Quadrocopter Pole Acrobatics

Dario Brescianini, Markus Hehn, and Raffaello D'Andrea

***Abstract*— We present the design of a system that allows quadrocopters to balance an inverted pendulum, throw it into the air, and catch and balance it again on a second vehicle. Based on first principles models, a launch condition for the pole is derived and used to design an optimal trajectory to throw the pole towards a second quadrocopter. An optimal catching instant is derived and the corresponding position is predicted by simulating the current position and velocity estimates forward in time. An algorithm is introduced that generates a trajectory for moving the catching vehicle to the predicted catching point in real time. By evaluating the pole state after the impact, an adaptation strategy adapts the catch maneuver such that the pole rotates into the upright equilibrium by itself. Experimental results demonstrate the performance of the system.**

I. Introduction

Due to their high maneuverability and their simple mechanics, quadrocopters have become a popular research subject in recent years. Most early research has been focussed on near-hover operation using simplified linear models (see [1], and references therein). More recently, a growing number of researchers have been working on exploiting the full dynamic capabilities of quadrocopters. Some of the more complex and demanding tasks that have been tackled with quadrocopters in the last few years include ball catching [2], aggressive flight maneuvers through narrow gaps and perching [3], high-speed flips [4], and cooperative tasks such as load carrying [5] or ball throwing [6].

The work presented herein is inspired by two challenging dynamic tasks previously accomplished with quadrocopters: balancing an inverted pendulum [7], and ball juggling [8]. The aim is to balance a pole on a quadrocopter, launch it off the vehicle into the air, and then catch and re-balance it with a second quadrocopter (see Fig. 1).

Stabilizing a pole about its upright equilibrium position is a well-established problem in feedback control. It has been studied for several decades and many different setups of the inverted pendulum have been successfully tried, such as the pole-on-cart or the Furuta pendulum (see [9], and references therein). However, throwing the pole into the air, catching and re-balancing it has - to the best of our knowledge - not been done before. Moreover, it demonstrates the recent progress made in quadrocopter research in performing agile maneuvers with high precision and can serve as a testbed to benchmark trajectory generation algorithms, control strategies and adaptation methods.

This paper demonstrates a system design based on first principles models and recent advances in optimal control, estimation and real-time trajectory generation. We combine optimal control methods to design a nominal throw and catch maneuver off-line, estimation techniques for hybrid systems, real-time trajectory generation as a means to react instantaneously, and adaptation strategies to compensate for systematic errors.

The authors are with the Institute for Dynamic Systems and Control, ETH Zurich, Switzerland. {bdario, hehnm, rdandrea}@ethz.ch

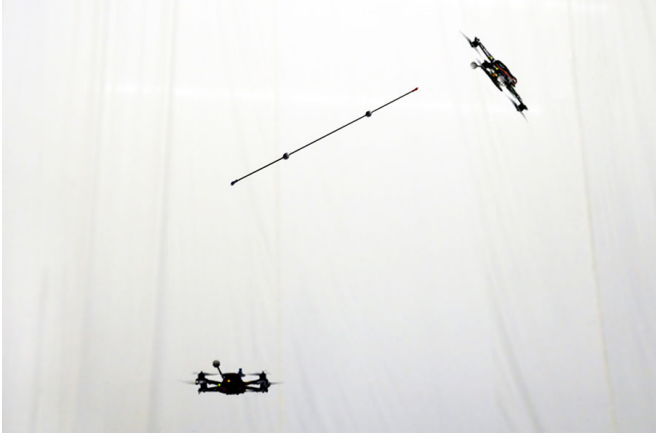


Fig. 1. Illustration of two quadrocopters passing a pole between them. The right quadrocopter throws the pole towards the left vehicle, which tries to catch and re-balance it.

The remainder of this paper is organized as follows: In Section II, the dynamics of the quadrocopter-pole system are derived. In Section III, the nominal throwing trajectory and optimal catching instant are derived. In Section IV, the state estimator is introduced. In Section V, the catch maneuver is presented and the necessary quadrocopter inputs are computed. An adaptation strategy is presented in Section VI to improve the system's performance over time. The experimental setup is explained in Section VII and the results are discussed subsequently. Finally, the paper is concluded in Section VIII, including an outlook to future work.

II. System Dynamics

In this section, we derive the dynamics of the quadrocopter-pole system for two cases: 1) where the pole flies freely through the air, and 2) where the pole has physical contact with the quadrocopter. In the former case, the dynamics of the pole and the quadrocopter are assumed to be independent. In the latter case, the quadrocopter and pole interact through their contact force. Given that the quadrocopter's inertia is an order of magnitude larger than the pole's inertia, it is assumed that the pole has no impact on the quadrocopter dynamics. The quadrocopter dynamics are therefore modelled independent of the pole for both cases.

If not explicitly mentioned otherwise, all vectors will be expressed in a stationary inertial frame I. For the ease of

notation, vectors are expressed as n-tuples $(x_1, x_2, \dots)$, with dimension and stacking clear from context.

