

Towards Aerial Physical Locomotion: the Contact-Fly-Contact Problem

Quentin Delamare, Paolo Robuffo Giordano, Antonio Franchi

▶ **To cite this version:**

Quentin Delamare, Paolo Robuffo Giordano, Antonio Franchi. Towards Aerial Physical Locomotion: the Contact-Fly-Contact Problem. IEEE Robotics and Automation Letters, 2018, 3 (3), pp.1514-1521. 10.1109/LRA.2018.2800798 . hal-01704043

HAL Id: hal-01704043
https://laas.hal.science/hal-01704043

Submitted on 8 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards Aerial Physical Locomotion: the Contact-Fly-Contact Problem

Quentin Delamare[1], Paolo Robuffo Giordano[1], and Antonio Franchi[2]

***Abstract*— In this paper we consider the problem of letting an aerial robot exploiting its contact with the environment in order to enhance its motion possibilities, in a way reminiscent of legged robots exploiting contact forces for locomotion purposes. As a representative and initial case study, we consider a quadrotor equipped with a 1-DOF arm able to hook at some pivot points, and needing to perform a maneuver from an initial hooked configuration to a final hooked configuration while passing though a free-flight phase between the two anchor points. To this end, we propose a dynamical modeling able to capture the various phases (hooked, free-flying) together with an optimization framework for generating optimal motion plans compatible with actuation constraints. Simulation results illustrate the effectiveness of the approach and the promising potential in terms of more advanced maneuvers.**

I. INTRODUCITON

The field of aerial robotics is being explored actively thanks to the huge technological developments of the last decades. Formulating specific control laws for Unmanned Aerial Vehicles (UAVs), such as multi-rotors, has been a challenging goal because of their typical underactuation and complex aerodynamics. Several methods have been proposed over the last decades for solving the pose regulation and trajectory tracking problem, which now allow to successfully handle the motion control of UAVs in many different conditions. However these control approaches are mainly intended for sensing and mapping applications in open space.

In recent years, the field of aerial *physical* interaction has also gained an increasing popularity, fostered, among others, by application-oriented large-scale projects, like AiRobots, ARCAS, AeroArms, and AeroWorks. In this kind of scenarios the UAVs are expected to apply a controlled wrench on their environment or on an object while flying. Examples range from tool operation [1], [2] to tethered flight [3], [4], and from surface inspection [5], [6] to aerial manipulation with articulated arms [7]–[9]. Differently from typical aerial surveillance/monitoring tasks, which can be performed by remaining far enough from the environment, in the aerial physical interaction domain the aerial robots are forced to move close to the environment, including the parts with which no interaction is required. Additionally, there are many cases even in the contact-free aerial domain in which the UAVs need to operate in the vicinity of obstacles, such as in close-view monitoring of complex areas, in search and rescue missions of disaster areas, when flying through forests, or in cluttered urban environments.

In both cases (the contact-free and the contact-enabled ones), the surrounding environment has generally been considered as an undesired constraint for the vehicle's motion: the need of free/fast motion and of safely performing physical interaction has most often been perceived as antagonistic w.r.t. the presence of other undesired physical objects. This is demonstrated by the fact that, when it comes to trajectory generation and control, physical objects are typically considered as regions to be either avoided via complex and potentially dangerous maneuvers [10], or to be exploited during 'resting phases', i.e., in absence of motion like in perching [11] and landing [12]. On the other hand, by taking a completely different perspective, the legged robot community has always considered motion to be intimately related to contact: the first cannot exist without the latter. Legged robots take advantage of the possible contacts with the environment instead of avoiding them, since contact forces are necessary for performing locomotion [13], [14].

Inspired by the use of contacts in legged robots, in this paper we propose the idea of a paradigm shift for the aerial community, that is, to exploit physical contact with the environment for the purpose of 'locomotion' during flight. In other words we believe that the presence of obstacles should be regarded also by aerial robots as a source of possible contact points and, thus, as an opportunity rather than as a disadvantage. Indeed by leveraging the possibility to exploit contacts with the environment, a full range of advantages could become available to aerial robots: 1) the platform could exploit the contact forces saving thrust w.r.t. a full-flight modality. The reduced thrust can be exploited, e.g., 2) to save energy, or to use the extra margin available in order 3) to increase the range of possible maneuvers, achieving an unprecedented agility, as done by athletes with parkour, 4) to significantly increase the set of reachable points in a cluttered environment, 5) to augment the safety, and in general 6) to increase the robustness versus disturbances and unpredicted events. Or, simply, one could 7) exploit the reduced actuation needs for adopting smaller and cheaper motors, or 8) to use the same motors and increase the payload.

The locomotion problem on the ground at normal speeds represents still nowadays a difficult and not completely solved problem, as it has been shown in the DARPA Robotics Challenge. In the case of *aerial physical locomotion* the robot needs to fly, to cope with possibly high impacts, to take into consideration the tighter limits and larger uncertainty of aerodynamic actuators, and to cope with its underactuation. The goal of this paper is, therefore, to propose a first step

This work has been partially funded by the European Union H2020 program under grant agreement No 644271 AEROARMS.

[1]CNRS, Univ Rennes, Inria, IRISA, Rennes, France, {quentin.delamare,prg}@irisa.fr

[2]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, antonio.franchi@laas.fr

towards the solution of this complex problem: in particular in this work we tackle a first basic, yet significative, problem of performing *contact-fly-contact* maneuvers in an optimal way. The chosen problem has a dual role: it will be used as building block in future developments for implementing more complex maneuvers, and it will also be useful in understanding the main bottlenecks and critical points of the new proposed direction of 'aerial locomotion'.

Specifically, we propose to study the case of a planar quadrotor equipped with an onboard 1-DOF arm: the quadrotor needs to execute a maneuver from an initial configuration in contact with the environment (the arm 'hooks' a first pivot point) to a final configuration with another contact (the arm hooks a second pivot point), by passing through an intermediate 'free-flying' phase. To this end, we propose a dynamical model for the quadrotor/arm system in both phases (hooked and free-flying) that also considers the effects of possible impacts with the environment when performing a hooking during flight. Then, an optimization algorithm able to generate optimal trajectories linking two given initial and final configurations (while coping with actuation constraints) is proposed: (classical) optimality vs. execution time is considered, as well as concurrent optimality vs. the sensitivity of the state evolution against parametric uncertainties (for generating intrinsically robust motion plans). Subsequently, two tracking controllers for the hooked and free-flight phases are presented, and several simulation results validate the overall approach and illustrate the main features. The proposed combination of quadrotor/arm is hereafter denoted as *MonkeyRotor* because of its resemblance with a monkey exploiting its arms for attaching to a branch and jumping from one branch to the next one.

The paper is structured as follows. In Sect. II we introduce the dynamical modeling of the MonkeyRotor in its different phases and Sect. III presents the adopted trajectory planning strategy for solving the sought hook-fly-hook maneuver. Then, in Sect. IV we design the control laws able to track the generated optimal trajectories in both phases (hooked and free-flight), and simulation results are then illustrated in Sect. V. Finally, Sect. VI concludes the paper and discusses possible future works.

