introduction of the carrier and the NMPC, trajectory tracking experiments are carried out at different speeds in three scenarios: (i) partitioned controller with a constant reference $\boldsymbol{p}_{c,ref}$ for a motionless carrier, (ii) partitioned controller with $\boldsymbol{p}_{c,ref}$ as the solution of redundancy resolution strategy (11), (iii) NMPC with $\boldsymbol{p}_{c,ref}$ as the solution of redundancy resolution strategy (11).

Measuring the power consumption of the AM is straightforward using battery voltage U and current i_e measurements from ESCs. The electrical power P_e delivered to an ESC is $P_e = Ui_e$. The power consumption of the CDPR can be estimated by adding the winch mechanical power output and the resistive losses, neglecting motor core losses and drive MOSFETs switching losses. The mechanical power output of a motor is the product of its torque τ_c by its rotational velocity $\dot{\alpha}$. Since the torque is proportional to the current i_c, the mechanical power becomes $P_m = k_{em} i_c \dot{\alpha}$, with $k_{em} = 6.03 \times 10^{-2}\,\mathrm{N\,m/A}$ the torque constant of the motor. Knowing the resistance of the windings $R_m = 1.16\,\Omega$, the resistive losses are $P_c = R_m i_c^2$. Hence, the estimated power consumption of a winch motor is given by $P_{motor} = k_{em} i_c \dot{\alpha} + R_m i_c^2$. The current measurement is given by the motor drive thanks to Hall-effect sensors and the velocity $\dot{\alpha}$ is derived from the motor encoder values. During the experiments, the power consumption of the CDPR is significantly lower than that of the AM, and is neglected in the following results.

Table VII shows the energy consumption during the three scenarios for trajectories with $l_p = 5\,\mathrm{cm}$ and $t_p \in \{2, 4, 6\}\mathrm{s}$. Introducing the carrier reduces by approximately 20% the energy consumption with partitioned controller. Then, replacing partitioned controller by NMPC further reduces energy consumption, almost 60% compared to the first scenario with the partitioned controller and the static carrier. The improvements do not have a significant dependency on the trajectory velocity.

Main factors justifying the efficiency of NMPC over partitioned control are identified in the previous step response analysis. On one hand, the chattering in the control signal is reduced with NMPC thanks to the penalty on the variation of the thrust reference $\boldsymbol{\tau}_{ref}$. As a consequence, core losses on the actuators are reduced. On the other hand, NMPC anticipates slow carrier dynamics. This reduces the moment on the AM due to the suspension spring during transients, yielding lower thrusts and energy consumption.

TABLE VII
ENERGY CONSUMPTION

Scenario	(i) [Wh]	(ii) [Wh]	(iii) [Wh]
$t_p = 2\,\mathrm{s}$	55	43 (-21%)	22 (-59%)
$t_p = 4\,\mathrm{s}$	101	82 (-18%)	44 (-56%)
$t_p = 6\,\mathrm{s}$	153	112 (-26%)	67 (-56%)

VII. Conclusion and Perspectives

This paper addresses the control of a macro-mini aerial manipulator with elastic suspension. An omnidirectional aerial manipulator is suspended by a spring from a carrier with a Cartesian motion. This architecture benefits from the large workspace of the carrier and the high dynamics of the aerial manipulator while improving energy efficiency.

A partitioned controller is first implemented to separately regulate the aerial manipulator pose with a computed torque controller and the carrier position with a kinematic controller. To resolve the redundancy, the desired position of the carrier is determined from the reference pose of the aerial manipulator to reduce energy consumption at steady state. With this strategy, the carrier reduces the energy consumption of the system. However, there is still room for improvements since the carrier lags behind the aerial manipulator because of its slower dynamics.

A nonlinear model predictive controller (NMPC) overcomes the drawbacks of the previous strategy by anticipating the slow carrier dynamics. A sufficient condition for offset-free control is proposed for continuous-time systems and using the delta-input formulation. Experiments validate that there is no steady-state error and the energy consumption is improved.

Future research should consider the inclusion of the energy consumption minimization (currently solved in a dedicated optimization problem) into the NMPC. Thus, the energy consumption could be minimized over the whole trajectory on the prediction horizon, instead of the less computationally expensive but suboptimal solution proposed here.

REFERENCES

[1] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial Manipulation: A Literature Review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 7 2018.

[2] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, Present, and Future of Aerial Robotic Manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 626–645, 2 2022.

[3] H. Bonyan Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation—A literature survey," *Robotics and Autonomous Systems*, vol. 107, pp. 221–235, 9 2018.

[4] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 5 2011, pp. 2491–2498.

[5] 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*. IEEE, 9 2014, pp. 3417–3422.

[6] A. Suarez, P. R. Soria, G. Heredia, B. C. Arrue, and A. Ollero, "Anthropomorphic, compliant and lightweight dual arm system for aerial manipulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 9 2017, pp. 992–997.

[7] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, "Design of multirotor aerial vehicles: A taxonomy based on input allocation," *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 1015–1044, 8 2021.