A. Quadrocopter

The quadrocopter is modelled as a rigid body with its control inputs being the angular rates $\Omega_q = (\Omega_{q,x}, \Omega_{q,y}, \Omega_{q,z})$ about the body-fixed coordinate axes and a mass-normalized collective thrust a_q along the vehicle's z-axis (Fig. 2). The angular rate inputs Ω_q are tracked on-board by a high-bandwidth controller using gyroscope feedback and the rotational dynamics can thus be neglected [4]. The control inputs are constrained to

$$0 < a_{q,min} \leq a_q \leq a_{q,max} \tag{1}$$

and

$$|\Omega_{q,i}| \leq \Omega_{q,max}, \quad i \in \{x, y, z\}. \tag{2}$$

The translational degrees of freedom are described by the position of the vehicle's center of mass $p_q = (x_q, y_q, z_q)$, and the rotational degrees of freedom are parametrized using the zyx-Euler angles [10] (ψ, θ, ϕ). The rotation matrix that maps a vector from the inertial frame I into the quadrocopter-fixed coordinate frame B is computed by rotating first about the inertial z-axis by yaw angle ψ, thereafter about the new y-axis by pitch angle θ, and finally about the new x-axis by roll angle ϕ:

$${}^{B}_{I}R(\psi, \theta, \phi) = R_x(\phi) R_y(\theta) R_z(\psi), \tag{3}$$

with $R_{(\cdot)}$ being a rotation matrix about the corresponding axis. The translational acceleration of the vehicle is then fully determined by the quadrocopter attitude and the collective thrust input:

$$\ddot{p}_q = {}^{I}_{B}R(\psi, \theta, \phi) \begin{bmatrix} 0 \\ 0 \\ a_q \end{bmatrix} - g, \tag{4}$$

where ${}^{I}_{B}R$ is the inverse of ${}^{B}_{I}R$ and g denotes gravity. The rotational dynamics are obtained by converting the angular rate inputs into Euler rates [10]:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}^{-1} \begin{bmatrix} \Omega_{q,x} \\ \Omega_{q,y} \\ \Omega_{q,z} \end{bmatrix}. \tag{5}$$

We define the quadrocopter state s_q and the control input vector u_q to be

$$s_q = (p_q, \dot{p}_q, \phi, \theta, \psi) \tag{6}$$

$$u_q = (a_q, \Omega_q). \tag{7}$$

The first-order differential equation describing the evolution of the quadrocopter state is then

$$\dot{s}_q = f_q(s_q, u_q), \tag{8}$$

where f_q consists of the nonlinear equations (4) and (5).

B. Pole

The pole considered in this paper (see Fig. 2) is modelled as a thin rod of length $2L$, mass m_p, and an inertia tensor $\bar{\Theta}$ with respect to its center of mass expressed in the pole-fixed frame K:

$$\bar{\Theta} = \begin{bmatrix} \frac{1}{3}m_p L^2 & 0 & 0 \\ 0 & \frac{1}{3}m_p L^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{9}$$

Some of the equations and derivations of the pole dynamics are too long to be presented herein or are left out for the sake of readability. However, the full derivations of all equations are available online at `http://bit.ly/2pN1O2q`.

1) Free flight: The pole mass center position is represented by $p_p = (x_p, y_p, z_p)$ and its reduced attitude [11] is expressed using the unit vector n pointing along the pole z-axis. During flight, aerodynamic drag and gravity act upon the pole. Due to the small cross section of the pole along its z-axis and the comparatively large cross section orthogonal to it, the pole's drag properties depend heavily on its orientation. Consequently, the drag force is split into two components: a force in the direction of the pole z-axis and a force in its xy-plane. The drag forces themselves are modelled proportionally to the speed squared:

$$F_{drag,z} = -c_z \int_{-L}^{L} \|\dot{p}_z(\xi)\| \dot{p}_z(\xi) d\xi \tag{10}$$

$$F_{drag,xy} = -c_{xy} \int_{-L}^{L} \|\dot{p}_{xy}(\xi)\| \dot{p}_{xy}(\xi) d\xi, \tag{11}$$

where $c_{(\cdot)}$ is the drag coefficient and $\dot{p}(\xi)$ is the velocity of a point on the pole at a distance of ξ from the center. Given the pole center velocity $\dot{p}_p$ and its angular rate Ω_p, $\dot{p}(\xi)$ is given by:

$$\dot{p}(\xi) = \dot{p}_p + \Omega_p \times (\xi n). \tag{12}$$

The pole-fixed z- and xy-components of the velocity can then be calculated:

$$\dot{p}_z(\xi) = \left(\dot{p}(\xi)^{\mathrm{T}} n\right) n \tag{13}$$

$$\dot{p}_{xy}(\xi) = \dot{p}(\xi) - \dot{p}_z(\xi). \tag{14}$$

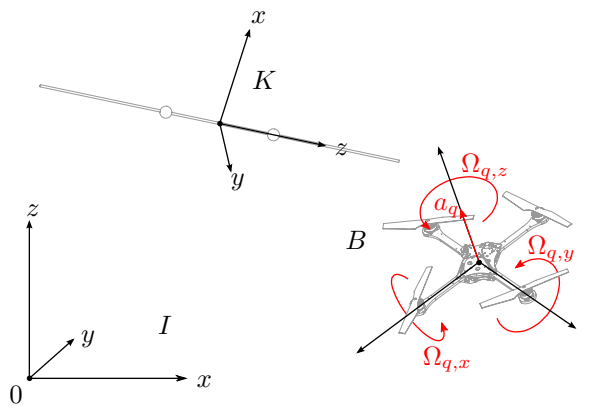


Fig. 2. Inertial reference frame I and the quadrocopter and pole with their body-fixed coordinate frames B and K, respectively. The quadrocopter control inputs are a mass-normalized thrust a_q along the body-fixed z-axis and the angular rates $(\Omega_{q,x}, \Omega_{q,y}, \Omega_{q,z})$.