II. DYNAMICAL MODELING

The MonkeyRotor consists of a quadrotor UAV equipped with an actuated 1-DOF arm meant to grasp a pivot point (e.g., a branch) in the environment with its end-effector. In this section we illustrate the dynamical model of the MonkeyRotor during each phase (hooked and free-flight) by borrowing from our previous works [4], [9] which have considered similar scenarios (in particular, [9] has considered a quadrotor with actuated arm but only in free-flight, while [4] has considered the hooked case but with a passive arm). As already done in many previous works on similar subjects, see, e.g., [4], [9], [15]–[17], we restrict the analysis to the vertical plane.

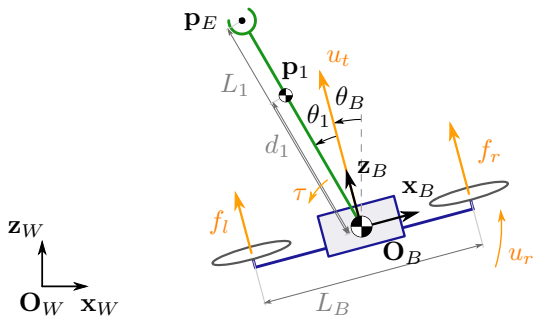


Fig. 1: Geometry of the MonkeyRotor, a flying robot with an actuated arm.

A. Definitions

With reference to Fig. 1, let $\mathcal{F}_W$ be an inertial world frame with axes $\{\mathbf{x}_W, \mathbf{z}_W\}$ and origin $\mathbf{O}_W$, and $\mathcal{F}_B$ a body frame attached to the quadrotor with axes $\{\mathbf{x}_B, \mathbf{z}_B\}$: the axis $\mathbf{z}_B$ represents the body-frame thrust direction, and the origin $\mathbf{O}_B$ is placed at the quadrotor center of mass (CoM). The configuration of the quadrotor can be specified by the position of $\mathbf{O}_B$ in $\mathcal{F}_W$, denoted as $\mathbf{p}_B = [x_B \; z_B]^T \in \mathbb{R}^2$, and the orientation of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$ here parametrized by the angle θ_B from $\mathbf{z}_W$ to $\mathbf{z}_B$. The arm is assumed to have length L_1 and to have its joint mounted at $\mathbf{p}_B$ (the quadrotor CoM), around which it can rotate by an angle θ_1 (the angle from $\mathbf{z}_B$ to the arm direction). The CoM of the arm, denoted as $\mathbf{p}_1$, is placed at a distance d_1 from $\mathbf{O}_B$. The configuration of the whole MonkeyRotor (quadrotor + arm) is then denoted as $\mathbf{q} = [\mathbf{p}_B^T \; \boldsymbol{\theta}^T]^T \in \mathbb{R}^4$ where we let $\boldsymbol{\theta} = [\theta_B \; \theta_1]^T$.

The quadrotor is equipped with two propellers generating two thrust vectors $f_l \mathbf{z}_B$ and $f_r \mathbf{z}_B$: the forces produced by the propellers result in a total thrust vector $u_t \mathbf{z}_B = (f_r + f_l)\mathbf{z}_B$ and torque $u_r = \frac{L_B}{2}(f_r - f_l)$, with L_B being the distance between the two propellers. The arm is also assumed actuated by a torque τ acting at $\mathbf{O}_B$. These three inputs for the whole MonkeyRotor are then denoted as $\mathbf{u} = [u_t \; u_r \; \tau]^T \in \mathbb{R}^3$. For convenience, we also define the alternative input vector $\mathbf{u}_f = [f_r \; f_l \; \tau] = \mathbf{K}\mathbf{u}$ where

$$\mathbf{K} = \begin{bmatrix} 1/2 & 1/L_B & 0 \\ 1/2 & -1/L_B & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

Indeed, while the MonkeyRotor dynamics are more naturally expressed in terms of the input vector $\mathbf{u}$, the physical actuation constraints (min/max joint torque and propeller thrusts) affect the input $\mathbf{u}_f$. This distinction will be important in the next developments. We finally let m_B, J_B, m_1, J_1 be the mass and inertia of the quadrotor and arm, respectively.

We now describe the dynamical model of the MonkeyRotor in the two considered phases of hooked and free-flight.

B. Hooked phase

Let

$$\mathbf{p}_E = \mathbf{p}_B + L_1 \begin{bmatrix} -\sin(\theta_1 + \theta_B) \\ \cos(\theta_1 + \theta_B) \end{bmatrix} \tag{2}$$

represent the position of the arm end-effector in $\mathcal{F}_W$ and $\mathbf{p}_E^* \in \mathbb{R}^2$ the (fixed) position of the hook in $\mathcal{F}_W$. Following [4], the hook constraint $\mathbf{p}_E(\mathbf{q}) = \mathbf{p}_E^*$ restricts the MonkeyRotor motion to a circle centered at $\mathbf{p}_E^*$. In this constrained case the MonkeyRotor configuration is fully determined by the configuration variables $\boldsymbol{\theta}$: by applying standard techniques (Euler-Lagrange procedure), one can then obtain the following (reduced) dynamical model governing the behavior of the states $(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$

$$\mathbf{M}_h(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{g}_h(\boldsymbol{\theta}) = \mathbf{G}_h(\boldsymbol{\theta})\mathbf{u} \tag{3}$$

where

$$\mathbf{M}_h(\boldsymbol{\theta}) = \begin{bmatrix} J_B & 0 \\ 0 & J_1 + m_B L_1^2 + m_1(L_1 - d_1)^2 \end{bmatrix}, \tag{4}$$

$$\mathbf{G}_h(\boldsymbol{\theta}) = \begin{bmatrix} 0 & 1 & -1 \\ L_1 \sin(\theta_1) & 0 & 1 \end{bmatrix}, \tag{5}$$

and $\mathbf{g}_h(\boldsymbol{\theta}) = [0 \ (m_B L_1 + m_1(L_1 - d_1))g \sin(\theta_B + \theta_1)]^\top$. Since matrix $\mathbf{G}_h(\boldsymbol{\theta})$ is always full rank, the constrained MonkeyRotor is *overactuated*, with two controlled variables $\boldsymbol{\theta}$ for the three control inputs $\mathbf{u}$. We note that in [4] the joint arm was considered passive ($\tau = 0$) thus resulting in a fully-actuated system with a singularity for $\theta_1 = 0$ as opposed to the case under consideration (which is singularity-free). Sect. IV-A will elaborate more about the possible use of the MonkeyRotor overactuation.