[8] D. Brescianini and R. D'Andrea, "An omni-directional multirotor vehicle," *Mechatronics*, vol. 55, pp. 76–93, 11 2018.

[9] S. Park, J. Lee, J. Ahn, M. Kim, J. Her, G.-H. Yang, and D. Lee, "ODAR: Aerial Manipulation Platform Enabling Omnidirectional Wrench Generation," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1907–1918, 8 2018.

[10] M. Ryll, D. Bicego, and A. Franchi, "Modeling and control of FAST-Hex: A fully-actuated by synchronized-tilting hexarotor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10 2016, pp. 1689–1694.

[11] M. Brunner, K. Bodie, M. Kamel, M. Pantic, W. Zhang, J. Nieto, and R. Siegwart, "Trajectory Tracking Nonlinear Model Predictive Control for an Overactuated MAV," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2020, pp. 5342–5348.

[12] M. Tognon and A. Franchi, "Omnidirectional Aerial Vehicles With Unidirectional Thrusters: Theory, Optimal Design, and Control," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2277–2282, 7 2018.

[13] ——, *Theory and Applications for Control of Aerial Robots in Physical Interaction Through Tethers*, ser. Springer Tracts in Advanced Robotics. Cham: Springer International Publishing, 2021, vol. 140.

[14] K. Zhang, P. Chermprayong, T. M. Alhinai, R. Siddall, and M. Kovac, "SpiderMAV: Perching and stabilizing micro aerial vehicles with bio-inspired tensile anchoring systems," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 9 2017, pp. 6849–6854.

[15] Y. S. Sarkisov, M. J. Kim, D. Bicego, D. Tsetserukou, C. Ott, A. Franchi, and K. Kondak, "Development of SAM: cable-Suspended Aerial Manipulator *," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2019, pp. 5323–5329.

[16] S. Kim, "Operational quality analysis of parallel manipulators with actuation redundancy," in *Proceedings of International Conference on Robotics and Automation*, vol. 3. IEEE, 1997, pp. 2651–2656.

[17] F. Lallman, J. Davidson, and W. Bundick, "Integrated reconfigurable control allocation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Reston, Virigina: American Institute of Aeronautics and Astronautics, 8 2001.

[18] Yu Luo, A. Serrani, S. Yurkovich, D. Doman, and M. Oppenheimer, "Model predictive dynamic control allocation with actuator dynamics," in *Proceedings of the 2004 American Control Conference*. IEEE, 2004, pp. 1695–1700.

[19] L. Cuvillon, M. A. Perozo, A. Yiğit, S. Durand, and J. Gangloff, "Offset-free nonlinear model predictive control for improving dynamics of cable-driven parallel robots with on-board thrusters," *Mechanism and Machine Theory*, vol. 180, p. 105141, 2 2023.

[20] H. Nguyen, M. Kamel, K. Alexis, and R. Siegwart, "Model Predictive Control for Micro Aerial Vehicles: A Survey," in *Proceedings of the IEEE European Control Conference*, 2021, pp. 1556–1563.

[21] U. Maeder, F. Borrelli, and M. Morari, "Linear offset-free Model Predictive Control," *Automatica*, vol. 45, no. 10, pp. 2214–2222, 10 2009.

[22] M. Morari and U. Maeder, "Nonlinear offset-free model predictive control," *Automatica*, vol. 48, no. 9, pp. 2059–2067, 9 2012.

[23] G. Pannocchia, M. Gabiccini, and A. Artoni, "Offset-free MPC explained: novelties, subtleties, and applications," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 342–351, 2015.

[24] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear Model Predictive Control with Enhanced Actuator Model for Multi-Rotor Aerial Vehicles with Generic Designs," *Journal of Intelligent & Robotic Systems*, 9 2020.

[25] A. Yiğit, M. Arpa Perozo, L. Cuvillon, S. Durand, and J. Gangloff, "Improving Dynamics of an Aerial Manipulator with Elastic Suspension Using Nonlinear Model Predictive Control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2021.

[26] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A Comparative Study of Nonlinear MPC and Differential-Flatness-Based Control for Quadrotor Agile Flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, 12 2022.

[27] A. Yiğit, M. Arpa Perozo, L. Cuvillon, S. Durand, and J. Gangloff, "Novel Omnidirectional Aerial Manipulator With Elastic Suspension: Dynamic Control and Experimental Performance Assessment," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 612–619, 2021.

[28] "Arrêté du 17 décembre 2015 relatif à la conception des aéronefs civils qui circulent sans personne à bord, aux conditions de leur emploi et aux capacités requises des personnes qui les utilisent."

[29] A. Yiğit, M. Arpa Perozo, M. Ouafo, L. Cuvillon, S. Durand, and J. Gangloff, "Aerial Manipulator Suspended From a Cable-Driven Parallel Robot: Preliminary Experimental Results," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[30] R. Mahony, V. Kumar, and P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 9 2012.

[31] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, 10 2021.

[32] R. Kelly and J. Moreno, "Manipulator motion control in operational space using joint velocity inner loops," *Automatica*, vol. 41, no. 8, pp. 1423–1432, 8 2005.