The torque caused by the aerodynamic drag follows analogously to (11):

$$M_{drag} = -c_{rot} \int_{-L}^{L} (\xi n) \times \left(\|\dot{p}(\xi)\| \dot{p}(\xi) \right) d\xi. \tag{15}$$

The Newton-Euler equation for the pole in free flight yields

$$m_p \ddot{p}_p = F_{drag,z} + F_{drag,xy} - m_p g \tag{16}$$

$$\left({}^{I}_{K}R \bar{\Theta} {}^{K}_{I}R \right) \dot{\Omega}_p = M_{drag}, \tag{17}$$

where ${}^{I}_{K}R$ is the rotation matrix from the pole-fixed frame K to the inertial frame I. In the online appendix [12], we show that the pole's moment of inertia in the inertial frame I can be written as

$${}^{I}_{K}R \bar{\Theta} {}^{K}_{I}R = \Theta \left(\mathbf{1} - nn^{\mathrm{T}} \right), \tag{18}$$

with $\Theta = \frac{1}{3} m_p L^2$. It follows from (15) that no torque is induced about the pole z-axis, and hence it is assumed that the angular rate about this axis remains constant, i.e. the angular acceleration along the pole-fixed z-axis is zero. In this case, the rotational dynamics (17,18) simplify to

$$\dot{\Omega}_p = \frac{1}{\Theta} M_{drag}. \tag{19}$$

The dynamics for the pole in free flight can be written as

$$\dot{s}_p = f_p(s_p) \tag{20}$$

with the pole state s_p being defined as

$$s_p = (p_p, \dot{p}_p, n, \Omega_p) \tag{21}$$

and f_p containing (16), (19) and the kinematic relation

$$\dot{n} = \Omega_p \times n. \tag{22}$$

2) On quadrocopter: In the case where the pole is in contact with a quadrocopter, it is assumed that the pole is rigidly attached to the quadrocopter's mass center. The pole position then depends on the quadrocopter:

$$p_p = p_q + Ln, \tag{23}$$

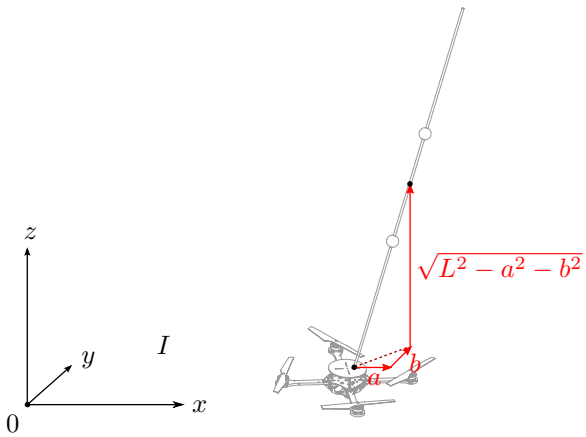


Fig. 3. Quadrocopter-pole system with the pole balanced on the quadrocopter. It is assumed that the pole is attached to the quadrocopter's center of gravity.

where we assume, without loss of generality, that n points from the quadrocopter towards the pole mass center. Subsequently, the pole attitude is parametrized by the deflection of the pole center relative to its supporting point (see Fig. 3):

$$Ln = \left(a, b, \sqrt{L^2 - a^2 - b^2} \right), \tag{24}$$

where a denotes the deflection in the inertial x-direction and b in the y-direction, respectively. The Lagrangian of the pole can be shown to be (see online appendix [12])

$$\mathscr{L} = \frac{1}{2} m_p \dot{p}_p^{\mathrm{T}} \dot{p}_p + \frac{1}{2} \Theta \widetilde{\Omega}_p^{\mathrm{T}} \widetilde{\Omega}_p - m_p g^{\mathrm{T}} p_p \tag{25}$$

with

$$\dot{p}_p = \left(\dot{x}_q + \dot{a}, \dot{y}_q + \dot{b}, \dot{z}_q - \frac{a\dot{a} + b\dot{b}}{\sqrt{L^2 - a^2 - b^2}} \right) \tag{26}$$

$$\widetilde{\Omega}_p = n \times \dot{n}. \tag{27}$$

Note that, by computing the angular rate according to (27), the rotational rate about the pole z-axis is not recovered. However, since the pole has no moment of inertia about this axis, rotating about it does not add kinetic energy to the system and therefore has no influence on the Lagrangian. The nonlinear dynamics are derived applying Lagrangian dynamics:

$$\frac{d}{dt} \left(\frac{\partial \mathscr{L}}{\partial \dot{a}} \right) - \frac{\partial \mathscr{L}}{\partial a} = 0 \tag{28}$$

$$\frac{d}{dt} \left(\frac{\partial \mathscr{L}}{\partial \dot{b}} \right) - \frac{\partial \mathscr{L}}{\partial b} = 0 \tag{29}$$

and are presented in more detail in the online appendix [12]. The dynamics of the combined quadrocopter-pole system can be described by

$$\dot{s}_{qp} = f_{qp}(s_{qp}, u_q), \tag{30}$$

where s_{qp} is the combined quadrocopter-pole state

$$s_{qp} = \left(s_q, a, b, \dot{a}, \dot{b} \right) \tag{31}$$

and f_{qp} contains the quadrocopter dynamics f_q (8) and the pole dynamics derived from (28,29).

III. Nominal Maneuver Design

In this section, we first introduce the nominal throw maneuver and then identify the optimal catching instant.

A. Throw Maneuver

The throwing of the pole is designed to take place in the vertical xz-plane. We therefore use a simplified two-dimensional quadrocopter-pole model with the quadrocopter Euler angles $\phi = \psi = 0$ and the pole deflection $b = 0$ (see Fig. 4). The pole attitude is parametrized by the tilt angle α:

$$\alpha = \sin^{-1} \left(\frac{a}{L} \right). \tag{32}$$

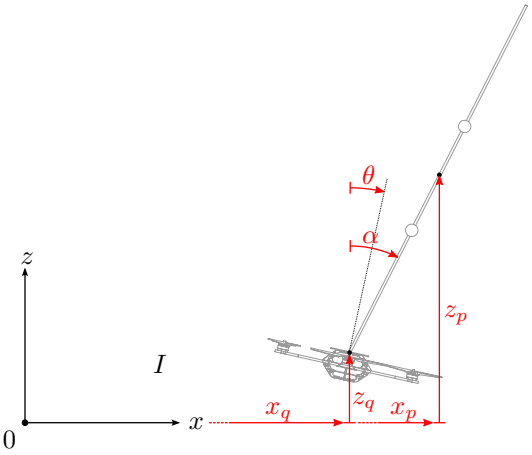


Fig. 4. A simplified two-dimensional model of the quadrocopter-pole system in the xz-plane. The pole attitude is parametrized by the tilt angle α and the quadrocopter attitude is fully described by the pitch angle θ.