The behavior of the remaining MonkeyRotor states $(\mathbf{p}_B, \dot{\mathbf{p}}_B)$ can then be algebraically expressed as a function of $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ by exploiting the hook constraint $\mathbf{p}_E(\mathbf{q}) = \mathbf{p}_E^*$ as

$$\mathbf{p}_B = \mathbf{p}_E^* - L_1 \begin{bmatrix} -\sin(\theta_1 + \theta_B) \\ \cos(\theta_1 + \theta_B) \end{bmatrix} \tag{6}$$

and

$$\dot{\mathbf{p}}_B = L_1(\dot{\theta}_1 + \dot{\theta}_B) \begin{bmatrix} \cos(\theta_1 + \theta_B) \\ \sin(\theta_1 + \theta_B) \end{bmatrix}. \tag{7}$$

C. Free-flying phase

The free-flying dynamical model of the MonkeyRotor is a particular case of the system presented in [9]. In particular one has

$$\mathbf{M}_f(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}_f(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_f(\mathbf{q}) = \mathbf{G}_f\mathbf{u}, \tag{8}$$

where the expression of the various terms can be found in [9].

We note that, as opposed to the hooked scenario, the MonkeyRotor is *underactuated* during free-flight (three inputs $\mathbf{u}$ for four configuration variables $\mathbf{q}$). However, as discussed in [9], it is possible to find a *flat output* or *linearizing output* [18] which allows for full dynamic linearization of the system dynamics. More details about this point are given in Sect. IV-B.

D. Impact Model

Let t_h be the time at which the MonkeyRotor switches from a free-flight phase to a hooked phase because the end-effector has reached the pivot location $\mathbf{p}_E^*$ and performed a successful hook. If $\dot{\mathbf{p}}_E(t_h^-) \neq \mathbf{0}$ (the end-effector velocity is non-zero just before hooking), a sudden impact will occur

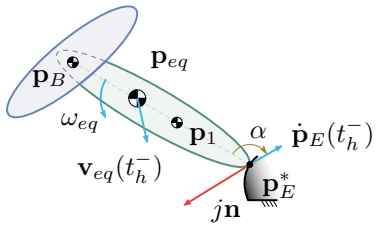


Fig. 2: Notations for the collision model. The large and 'instantaneous' reaction force at E is synthetized in the impulse vector $j\mathbf{n}$.

affecting the evolution of the MonkeyRotor state $(\mathbf{q}, \dot{\mathbf{q}})$. The goal of this section is to propose a simple impact model based on impulse theory (see, e.g., [19]) able to capture the instantaneous change from $\dot{\mathbf{q}}(t_h^-)$ to $\dot{\mathbf{q}}(t_h^+)$ because of a possible collision between the end-effector and the hook[1]. Availability of this impact model will then allow the trajectory planning algorithm discussed in the next section to be 'aware' of the effects of a possible collision and, thus, generate more realistic motion plans that can also take advantage of the (controlled) collision between end-effector and pivot (e.g., for quickly reducing the system kinetic energy).

Recalling that $\dot{\mathbf{q}} = [\dot{\mathbf{p}}_B^T \ \dot{\boldsymbol{\theta}}^T]^T$, we first consider the effects on $\dot{\boldsymbol{\theta}} = [\dot{\theta}_B \ \dot{\theta}_1]^T$. First of all, we remark that the choice of placing the joint base at the quadrotor CoM (a property also known as *protocentricity* [8]) implies that the rotational dynamics of the quadrotor base is completely decoupled from the dynamics of the collision. Therefore one has $\dot{\theta}_B(t_h^+) = \dot{\theta}_B(t_h^-)$. Concerning $\dot{\theta}_1(t_h^+)$, we can proceed as follows: for the sake of impact modeling, one can assimilate the MonkeyRotor to an equivalent body with mass $m_{eq} = m_B + m_1$, CoM $\mathbf{p}_{eq} = \dfrac{m_B\mathbf{p}_B + m_1\mathbf{p}_1}{m_{eq}}$, inertia $J_{eq} = J_1 + m_1\|\mathbf{p}_1 - \mathbf{p}_{eq}\|^2 + m_B\|\mathbf{p}_B - \mathbf{p}_{eq}\|^2$, and with an equivalent linear velocity $\mathbf{v}_{eq} = \dot{\mathbf{p}}_{eq}$ and angular velocity $\omega_{eq} = \dot{\theta}_B + \dot{\theta}_1$. Let now $\mathbf{j} = j\mathbf{n}$ be the momentum exchanged by the end-effector and the pivot during the collision, with $\mathbf{n}$ (the direction of $\mathbf{j}$) taken as $\mathbf{n} = -\dot{\mathbf{p}}_E(t_h^-)/\|\dot{\mathbf{p}}_E(t_h^-)\|$, see Fig. 2 (therefore, the direction $\mathbf{n}$ is determined by the MonkeyRotor state at t_h^-).

The change in the linear and angular velocities $\mathbf{v_{eq}}$ and ω_{eq} before and after the collision can be modeled as

$$\mathbf{v}_{eq}(t_h^+) = \mathbf{v}_{eq}(t_h^-) + \frac{\mathbf{j}}{m_{eq}} \tag{9}$$

$$\omega_{eq}(t_h^+) = \omega_{eq}(t_h^-) + \frac{j}{J_{eq}}\|\mathbf{p}_E^* - \mathbf{p}_{eq}\| \sin\alpha \tag{10}$$

where $\mathbf{p}_E^*$ is the location of the pivot point (where the collision occurs) and α is the angle between vectors $\mathbf{p}_E^* - \mathbf{p}_{eq}(t_h^-)$ and $\mathbf{n}$. By combining (9–10) with the kinematics

[1]As customary, we assume continuity of $\mathbf{q}$, i.e., $\mathbf{q}(t_h^-) = \mathbf{q}(t_h^+)$, in presence of an instantaneous impact [19].

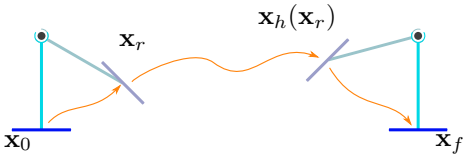


Fig. 3: Optimization scheme, where $\mathbf{x}_0$ is the initial state, $\mathbf{x}_r$ the transition state where the system passes from its hooked dynamics to its free flying one, $\mathbf{x}_h$ the reciprocal one and $\mathbf{x}_f$ the final state.

relationships

$$\begin{cases} \mathbf{v}_{eq}(t_h^+) = \dot{\mathbf{p}}_1(t_h^+) + \mathbf{S}(\omega(t_h^+)) \cdot (\mathbf{p}_1 - \mathbf{p}_{eq}) \\ \dot{\mathbf{p}}_E(t_h^+) = \dot{\mathbf{p}}_1(t_h^+) + \mathbf{S}(\omega(t_h^+)) \cdot (\mathbf{p}_1 - \mathbf{p}_E) \end{cases} \tag{11}$$

where $\mathbf{S}(a) = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}$, and by using the fact that $\dot{\mathbf{p}}_E(t_h^+) = \mathbf{0}$ (the end-effector velocity is zero after the impact), one can solve for $j = \|\mathbf{j}\|$ as

$$j = \frac{\|\dot{\mathbf{p}}_E(t_h^-)\| m_{eq}}{1 + \dfrac{\|\mathbf{p}_{eq}(t_h^-) - \mathbf{p}_E^*\|^2 m_{eq}}{J_{eq}} \sin\alpha}. \tag{12}$$

Note that j can be expressed in terms of only known quantities, in particular the MonkeyRotor state $(\mathbf{q}(t_h^-), \dot{\mathbf{q}}(t_h^-))$ just before the collision. By now plugging (12) in (10) yields the value of $\omega_{eq}(t_h^+) = \dot{\theta}_B(t_h^+) + \dot{\theta}_1(t_h^+)$, which in turn determines $\dot{\theta}_1(t_h^+)$ since, as explained before, $\dot{\theta}_B(t_h^+)$ is known. Having obtained $\dot{\theta}_B(t_h^+)$ and $\dot{\theta}_1(t_h^+)$, the relationship (7) finally allows determining the remaining $\dot{\mathbf{p}}_B(t_h^+)$ and, thus, the whole vector $\dot{\mathbf{q}}(t_h^+)$ as sought.