[33] I. Soto and R. Campa, "Two-Loop Control of Redundant Manipulators: Analysis and Experiments on a 3-DOF Planar Arm," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 85, 1 2013.

[34] R. Campa, J. Bernal, and I. Soto, "Kinematic modeling and control of the Hexapod parallel robot," in *2016 American Control Conference (ACC)*. IEEE, 7 2016, pp. 1203–1208.

[35] R. Chellal, L. Cuvillon, and E. Laroche, "A kinematic vision-based position control of a 6-DoF cable-driven parallel robot," in *Mechanisms and Machine Science*. Springer, Cham, 2015, vol. 32, pp. 213–225.

[36] S. Vichik and F. Borrelli, "Solving linear and quadratic programs with an analog circuit," *Computers & Chemical Engineering*, vol. 70, pp. 160–171, 11 2014.

[37] G. Grimm, M. Messina, S. Tuna, and A. Teel, "Model predictive control: for want of a local control Lyapunov function, all is not lost," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 546–558, 5 2005.

[38] J. Gangloff, A. Yiğit, and M. Lesellier, "RPIt," 2020.

[39] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "PAMPC: Perception-Aware Model Predictive Control for Quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10 2018, pp. 1–8.

[40] M. Jacquet, M. Kivits, H. Das, and A. Franchi, "Motor-Level N-MPC for Cooperative Active Perception With Multiple Heterogeneous UAVs," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2063–2070, 4 2022.

[41] M. Gouttefarde, J.-F. Collard, N. Riehl, and C. Baradat, "Geometry Selection of a Redundantly Actuated Cable-Suspended Parallel Robot," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 501–510, 4 2015.

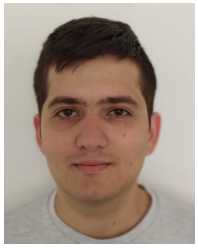
[42] D. Bury, J.-B. Izard, M. Gouttefarde, and F. Lamiraux, "Continuous Collision Detection for a Robotic Arm Mounted on a Cable-Driven Parallel Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11 2019, pp. 8097–8102.
[43] R. Chellal, L. Cuvillon, and E. Laroche, "Model identification and vision-based H_∞ position control of 6-DoF cable-driven parallel robots," *International Journal of Control*, vol. 90, no. 4, pp. 684–701, 4 2017.
[44] M. A. Khosravi and H. D. Taghirad, "Dynamic Analysis and Control of Cable Driven Robots with Elastic Cables," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 35, no. 4, pp. 543–557, 12 2011.
[45] G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 3 2020.
[46] B. Houska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, 10 2011.

Arda Yiğit received the B.Eng. degree in mechatronics from the Institut National des Sciences Appliquées (INSA), Strasbourg, France, in 2018, and the M.Sc. and Ph.D. degrees in robotics from the University of Strasbourg, Strasbourg, France, in 2018 and 2021, respectively.

Since 2022, he is a Postdoctoral Fellow at Laval University, Québec, Canada. His main research interests include the design and control aerial and parallel robots, as well as physical human-robot interaction.

Loïc Cuvillon was born in France in 1979. He received the M.Sc. and Ph.D. degrees in robotics from the University Louis Pasteur, Strasbourg, in 2002 and 2006, respectively.

Since 2008, he is an Associate Professor at the University of Strasbourg. He is currently a member of the Robotics, Data science and Healthcare Technologies team at the ICube Laboratory, Strasbourg. His research interests include medical robotics, visual servoing of robotic manipulator, control of flexible structures and, more recently, cable-driven parallel robots.



Miguel Arpa Perozo received the B.Eng. degree in mechatronics from the Institut National des Sciences Appliquées (INSA), Strasbourg, France, in 2020, and the M.Sc. degrees in robotics from Strasbourg University, Strasbourg, France, in 2020.

Since 2020, he is pursuing a robotics Ph.D. at the University of Strasbourg under the supervision of Prof. Jacques Gangloff.

Sylvain Durand received the M.Sc.Eng. degree in control and embedded systems from Grenoble-INP, in 2007, and the Ph.D. degree in systems and control theory from the University of Grenoble, in 2011, Grenoble, France. He is currently an Associate Professor with INSA Strasbourg and the ICube laboratory in Strasbourg, France.

His research interests include event-based control and control of embedded cyber-physical and robotic systems with limited resources.

Jacques Gangloff was born in Strasbourg, France, in 1969. He graduated from the Ecole Normale Supérieure de Cachan, Cachan, France, in 1995. He received the M.S. and Ph.D. degrees in robotics from the University Louis Pasteur, Strasbourg, France, in 1996 and 1999, respectively.

Between 1999 and 2005, he was an associate professor at the University of Strasbourg. He is currently a full professor at the same University and a member of the Robotics, Data Science and Healthcare Technologies team at the ICube laboratory. His research interests include visual servoing, predictive control, medical robotics, cable driven parallel robotics and, more recently, aerial manipulation. He has authored or coauthored over 120 publications and received numerous awards.