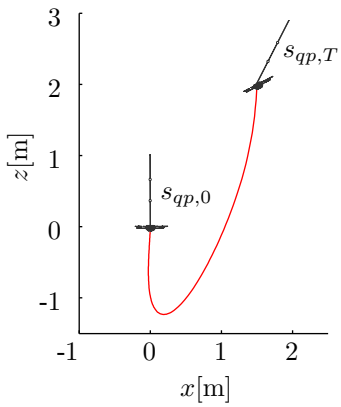


Fig. 5. Throwing trajectory for a maneuver length of $T = 2.5\,\mathrm{s}$. The final constraint $s_{qp,T}$ consists of position $(x_q, z_q) = (1.5\,\mathrm{m}, 2.0\,\mathrm{m})$, velocity $(\dot{x}_q, \dot{z}_q) = (0.3\,\mathrm{m\,s^{-1}}, 5.5\,\mathrm{m\,s^{-1}})$, attitudes $(\alpha, \theta) = (0.48\,\mathrm{rad}, -0.48\,\mathrm{rad})$, and angular rate $\dot{\alpha} = 2.5\,\mathrm{rad\,s^{-1}}$, and with the final control input $u_{q,T}$ being $(a_q, \Omega_{q,y}) = (4.48\,\mathrm{m\,s^{-2}}, 2.5\,\mathrm{rad\,s^{-1}})$.

The normal force between the quadrocopter and pole is (see online appendix [12])

$$F_N(s_{qp}, u_q) = m_p \left(a_q - \frac{3}{4} a_q \sin^2(\alpha - \theta) - L\dot{\alpha}^2 \cos(\alpha - \theta) \right) \tag{33}$$

For the pendulum to be launched off the vehicle, $F_N < 0$ is required. We note that, as discussed in Section II, $a_q > 0$. The pole can therefore not be thrown vertically, but must be accelerated to a sufficiently high rotational rate $\dot{\alpha}$.

We generate a trajectory to maneuver the quadrocopter from a stationary starting point to a state at which the pole leaves the quadrocopter, using optimal control methods:

$$\begin{aligned} \text{minimize} \quad & \int_0^T 1.5\alpha(t)^2 + (a_q(t) - g)^2 + 8\Omega_{q,y}(t)^2 dt \\ \text{subject to} \quad & \dot{s}_{qp}(t) = f_{qp}(s_{qp}(t), u_q(t)) \\ & u_{q,min} \leq u_q(t) \leq u_{q,max} \\ & 0 \leq F_N(s_{qp}(t), u_q(t)) \\ & s_{qp}(t = 0) = 0 \\ & s_{qp}(t = T) = s_{qp,T}, \end{aligned} \tag{34}$$

The initial constraint represents the quadrocopter hovering at the origin with the pole balanced in the equilibrium position. The final state $s_{qp,T}$ is a design parameter, that, in combination with a control input $u_{q,T}$, must satisfy $F_N(s_{qp,T}, u_{q,T}) = 0$. The cost function is designed so as to yield a smooth throwing trajectory. A solution to the optimization problem (34) for a trajectory duration $T = 2.5\,\mathrm{s}$ is shown in Fig. 5.

B. Catching Instant

Ideally, the pole rotates to the upright equilibrium position by itself after it is caught by the quadrocopter. Since the complete throw maneuver is performed in the xz-plane, the two-dimensional model is considered again to compute this optimal catching instant. For reasons that will be stated in Section V, the pole is caught with the quadrocopter at rest. Large forces act upon the pole when it hits the quadrocopter, causing the pole velocity and angular rate to jump. Assuming again that the effects of the impact on the quadrocopter are negligible due to the much larger inertia, the pole dynamics at impact ($x_p = x_q + L\sin\alpha$, $z_p = z_q + L\cos\alpha$) can be written as

$$m_p(\dot{x}_p^+ - \dot{x}_p^-) = \Lambda_T \tag{35}$$
$$m_p(\dot{z}_p^+ - \dot{z}_p^-) = \Lambda_N \tag{36}$$
$$\Theta(\dot{\alpha}^+ - \dot{\alpha}^-) = -\Lambda_T L\cos\alpha + \Lambda_N L\sin\alpha, \tag{37}$$

where Λ_N and Λ_T are the impulses in normal and tangential direction to the surface, respectively. The superscripts $-$ and $+$ indicate values before and after the impact, respectively. We assume a totally inelastic impact with large friction, meaning that the velocities of the pole and the quadrocopter at the contact point are equal after the impact [13]. With the quadrocopter at rest, this is expressed by

$$\dot{x}_p^+ - \dot{\alpha}^+ L\cos\alpha = 0 \tag{38}$$
$$\dot{z}_p^+ + \dot{\alpha}^+ L\sin\alpha = 0. \tag{39}$$

As a result, the pole will stick to the quadrocopter after the impact and only rotate about the contact point. Inserting conditions (38,39) into the impact dynamics (35-37), the angular rate of the pole after the impact can be calculated to be

$$\dot{\alpha}^+ = \frac{1}{4}\dot{\alpha}^- + \frac{3}{4L}\left(\dot{x}_p^- \cos\alpha - \dot{z}_p^- \sin\alpha\right). \tag{40}$$

The angular rate necessary for the pole to rotate into the equilibrium is computed by applying the law of energy conservation:

$$m_p g L = m_p g L\cos\alpha + \frac{1}{2} m_p (L\dot{\alpha}^+)^2 + \frac{1}{2}\Theta(\dot{\alpha}^+)^2. \tag{41}$$

The left hand side of (41) represents the potential energy of the upright equilibrium and the right hand side represents the total energy of the pole after the impact, and is comprised of a potential and kinetic part. Note that with a pole energy

according to (41), the pole will converge to the unstable equilibrium in infinite time. Inserting (40) into (41) finally yields the condition for finding the optimal catching instant:

$$\dot{\alpha}^{-}\mathrm{sgn}\alpha - \frac{3}{L}\left(\dot{x}_p^{-}\cos\alpha - \dot{z}_p^{-}\sin\alpha\right) + \sqrt{\frac{24g\left(1-\cos\alpha\right)}{L}} = 0. \tag{42}$$

The height of the lower pole end, where (42) is satisfied for the first time, is defined to be the nominal catching height.