III. TRAJECTORY PLANNING

In this Section we discuss a trajectory planning strategy meant to generate feasible trajectories for letting the MonkeyRotor passing from a hooked configuration to another hooked configuration. Figure 3 depicts the considered scenario: let $\mathbf{x} = [\mathbf{q}^T \ \dot{\mathbf{q}}^T]^T \in \mathbb{R}^n$, $n = 8$, represent the MonkeyRotor state, and assume two initial and final states $\mathbf{x_0}, \mathbf{x_f}$ are given corresponding to the MonkeyRotor hovering stationary while hooked to the initial and final pivot point. Let also

$$\underline{\mathbf{u}}_f \leq \mathbf{u}_f \leq \bar{\mathbf{u}}_f \tag{13}$$

represent the actuation constraints on the MonkeyRotor input $\mathbf{u}_f = \mathbf{K}\mathbf{u}$ (see (1)). The goal is to find an optimal (w.r.t. a cost of interest) and feasible trajectory for the pair $(\mathbf{x}(t), \mathbf{u}(t))$ over a time interval $t = [t_0, t_f]$ able to bring the MonkeyRotor from $\mathbf{x}(t_0) = \mathbf{x_0}$ to $\mathbf{x}(t_f) = \mathbf{x_f}$ while coping with the actuation constraints. Depending on the conditions (initial/final states, actuation constraints), one can expect the optimal trajectory to involve an initial 'swinging' (attached to the first pivot point) until the hook is released (state $\mathbf{x}_r$ in Fig. 3), followed by a free-flying phase, and subsequently a possible final 'swinging' when re-hooking with the next pivot point (state $\mathbf{x}_h$ in Fig. 3). Indeed these swinging maneuvers can be exploited for efficiently building up/losing energy, thus fully exploiting the possibility to actively exchange forces with the environment (as in a locomotion task) in addition to the available thrust/torque inputs.

The complexity of this optimization problem, also due to the change in the MonkeyRotor dynamics when switching from a hooked phase to a free-flying phase, does not allow for an analytical solution (i.e., finding the complete optimal trajectory over $t = [t_0, t_f]$). Therefore, a numerical optimization method needs to be employed: among the many possible strategies, we now discuss the adopted (possibly sub-optimal) one which we found amenable to a numerical resolution.

A. Optimization procedure

In order to simplify the optimization problem, we split it in two loops: the inner loop looks for an optimal trajectory *given* a candidate release state $\mathbf{x}_r$. The outer loop then tries to optimize the candidate $\mathbf{x}_r$.

1) Inner loop: given a candidate release state $\mathbf{x}_r$ and a cost function $J_1(\mathbf{x})$ (to be specified later on), this first optimization problem

$$\begin{aligned} J_1^*(\mathbf{x}_r) = \min_{\mathbf{u}(t),\, t\in[t_0,\, t_r]} \quad & J_1(\mathbf{x}) \\ \text{subject to} \quad & \dot{\mathbf{x}} = \mathbf{f_h}(\mathbf{x}) + \mathbf{G_h}(\mathbf{x})\mathbf{u} \\ & \mathbf{x}(t_0) = \mathbf{x}_0 \\ & \mathbf{x}(t_r) = \mathbf{x}_r \\ & \underline{\mathbf{u}}_f \leq \mathbf{K}\mathbf{u} \leq \bar{\mathbf{u}}_f \end{aligned}$$

returns the optimal trajectory w.r.t. the cost $J_1(\mathbf{x})$ for joining $\mathbf{x}(t_0) = \mathbf{x}_0$ with $\mathbf{x}(t_r) = \mathbf{x}_r$ at some release time $t_r > t_0$ to be determined by the optimization algorithm. Here, $\dot{\mathbf{x}} = \mathbf{f_h}(\mathbf{x}) + \mathbf{G_h}(\mathbf{x})\mathbf{u}$ is a shorthand for the MonkeyRotor constrained dynamics (3)–(6–7). Note also that the optimal cost $J_1^*(\mathbf{x}_r)$ is a function of the release state $\mathbf{x}_r$.

Subsequently, this second optimization problem

$$\begin{aligned} J_2^*(\mathbf{x}_r) = \min_{\mathbf{u}(t),\, t\in[t_r,\, t_h]} \quad & J_2(\mathbf{x}) \\ \text{subject to} \quad & \dot{\mathbf{x}} = \mathbf{f_f}(\mathbf{x}) + \mathbf{G_f}(\mathbf{x})\mathbf{u} \\ & \mathbf{x}(t_r) = \mathbf{x}_r \\ & \mathbf{p}_E(t_h) = \mathbf{p}_E^* \\ & \|\dot{\mathbf{p}}_E(t_h)\| \leq \mathbf{v}_{max} \\ & \underline{\mathbf{u}}_f \leq \mathbf{K}\mathbf{u} \leq \bar{\mathbf{u}}_f \end{aligned}$$

finds an optimal trajectory for bringing the (now free-flying) MonkeyRotor from $\mathbf{x}(t_r) = \mathbf{x}_r$ to a hooked state with the second pivot point represented by the hook constraint $\mathbf{p}_E(t_h) = \mathbf{p}_E^*$, where $t_h > t_r$ (the hooking time) is to be determined by the optimization. Here, similarly to before, the notation $\dot{\mathbf{x}} = \mathbf{f_f}(\mathbf{x}) + \mathbf{G_f}(\mathbf{x})\mathbf{u}$ is a shorthand for the free-flying MonkeyRotor dynamics (8).

Note that the expected constraint $\dot{\mathbf{p}}_E(t_h) = \mathbf{0}$ (null end-effector velocity when hooking) is here replaced by the milder $\|\dot{\mathbf{p}}_E(t_h)\| \leq v_{max}$, with $v_{max} > 0$ being a small positive threshold. Indeed, we empirically found that accepting a nonzero (but small) $\|\dot{\mathbf{p}}_E(t_h)\|$ facilitates the optimization procedure since the optimal trajectory is allowed to 'exploit'

a hard (but controlled) impact with the pivot for quickly reducing the system energy without spending control effort, in a way, again, reminiscent of how humans/animals exploit contact when moving (we note that the effects of a possible nonzero $\|\dot{\mathbf{p}}_E(t_h)\|$ are taken into account by the impact modeling discussed in Sect. II-D). Finally, note that the optimal cost $J_2^*(\mathbf{x}_r)$ and the whole optimal state evolution $\mathbf{x}^*(t)$, $t \in [t_r,\, t_h]$, are again a function of the release state $\mathbf{x}_r$. We will then denote with $\mathbf{x}_h(t_h;\ \mathbf{x}_r)$ the final hook state reached at t_h as a function of the release state $\mathbf{x}_r$.