This completes the nominal maneuver design. The throwing is defined by the solution of the optimization problem (34). The pole state then evolves according to (20) until the optimal catching instant (42). By design, the pole rotates into the upright equilibrium after the impact.

IV. Estimation

In this section, we introduce the estimator used to estimate the pole's state. Because the quadrocopter dynamics are modelled independent of the pole, the quadrocopter state is estimated separately (using a Luenberger observer, e.g. see [14]) and is assumed to be known.

The dynamics of the pole depend on whether it is in free flight or in contact with a quadrocopter. To capture the switching nature of the dynamics, a state observer for hybrid systems is implemented [15]. The observer consists of multiple continuous-time extended Kalman filters with discrete-time measurements, each matched to a specific mode: free flight according to the dynamics (20), and in contact with either one of the quadrocopters according to the dynamics (30). Each estimator has its own state and covariance estimate. A motion capture system provides the estimators with pose information of the quadrocopters and the pole. For each estimator, the ratio of the innovation to the measurement covariance is computed. Depending on these error measures, the most likely mode is chosen and the corresponding estimate is taken as current belief of the state. This state estimate is then fed back to the estimators running in the other modes to correct their states.

V. Real-time trajectory generation for the catch maneuver

As soon as the pole is launched off the throwing vehicle, the system predicts the catching position (i.e. the position where the pole crosses the nominal catching height) and the duration until impact T_{xy}. A catch maneuver, ending at the catching position, is then planned such that the catching vehicle is nominally at rest at the time of impact. This makes the system resistant to small timing errors by permitting the vehicle to wait at the impact location. The catching trajectory is constantly updated, thereby accounting for the improvement in the prediction accuracy as the prediction horizon decreases. When the impact of the pole on the catching vehicle is detected, the feedback control law for the catching quadrocopter is switched to a pole balancing controller.

A. xy-Plane

The horizontal correction maneuver begins with the first prediction of the catching point. A minimum snap trajectory has been chosen to guide the quadrocopter to the catching position because it yields continuous control inputs [16], allowing good tracking under feedback control. With the state vector for each x and y being defined as $s = (q, \dot{q}, \ddot{q}, \dddot{q})$, where q denotes the x- or y-position, respectively, the dynamics are given by a quadruple integrator:

$$\dot{s} = f(s,u) = (\dot{q}, \ddot{q}, \dddot{q}, u), \tag{43}$$

with the snap being the control input u. The trajectory in the x-direction, starting from rest at $x = x_0$ and ending with hovering at the target position x_T, is obtained by solving

$$\begin{array}{ll} \text{minimize} & \int_0^T u(t)^2 dt \\ \text{subject to} & \dot{s}(t) = f(s(t), u(t)) \\ & s(t=0) = (x_0, 0, 0, 0) \\ & s(t=T_{xy}) = (x_T, 0, 0, 0). \end{array} \tag{44}$$

The optimization problem (44) has the analytical solution [17]

$$u(t) = \frac{840}{T_{xy}^7}(x_T - x_0)(-20t^3 + 30t^2 T_{xy} - 12tT_{xy}^2 + T_{xy}^3). \tag{45}$$

The trajectory in the y-direction is computed likewise by replacing x_0 and x_T with y_0 and y_T, respectively. The mass-normalized force that the quadrocopter must produce to follow a three-dimensional path is

$$f = \begin{bmatrix} \ddot{x}_q \\ \ddot{y}_q \\ \ddot{z}_q \end{bmatrix} + g. \tag{46}$$

Hence, using (4), the nominal thrust input to the quadrocopter is $a_q = \|f\|$. The necessary angular rates for a given, constant yaw angle ψ, can be shown to be [18]

$$\begin{bmatrix} \Omega_y \\ -\Omega_x \\ 0 \end{bmatrix} = {}^{B}_{I}R(\psi,\theta,\phi)\left(\frac{\nu}{\|f\|} - \frac{ff^{\mathrm{T}}\nu}{\|f\|^3}\right), \tag{47}$$

where ν denotes the jerk: $\nu = (\dddot{x}_q, \dddot{y}_q, \dddot{z}_q)$. The pitch and roll angle θ and ϕ, required to compute ${}^{B}_{I}R$, can be calculated using (4).

B. z-Direction

It remains to design a vertical motion. As can be seen from (4) and (47), the magnitude of the rotational control inputs and the roll and pitch angles required to follow the trajectory (45) reduce with increasing values of f. We therefore seek to find a trajectory in which the quadrocopter produces a high thrust during the correction time T_{xy} and ends at rest at the catching height.

We use the solution to a minimax acceleration problem as presented in [6] to move the quadrocopter from a starting height z_0 to the catching height z_T. The resulting acceleration profile is shown in Fig. 6.

The free parameters of the maneuver are chosen to be k_{max} (the maximum jerk), a_{max} (the maximum acceleration), and the duration of the high-thrust phase $[t_3, t_5]$.