Finally, this third optimization problem

$$\begin{aligned} J_3^*(\mathbf{x}_r) = \min_{\mathbf{u}(t),\, t\in[t_h,\, t_f]} \quad & J_3(\mathbf{x}) \\ \text{subject to} \quad & \dot{\mathbf{x}} = \mathbf{f_h}(\mathbf{x}) + \mathbf{G_h}(\mathbf{x})\mathbf{u} \\ & \mathbf{x}(t_h) = \boldsymbol{\Gamma}(\mathbf{x}_h(t_h^-;\ \mathbf{x}_r)) \\ & \mathbf{x}(t_f) = \mathbf{x}_f \\ & \underline{\mathbf{u}}_f \le \mathbf{K}\mathbf{u} \le \bar{\mathbf{u}}_f \end{aligned}$$

finds an optimal trajectory for bringing the (now hooked) MonkeyRotor from $\mathbf{x}(t_h)$ to the final state $\mathbf{x}(t_f) = \mathbf{x}_f$, where $t_f > t_h$ is to be determined by the optimization algorithm. Here $\boldsymbol{\Gamma}(\mathbf{x}_h(t_h^-;\ \mathbf{x}_r))$ is a shorthand for the reset action performed by the collision model of Sect. II-D because of the (possibly) nonzero $\dot{\mathbf{p}}_E(t_h)$. Finally, the optimal cost $J_3^*(\mathbf{x}_r)$ is, again, a function of the release state $\mathbf{x}_r$.

These three optimization problems are solved by exploiting the direct transcription method, in particular the Matlab implementation of the Drake libraries [20], on second order spline trajectories for $\mathbf{x}$ and $\mathbf{u}$. Other possible approaches could include the use of the flatness property for the MonkeyRotor in order to avoid numerical integration of the system dynamics, or a direct collocation method.

2) Outer loop: the outer loop attempts to determine the optimal release state $\mathbf{x}_r^*$ by solving the following minimization problem

$$\mathbf{x}_r^* = \arg\min_{\mathbf{x}_r}(J_1^*(\mathbf{x}_r) + J_2^*(\mathbf{x}_r) + J_3^*(\mathbf{x}_r)).$$

In this case, we opted for a simple grid search algorithm for finding the optimal $\mathbf{x}_r^*$. Indeed $\mathbf{x}_r$ can be parameterized by the pair $(\boldsymbol{\theta},\, \dot{\boldsymbol{\theta}})$ (four variables) since it must be compatible with the hook constraints (6–7), thus considerably reducing the search space.

B. Cost function

Reasonable choices for the cost functions $J_1(\mathbf{x})$, $J_2(\mathbf{x})$ and $J_3(\mathbf{x})$ could be the execution time or control effort/energy for generating minimum-time or minimum-effort/energy trajectories from $\mathbf{x}_0$ to $\mathbf{x}_f$. Motivated by the recent [21], in this work we however choose to also consider optimality of the *state sensitivity* w.r.t. variations in the system parameters (e.g., mass, inertia, CoM location, propeller characteristics, and so on). Indeed, one can expect some unavoidable level of uncertainty in the various parameters of the MonkeyRotor dynamical and actuation model, and it is then interesting (and useful) to generate an optimal state trajectory $\mathbf{x}^*(t)$ which, by construction, results most insensitive to parametric variations (therefore, its tracking will be facilitated when a parameter is poorly known as explained in [21]).

We here recap, for the reader convenience, the essential machinery for computing the sought state sensitivity. Let then $\boldsymbol{\pi} = [J_B\ J_1\ m_B\ m_1\ L_B\ d_1] \in \mathbb{R}^p$, $p = 6$, be the vector of parameters of interest in our case, and define

$$\boldsymbol{\Pi} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\pi}} \in \mathbb{R}^{n\times p} \tag{14}$$

as the state sensitivity matrix w.r.t. the parameters $\boldsymbol{\pi}$. Although $\boldsymbol{\Pi}$ does not admit, in general, a closed-form expression, one can find an expression for its dynamics as

$$\dot{\boldsymbol{\Pi}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\boldsymbol{\Pi} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\pi}}, \qquad \boldsymbol{\Pi}(t_0) = \mathbf{0}, \tag{15}$$

where $\mathbf{f}$ in our context stands for the hooked or free-flying dynamics depending on the particular phase. It is then possible to numerically integrate (15) over the interval $[t_0,\, t_f]$ for obtaining the behavior of $\boldsymbol{\Pi}(t)$.

By exploiting availability of $\boldsymbol{\Pi}$, we then choose, as a representative case, to minimize a weighted sum of the total execution time $t_f - t_0$ *and* of a norm of the state sensitivity at the hook time $\boldsymbol{\Pi}(t_h)$, with the aim of generating near minimum-time trajectories that are also most insensitive to uncertainties in the MonkeyRotor parameters when approaching the second hook. This is formally obtained by letting

$$J_1 = t_r - t_0, \quad J_2 = t_h - t_r + \gamma\|\boldsymbol{\Pi}(t_h)\|_{\mathbf{w}}, \quad J_3 = t_f - t_h. \tag{16}$$

The gain $\gamma > 0$ is meant to tune the relative weight between the two optimization objectives, and the matrix norm is defined as

$$\|\boldsymbol{\Pi}\|_{\mathbf{w}} = \sum_{i,j} w_{ij}\Pi_{ij}^2 \tag{17}$$

for a set of non-negative weight $\mathbf{w} = [\ldots w_{ij} \ldots]$ whose purpose is to select (and give relative importance to) the desired entries in matrix $\boldsymbol{\Pi}$.

IV. FLIGHT CONTROL

A. Hooked phase

The goal of the control in the hooked phase is to let the MonkeyRotor configuration $\boldsymbol{\theta}$ track the reference optimal trajectory $\boldsymbol{\theta}^*(t)$ generated by the planning algorithm. This can be accomplished by implementing a static feedback linearization of the MonkeyRotor constrained dynamics (3)

$$\mathbf{u} = \mathbf{G}_h^{\dagger}(\boldsymbol{\theta})(\mathbf{M}_h(\boldsymbol{\theta})\boldsymbol{\nu} + \mathbf{g}_h(\boldsymbol{\theta})) + \lambda\mathbf{n}_h \tag{18}$$

where the † operator indicates the usual MoorePenrose pseudoinverse, $\lambda \in \mathbb{R}$ is a scalar gain and

$$\mathbf{n_h} = \begin{bmatrix} 1 \\ -L_1\sin(\theta_1) \\ L_1\sin(\theta_1) \end{bmatrix} \tag{19}$$

is a vector spanning the one-dimensional null-space of matrix $\mathbf{G}_h$ (due to the MonkeyRotor overactuation during the hooked phase).