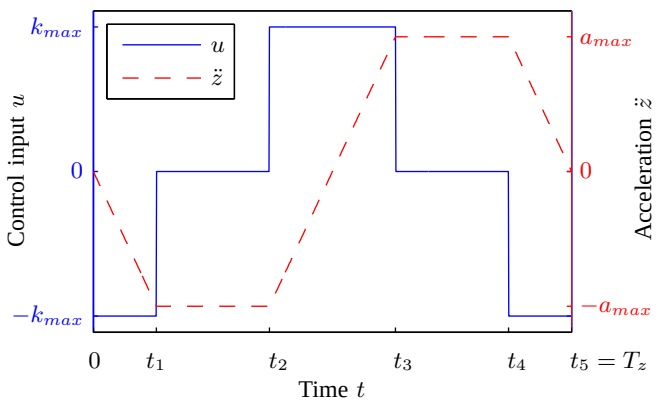


Fig. 6. The jerk input u and the acceleration $\ddot{z}$ that solve the minimax acceleration problem. The solution consists of 5 intervals of constant jerk. The solution corresponds to a quadrocopter motion where the quadrocopter starts at hover at z_0, then drops down and increases its thrust in order to come to rest at a height of z_T.

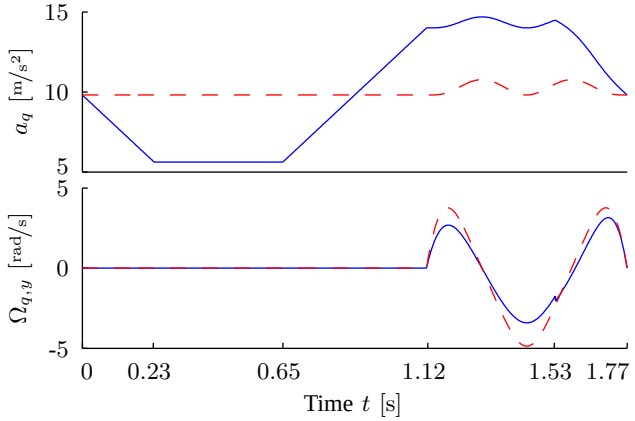


Fig. 7. Control inputs a_q (top) and $\Omega_{q,y}$ (bottom) for a catch maneuver with $T_{xy} = 0.65\,\mathrm{s}$, initial horizontal position $(x_0, y_0) = (0\,\mathrm{m}, 0\,\mathrm{m})$, and final position $(x_T, y_T) = (0.25\,\mathrm{m}, 0\,\mathrm{m})$. The solid lines represent the control inputs when the minimax acceleration maneuver ($a_{max} = 4.19\,\mathrm{m\,s^{-2}}$, $k_{max} = 18\,\mathrm{m\,s^{-3}}$) in the z-direction is flown simultaneously, whereas the dashed lines correspond to a correction maneuver at constant height.

Because we want to fly the xy-correction maneuver during the period with high thrust, T_{xy} is chosen to be identical to the interval length $[t_3, t_5]$. As a result, it can be shown that the maneuver in the z-direction lasts

$$T_z = 2\left(\frac{a_{max}}{k_{max}} + T_{xy}\right) \tag{48}$$

and the initial height is given by

$$z_0 = z_T + \frac{a_{max}^2 T_{xy} + a_{max} k_{max} T_{xy}^2}{k_{max}}. \tag{49}$$

Fig. 7 shows the control inputs for a typical catch maneuver and compares them to the inputs if the quadrocopter remains at the catching height the entire time.

VI. Adaptation

The task of throwing and catching is repetitive and thus offers opportunities to use data from past throwing and catching attempts to eliminate systematic errors. To improve the system's performance in future iterations, adaptation strategies are applied for two key events:

A. Catching Height and Position

After each throw, the actual optimal catching instant according to (42) is evaluated. The catching height is then updated with this information using an iterative mean scheme. This correction accounts for when the throws do not correspond exactly to the nominal throw maneuver, and allows the catch to occur near the optimal height.

To account for horizontal deviations from the nominal throw, the xy-catching position is updated analogously. The adapted horizontal catching position is used as the waiting point of the catching vehicle, minimizing the distance required to travel to catch the pole.

B. Quadrocopter Offset

A second correction term is introduced for the catching bias, defined as the offset of the catching vehicle relative to the predicted catching position at the time of impact. Even though the catching trajectory is different for each catch, the catching quadrocopter repeatedly shows similar position errors when at rest after the catching maneuver. To compensate for this, we assume that the error dynamics are independent of the position in space and shift the complete catching position trajectory by the average catching bias of all past attempts.

VII. Results

A. Experimental Setup

The algorithms presented in this paper were implemented in the Flying Machine Arena [19], an indoor testbed for aerial vehicles at ETH Zurich. The algorithms run offboard on conventional desktop computers and control commands are sent to the quadrocopters at a frequency of $50\,\mathrm{Hz}$ through a low-latency radio link. The estimator receives position and attitude data about the quadrocopters and the pole from an infrared motion capture system at a rate of $200\,\mathrm{Hz}$. The estimator predicts the system state in order to compensate for the closed-loop latency, and provides the feedback controllers with full state information.