By plugging (18) into (3), one then obtains the linearized dynamics $\ddot{\boldsymbol{\theta}} = \boldsymbol{\nu}$ which can be stabilized along the reference trajectory $\boldsymbol{\theta}^*(t)$ by choosing

$$\boldsymbol{\nu} = \ddot{\boldsymbol{\theta}}^* + k_d(\dot{\boldsymbol{\theta}}^* - \dot{\boldsymbol{\theta}}) + k_p(\boldsymbol{\theta}^* - \boldsymbol{\theta}) \tag{20}$$

where $k_d > 0$ and $k_p > 0$ are suitable gains.

As well-known, setting $\lambda = 0$ in (18) yields the minimum-norm solution for vector $\mathbf{u}$. However, the null-space term $\lambda \mathbf{n}_h$ can be exploited for accomplishing a secondary objective besides the tracking of $\boldsymbol{\theta}^*(t)$. In our case, we chose to exploit this term for coping, as much as possible, with the actuation constraints (13). This is obtained as follows: by rewriting (18)–(20) as $\mathbf{u} = \mathbf{u}^* + \lambda \mathbf{n}_h$, we seek the optimal value λ^* solving this (linear) minimization problem

$$\begin{aligned} \lambda^* = &\arg\min |\lambda| \\ &\text{s.t.} \quad \underline{\mathbf{u}}_f \leq \mathbf{K}\mathbf{u}^* + \lambda \mathbf{K}\mathbf{n}_h \leq \bar{\mathbf{u}}_f. \end{aligned} \tag{21}$$

If a solution exists, then by setting $\lambda = \lambda^*$ in (18) will guarantee fulfilment of the tracking task and, at the same time, of the actuation constraints with the smallest possible norm for the control input $\mathbf{u}$. In case (21) does not admit a solution, no control action can meet the constraints while realizing the tracking task. In this case vector $\mathbf{u}_f$ is simply saturated[2].

B. Free-flying phase

As explained in Sect. II-C, during free-flight the MonkeyRotor is underactuated but one can still achieve full dynamical linearization of its dynamics by acting on a suitable flat/linearizing output. In short, this is obtained as follows: let $\theta_{1B} = \theta_1 + \theta_B$, define $\mathbf{y}(\mathbf{q}) = [\mathbf{p}_B^T \ \theta_{1B}]^T \in \mathbb{R}^3$ as the flat/linearizing output and let $\mathbf{y}^*(t)$ be the corresponding reference optimal trajectory generated by the trajectory planner of Sect. III. Let also $\bar{\mathbf{u}} = [\ddot{u}_t \ u_r \ \ddot{\tau}]^T$ be the new (extended) input vector, where two integrators have been placed on both the u_t and τ original inputs. The new (extended) state (including the dynamic extensions of the original inputs) is then denoted as $\bar{\mathbf{x}} = [\mathbf{p}_B^T \ \dot{\mathbf{p}}_B^T \ \boldsymbol{\theta}^T \ \dot{\boldsymbol{\theta}}^T \ u_t \ \dot{u}_t \ \tau \ \dot{\tau}]^T \in \mathbb{R}^{12}$. With these settings, one can show (see [9]) that differentiating the flat output $\mathbf{y}$ four times yields

$$\ddddot{\mathbf{y}} = \bar{\mathbf{f}}(\bar{\mathbf{x}}) + \bar{\mathbf{A}}(\bar{\mathbf{x}})\bar{\mathbf{u}} \tag{22}$$

where $\bar{\mathbf{A}}(\bar{\mathbf{x}})$ is a square nonsingular matrix as long as $u_t \neq 0$. System (22) can then be inverted by choosing $\bar{\mathbf{u}} = \bar{\mathbf{A}}(\bar{\mathbf{x}})^{-1}(\bar{\boldsymbol{\nu}} - \bar{\mathbf{f}}(\bar{\mathbf{x}}))$. Tracking of the optimal trajectory $\mathbf{y}^*(t)$ is then obtained by choosing, as usual,

$$\bar{\boldsymbol{\nu}} = \ddddot{\mathbf{y}}^* + k_1(\dddot{\mathbf{y}}^* - \dddot{\mathbf{y}}) + k_2(\ddot{\mathbf{y}}^* - \ddot{\mathbf{y}}) + k_3(\dot{\mathbf{y}}^* - \dot{\mathbf{y}}) + k_4(\mathbf{y}^* - \mathbf{y}) \tag{23}$$

where k_1, k_2, k_3, $k_4 > 0$ are suitable gains.

[2]We note that this case is quite unlikely to occur in practice since the optimal trajectory $\boldsymbol{\theta}^*(t)$ is already compliant "by construction" with the actuation constraint. Any additional control authority needed to recover possible perturbations/disturbances during flight can then be typically accommodated by exploiting the null-space term $\lambda^* \mathbf{n}_h$.

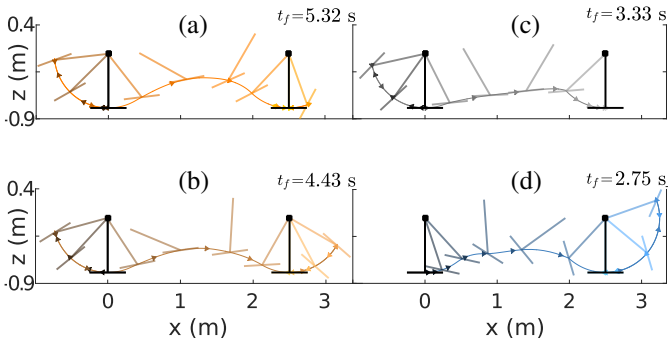


Fig. 4: Time-optimal trajectories of the MonkeyRotor CoM $\mathbf{p}_B(t)$ for the cases of a total thrust/weight ratio of (a) 60%, (b) 70%, (c) 90% and (d) 150%. Note how, depending on the case, a swinging maneuver is produced for either building up energy before flight and/or for quickly losing energy after flight.

Hook-hook distance (m)	d_1 (m)	L_1 (m)	L_B (m)	m_B (kg)	m_1 (kg)	J_B (kg m^2)	J_1 (kg m^2)
2.5	0.3	0.75	0.5	1.3	0.2	0.33	0.027

TABLE I: Values of the parameters used for the MonkeyRotor model.

V. RESULTS

In this section we present a number of simulation results meant to validate the proposed modeling, planning and control strategy for the MonkeyRotor. The first subsection is dedicated to the results of the trajectory planning algorithm, and the second one to the control tracking performance when also considering parameter uncertainty. The video attached to the paper illustrates some of these results.

A. Trajectory planning

We implemented the trajectory planning framework described in Sect. III with the values reported in Table I. We first report the results of only minimizing w.r.t. the execution time (by setting $\gamma = 0$ in (16)), and then we consider the concurrent minimization of the state sensitivity norm (by setting $\gamma = 1$).

1) Minimization w.r.t. execution time: in order to better appreciate the effects of the actuation constraints on the trajectory generation, we considered a total thrust limited to 60%, 70%, 90% and 150% of the total weight (while keeping the same constraints on the other inputs) for testing the MonkeyRotor behavior in different regimes (from low to high control effort modes) and ultimately assessing how the environment interaction can be exploited for performing the desired maneuver. The resulting trajectories are reported in Fig. 4. In all cases, the MonkeyRotor starts at rest while hooked at the left pivot, and ends at rest hooked at the right pivot. One can note how, in the 'low' control effort modes (cases (a)–(c) with thrust less than weight), an initial swing allows for building up the energy needed for reaching the second hook. In particular, in cases (a) and (b) the low thrust/weight ratio causes the trajectory to look approximately ballistic during the free-flying part. On

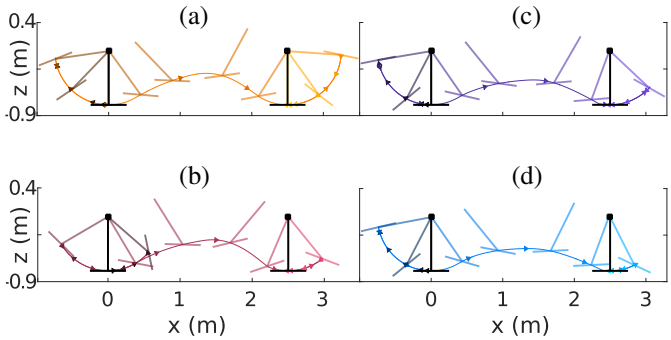


Fig. 5: MonkeyRotor trajectories when minimizing for time *and* state sensitivity at t_h. The considered sensitivities are (a) x and z w.r.t. m_B and m_1, (b) $\mathbf{p}_E$ w.r.t. $\boldsymbol{\pi}$, (c) q w.r.t. $\boldsymbol{\pi}$, (d) q w.r.t. d_1. These sensitivity minimized trajectories are less direct and therefore slower as compared to the time-optimal cases of Fig. 4, but they are also less sensitive to variations in the considered parameters.

States and parameters	$(x,\, z)$ vs. $(m_B,\, m_1)$	$\mathbf{p}_E$ vs. $\boldsymbol{\pi}$	$\mathbf{q}$ vs. $\boldsymbol{\pi}$	$\mathbf{q}$ vs. d_1
Only time-optimal	0.1902	0.1615	0.0376	0.2128
Time- and sensitivity-optimal	0.0827	0.1083	0.0163	0.1157

TABLE II: Comparison of the norm of the state difference at t_h between a time-optimal trajectory, and a time- and sensitivity-optimal trajectory. As expected, when perturbing the parameters, the perturbed state deviates less from the nominal state when executing a time- and sensitivity-optimal trajectory.

the other hand, when more thrust is available (cases (c) and even more (d)), the free-flying phase is much more "direct": however, the breaking phase at the second hook is nevertheless performed by exploiting hook constraint, either thanks to the allowed collision with the second hook in case (c), or by performing a final swing in case (d) (where the thrust exceeds the total weight).

We believe that the representative cases reported in Fig. 4 constitute a good validation of the MonkeyRotor concept, in particular of its switching dynamics which is cleverly leveraged by the trajectory optimization algorithm.

2) Minimization w.r.t. execution time and state sensitivity: focusing on the state sensitivity minimization, several trajectories have been generated while considering the sensitivity of different sets of states and parameters by suitably activating/deactivating the coefficients Π_{ij} via the weighting matrix $\mathbf{W}$ in (17). The thrust limit was always fixed at 70% of the total weight as in the case of Fig. 4(b). The resulting trajectories (and combinations of states/parameters) are reported in the four case studies of Fig. 5. We can notice that the trajectories, although close in shape, present some variations in their characteristics. In particular, the shape of the flying phase is more parabolic in the cases (a) and (b) while a bit flattened in the other cases. Furthermore, the re-hooking state $\mathbf{x}_h$ systematically comes later (i.e., θ_{1B} is closer to 0) than in the corresponding time-optimal trajectory of Fig. 4.

In order to verify the effectiveness of having also optimized w.r.t. the state sensitivity, we performed the following test: we simulated the evolution of the MonkeyRotor states when applying the optimal (open-loop) input $\mathbf{u}^*(t)$ in the nominal non-perturbed case and in the perturbed case (by increasing each considered parameter by 10%), and then evaluated the difference in the selected states at t_h when executing a trajectory only optimized w.r.t. time and when executing a trajectory *also* optimized w.r.t. the state sensitivity at t_h. Table II reports the results: one can note how the norm of the difference between nominal and perturbed states at t_h is always lower in the case of a trajectory also optimized w.r.t. the state sensitivity, thus indicating that (open-loop) execution of this trajectory results intrinsically more robust w.r.t. parametric variations (as expected).

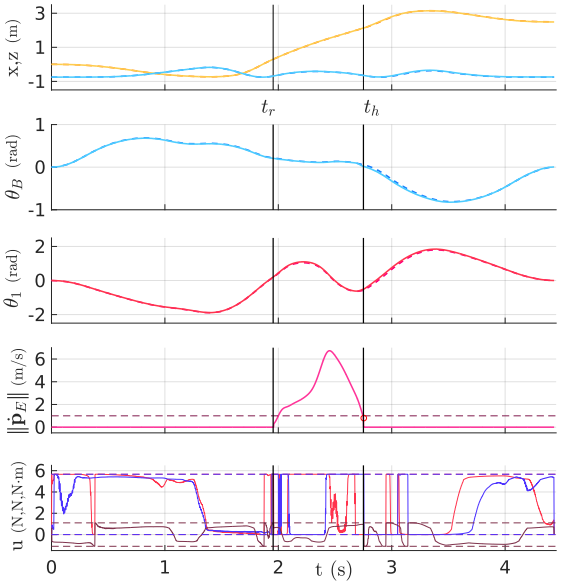


Fig. 6: Behavior of the MonkeyRotor states, inputs and end-effector norm velocity while tracking an optimal trajectory.

B. Trajectory tracking

We now illustrate the performance of the control laws described in Sect. IV in tracking the reference optimal trajectories generated by the planning algorithm. To this end, we implemented the (switching) MonkeyRotor dynamics and the control laws in Simulink, and employed, for the sake of visualization, the V-REP[3] simulation environment.

As a representative case study, Fig. 6 reports the tracking performance for a time-optimal trajectory obtained for a thrust limit of 70% of the weight, and with random perturbations to the parameters of $\pm 5\%$ their nominal values. It is worth noting how, despite the parametric variations, the control inputs always remain within their bounds (represented by dashed horizontal lines), and how the norm of the end-effector velocity $\|\dot{\mathbf{p}}_E\|$ falls below the threshold $\mathbf{v}_{max}$ at t_h as planned. The performance in tracking the reference optimal state (dashed lines) is also quite satisfactory.