All experiments were carried out with modified Ascending Technologies 'Hummingbird' quadrocopters [20], equipped with custom electronics. A circular plate $0.12\,\mathrm{m}$ in diameter is mounted on top of the vehicles, approximately $0.03\,\mathrm{m}$ above the vehicle's center of gravity, and serves as a supporting base for throwing, catching and balancing the pole. The pole used in the experiments consists of a carbon fibre tube with length $2L = 1.33\,\mathrm{m}$ and weight $m_p = 0.03\,\mathrm{kg}$. The aerodynamic drag properties were experimentally identified to be $c_z = 0.5\,\mathrm{g\,m^{-2}}$, $c_{xy} = 3.5\,\mathrm{g\,m^{-2}}$ and $c_{rot} = 2.1\,\mathrm{g\,m^{-2}}$. Shock absorbers, consisting of a flour-filled balloon on a sliding metal cap, are attached to

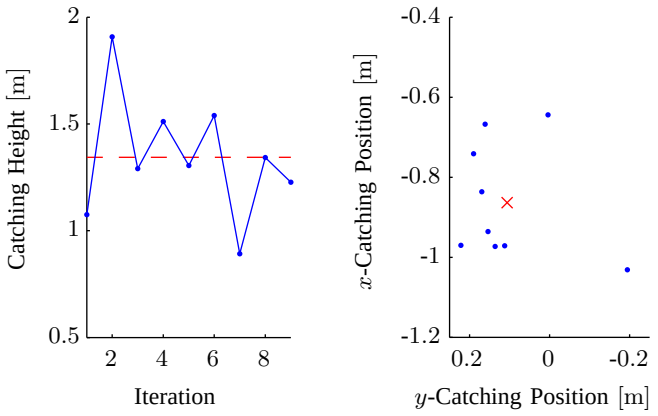


Fig. 9. Resulting optimal catching heights and catching positions for a set of 9 consecutive throws. The figure on the left shows the optimal catching height of each throw. The mean catching height, depicted by the dashed line, is at $1.343\,\mathrm{m}$. The plot on the right shows the impact position relative to the nominal impact position, caught at a height of $1.343\,\mathrm{m}$. The cross represents the mean catching point $(\mu_x, \mu_y) = (-0.863\,\mathrm{m}, 0.106\,\mathrm{m})$ and the standard deviation is $(\sigma_x, \sigma_y) = (0.146\,\mathrm{m}, 0.128\,\mathrm{m})$.

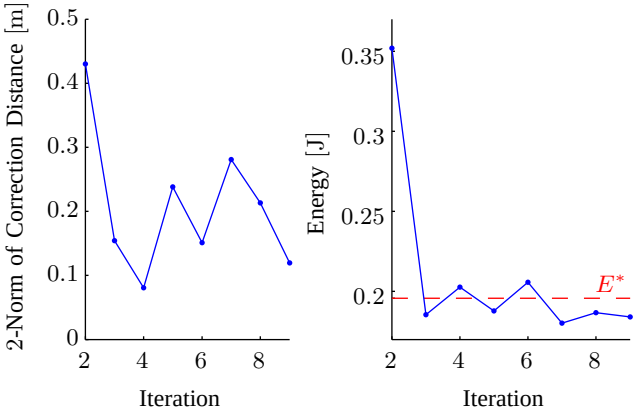


Fig. 10. The travel distance required to catch the pole and the pole energy after the impact when the catching height and waiting position are adapted as proposed in Section VI. The dashed line in the right plot represents the target energy $E^* = m_p g L$.

both pole ends to dampen the impact on the catching plate. This justifies our assumption of a totally inelastic impact in Section III.

The throwing trajectory is generated offline using the optimal control toolbox ACADO [21] in order to solve the optimization problem (34) numerically. Finite-horizon time-varying LQR controllers, obtained by linearizing the system dynamics (8) or (30) about the desired trajectories, are used to track the nominal throwing and catching maneuver. For the purpose of balancing the pole before the throw and after the catch, the infinite-horizon LQR controller from [7], tuned for a large basin of attraction, is applied.

B. Experimental Results

A typical series of 9 throws (with the same parameters as in Fig. 5) and 8 catches were executed. In the first run, only the throw maneuver was performed and the catching height and position were updated. For the iterations $2-7$, the catch maneuver was performed with a horizontal offset of $1\,\mathrm{m}$. During these iterations, the adapted coefficients converged. For the throws 8 and 9, the catching quadrocopter attempted to catch the pole. Fig. 8 shows an image sequence of a successful throw and catch attempt.

1) Pole Throwing: The launch of the pole is detected by the estimator approximately $0.21\,\mathrm{s}$ after the nominal launch point, highlighting a good match between the nominal design and experimental results. The flight duration until the optimal catching instant lasts on average $0.812\,\mathrm{s}$. From the launch of the pole, a further $0.075\,\mathrm{s}$ elapse before the catching position prediction converges. In order to catch all throws reliably, the quadrocopter must be at the impact position slightly before the average impact ($0.075\,\mathrm{s}$). This results in a correction time T_{xy} of approximately $0.66\,\mathrm{s}$.

Fig. 9 shows the optimal catching heights and the catching positions. The mean optimal catching height is at $1.343\,\mathrm{m}$. The average catching position, relative to the nominal catching position, is located at $(x, y) = (-0.863\,\mathrm{m}, 0.106\,\mathrm{m})$.

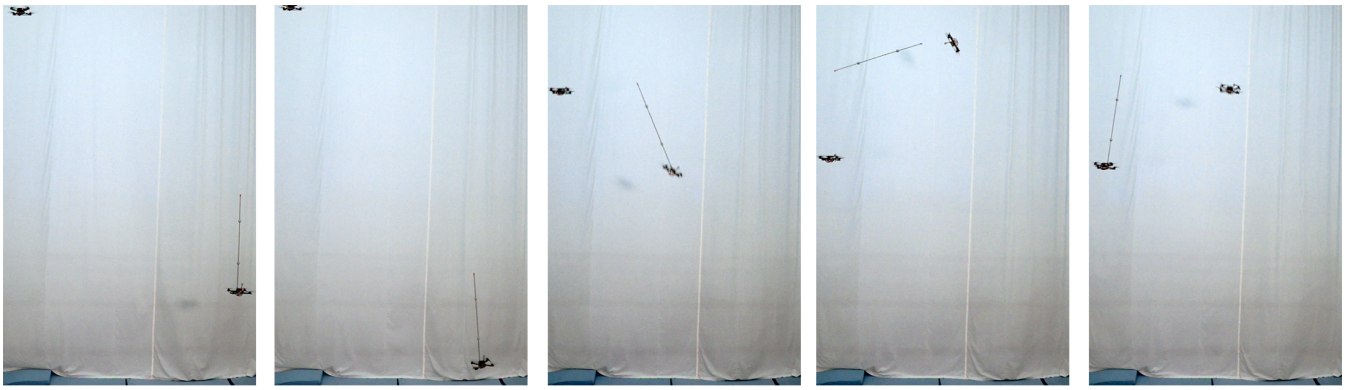
2) Pole Catching and Adaptation: The results of the applied adaptation strategies are shown in Fig. 10. The travel distance required to catch the pole converges to roughly $0.2\,\mathrm{m}$, which, as expected, is almost exactly the standard deviation of the catching point. The catching attempts in iteration 2 and 7 were aborted because the catching point was too far away from the catching vehicle. Although the optimal catching heights vary significantly (see Fig. 9), the pole energy after the impact is close to the target energy energy $E^* = m_p g L$ when caught at a previously defined height. After iteration 2, the deviation to the target energy is on average $0.0102\,\mathrm{J}$, where $0.01\,\mathrm{J}$ is equivalent to the pole being at rest at a tilt angle of $\alpha = 18.4°$. In both attempts 8 and 9, the catching vehicle successfully caught the pole.

A video showing the system presentend herein is available online at `http://bit.ly/2J2w9CC`.

VIII. Conclusion and Outlook

We have presented a method that allows a quadrocopter to balance a pole and throw it towards a second quadrocopter, which catches and re-balances it. Based on first principles models, a condition for launching the pole off the quadrocopter was identified and a trajectory to this state was generated. Furthermore, it was shown that the real-time correction maneuver in the xy-plane, which is necessary to catch the pole, can be tracked precisely by simultaneously flying an appropriate maneuver in the z-direction. An adaptation strategy was employed to adapt the catching height and expected catching position from previous iterations. Additionally, a strategy was implemented to compensate for the systematic offset error shown by the catching vehicle. The feasibility of these approaches was validated in the Flying Machine Arena.

Due to the high sensitivity to disturbances and the short amount of time to react to them, we intend to use the presented system as a testbed for high-performance trajectory generation algorithms and adaptation methods. Experiments with this system have highlighted various possibilities for extensions and improvements. The nominal throw maneuver



a) b) c) d) e)

Fig. 8. Image sequence of a successful throw and catch attempt: a) The quadrocopters wait at their nominal starting position. b) As soon as the throw maneuver starts, the catching quadrocopter shifts its position to compensate for the offset error of the throwing vehicle at the beginning of the throw maneuver. c) While the throwing quadrocopter is tracking the throw trajectory, the catching vehicle begins to fly towards the expected catching position such that the vehicle is in the high-thrust phase of the catching trajectory when the first catching position is predicted. d) While the pole is flying in the air, the catching position is predicted 50 times per second and the catching maneuver is adapted accordingly. e) As soon as the pole hits the catching quadrocopter, it begins to balance the pole.

design could be improved by not fixing the terminal state, but explicitly optimizing under the lift-off constraint. More sophisticated learning algorithms could be used to further eliminate systematic errors. While errors in the throw maneuver are currently accounted for in the catching phase, methods for improving the throw maneuver based on past trials should bring the executed maneuver closer to the nominal design.

Acknowledgments

This work is supported by and builds upon prior contributions by numerous collaborators in the Flying Machine Arena project. A list of past and present participants of the project is available at `http://bit.ly/RdO6g1`.
This research was funded in part by the Swiss National Science Foundation (SNSF).

References

[1] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *Control Systems, IEEE*, vol. 28, pp. 51 –64, 2008.

[2] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 279–284, 2012.

[3] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, pp. 664–674, 2012.

[4] S. Lupashin, A. Schoellig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1642 –1648, 2010.

[5] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed Autonomous Robotic Systems*, vol. 83, pp. 545–558, Springer, 2013.

[6] R. Ritz, M. Mueller, M. Hehn, and R. D'Andrea, "Cooperative quadrocopter ball throwing and catching," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4972 –4978, 2012.

[7] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 763 –770, 2011.

[8] M. Mueller, S. Lupashin, and R. D'Andrea, "Quadrocopter ball juggling," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5113 –5120, 2011.

[9] S. Jinglai, A. K. Sanyal, N. A. Chaturvedi, D. Bernstein, and H. McClamroch, "Dynamics and control of a 3d pendulum," in *43rd IEEE Conference on Decision and Control (CDC)*, vol. 1, pp. 323 – 328, 2004.

[10] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, pp. 1–35, 2006.

[11] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, "Rigid-body attitude control," *Control Systems, IEEE*, vol. 31, pp. 30–51, June.

[12] D. Brescianini, M. Hehn, and R. D'Andrea, "Quadrocopter pole acrobatics - online appendix." [Online], March 2013. Available: http://bit.ly/2pN1O2q.

[13] C. Glocker and F. Pfeiffer, "Multiple impacts with friction in rigid multibody systems," *Nonlinear Dynamics*, vol. 7, pp. 471–497, 1995.

[14] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[15] I. Hwang, H. Balakrishnan, and C. Tomlin, "State estimation for hybrid systems: applications to aircraft tracking," *Control Theory and Applications, IEEE*, vol. 153, pp. 556 – 566, 2006.

[16] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2520 –2525, 2011.

[17] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[18] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," in *18th IFAC World Congress*, vol. 18, pp. 1485–1491, 2011.

[19] S. Lupashin, A. Schoellig, M. Hehn, and R. D'Andrea, "The flying machine arena as of 2010," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2970 –2971, 2011.

[20] D. Gurdan, J. Stumpf, M. Achtelik, K. M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 khz," in *IEEE International Conference on Robotics and Automation*, pp. 361–366, 2007.

[21] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit - an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, pp. 298–312, 2011.