[3] http://www.coppeliarobotics.com/

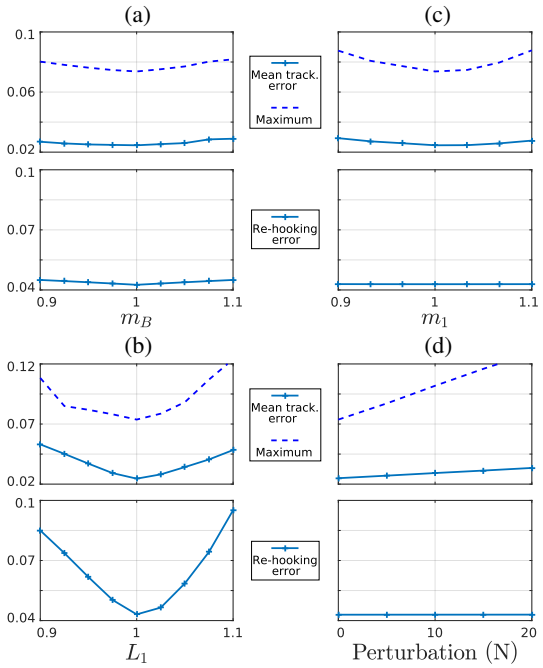


Fig. 7: Performance of the proposed planning/control framework under parametric variations (Figs. (a–c)) and external disturbances (Fig. (d)): in each figure the top plot reports the mean (solid line) and max (dashed line) values of the tracking error norm during the trajectory execution, and the bottom plot reports the value of the re-hooking error $\|\mathbf{p}_E(t_h) - \mathbf{p}_E^*\|$. The parameters m_B, m_1 and L_1 are varied from 90% to 110% of their nominal value, while the external perturbation (wind gust) has amplitude ranging from 0 N to 20 N.

As an additional validation, we also ran a statistical analysis of the overall tracking error (averaged over the whole trajectory) and end-effector re-hooking error at t_h on the trajectory of Fig. 5-(b) (whose state sensitivity is optimized against all the considered parameters $\boldsymbol{\pi}$) when varying some parameters of interest from 90% to 110% of their nominal value. Figure 7 reports the results of this analysis: in Figs. 7(a–c) we consider the variation of m_B, L_1 and m_1, while Fig. 7(d) considers the presence of an external disturbance, a wind gust of varying amplitude with duration $0.2\ s$ and applied during the free-flying case along the negative $\mathbf{x}_W$ axis. One can note how the performance (especially in terms of the re-hooking error) remains quite satisfactory despite the parameter variations and/or external disturbance, thus showing that the proposed combination of the state sensitivity minimization (planning stage) and the closed-loop tracking controller control are able to yield a successful MonkeyRotor maneuver also in more realistic conditions[4].

[4]We note that the re-hooking error remains almost constant except when changing the length of the arm, which is expected since the parameter L_1 does not affect the free-flying dynamics and thus is not taken into account in the sensitivity minimization.

VI. CONCLUSIONS

In this work we have introduced the concept of aerial physical locomotion by considering the MonkeyRotor system — a quadrotor UAV equipped with a 1-DOF arm able to hook at some pivot points and to exploit these contacts for enhancing its maneuvering possibilities. To this end, a suitable dynamical model for both the hooked and free-flying phases has been presented, together with an optimization framework for generating optimal motion plans under constrained actuation. Two control laws for the two hooked and free-flying phases have also been proposed, and the whole concept has been successfully validated in a number of simulations.

In the future we plan to build a physical prototype of the MonkeyRotor for assessing the proposed framework in real conditions. We are also considering the possibility of executing more complex maneuvers (e.g., jumping to multiple branches in sequence), as well as the use of online replanning strategies for continuously refining the initial optimal trajectory during motion.

REFERENCES

[1] H. Nguyen and D. Lee, "Hybrid force/motion control and internal dynamics of quadrotors for tool operation," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 3458–3464.

[2] B. Yüksel, C. Secchi, H. H. Bülthoff, and A. Franchi, "Reshaping the physical properties of a quadrotor through IDA-PBC and its application to aerial physical interaction," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May. 2014, pp. 6258–6265.

[3] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," in *14th Int. Symp. on Robotics Research*, Lucerne, Switzerland, Sep. 2009.

[4] M. Tognon and A. Franchi, "Dynamics, control, and estimation for aerial robots tethered by cables or bars," *IEEE Trans. on Robotics*, vol. 33, no. 4, pp. 834–845, 2017.

[5] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct 2012, pp. 3532–3537.

[6] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, and A. Franchi, "6D physical interaction with a fully actuated aerial robot," in *2017 IEEE Int. Conf. on Robotics and Automation*, Singapore, May 2017.

[7] G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Experiments on coordinated motion of aerial robotic manipulators," in *2016 IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, May 2016, pp. 1224–1229.

[8] M. Tognon, B. Yüksel, G. Buondonno, and A. Franchi, "Dynamic decentralized control for protocentric aerial manipulators," in *2017 IEEE Int. Conf. on Robotics and Automation*, Singapore, May 2017.

[9] B. Yüksel, N. Staub, and A. Franchi, "Aerial robots with rigid/elastic-joint arms: Single-joint controllability study and preliminary experiments," in *2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Daejeon, South Korea, Oct. 2016, pp. 1667–1672.

[10] D. Falanga, E. Mueggler, M. Faessler, and D. Scaramuzza, "Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision," in *2017 IEEE Int. Conf. on Robotics and Automation*, Singapore, 2016.

[11] H. Tsukagoshi, M. Watanabe, T. Hamada, D. Ashlih, and R. Iizuka, "Aerial manipulator with perching and door-opening capability," in *2015 IEEE Int. Conf. on Robotics and Automation*, May 2015, pp. 4663–4668.

[12] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, "Vision-based autonomous quadrotor landing on a moving platform," in *IEEE Int. Symp. on Safety, Security and Rescue Robotics*, May 2017.
[13] L. Sentis, "Compliant control of whole-body multi-contact behaviors in humanoid robots," 2009.
[14] S. Brossette, J. Vaillant, F. Keith, A. Escande, and A. Kheddar, "Point-cloud multi-contact planning for humanoids: Preliminary results," in *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Nov 2013, pp. 19–24.
[15] M. Hehn, R. Ritz, and R. DAndrea, "Performance benchmarking of quadrotor systems using time-optimal control," *Autonomous Robots*, vol. 33, no. 1–2, pp. 69–88, 2012.
[16] J. Thomas, G. Loianno, J. Polin, K. Sreenath, and V. Kumar, "Toward autonomous avian-inspired grasping for micro aerial vehicles," *Bioinspir. Biomim.*, vol. 9, no. 2, 2014.
[17] M. M. Nicotra, R. Naldi, and E. Garone, "Nonlinear control of a tethered uav: The taut cable case," *Automatica*, vol. 78, pp. 174–184, 2017.
[18] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.
[19] B. V. Mirtich, "Impulse-based dynamic simulation of rigid body systems," Ph.D. dissertation, 1996, aAI9723116.
[20] R. Tedrake and the Drake Development Team, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2016. [Online]. Available: http://drake.mit.edu
[21] A. Ansari and T. Murphey, "Minimum sensitivity control for planning with parametric and hybrid uncertainty," *Int. J. Rob. Res.*, vol. 35, no. 7, pp. 823–839, June 2016. [Online]. Available: http://dx.doi.org/10.1177/0278364915600536