AIAA SciTech Forum
January 3-7, 2022, San Diego, CA & Virtual
AIAA SCITECH 2022 Forum
10.2514/6.2022-1396

Robust Trajectory Tracking by a Multicopter Platform: A Game Against Nature

Bhargav Jha* and Ujjwal Gupta†
Technion-Israel Institute of Technology, Haifa, Israel, 3200003

Vladimir Turetsky‡
Ort Braude College of Engineering, Karmiel, Israel, 2161002

Tal Shima§
Technion-Israel Institute of Technology, Haifa, Israel, 3200003

In this paper, the problem of robust trajectory tracking by a multicopter platform is formulated as a linear-quadratic differential game against unknown external disturbances (nature). The desired trajectory is composed of position and yaw constraint, which is known to be a differentially flat output for multicopter state dynamics. Using this property, a linearized error dynamics is obtained in the vicinity of the nominal states and the nominal control inputs. To avoid singularities, the linearized error dynamics is derived using quaternion representation. Based on it, an optimal saddle point strategy for trajectory tracking is presented, where controller gains are calculated *apriori* by numerically solving a differential Ricatti equation. A salient feature of this control design is that both the position and the attitude control loops are integrated, and there are only two tuning parameters — one corresponding to the trade-off between control effort and tracking accuracy and the other corresponding to robustness. Simulations as well as experimental validations are presented to demonstrate the performance and applicability of the controller.

I. Introduction

Low cost, simple mechanical design, and small form factor have made multicopters ubiquitous in both civilian and military domains. Their wide application domains include surveillance, search and rescue works in inaccessible environments, agriculture, delivery services, and many more. Autonomous safe traversal of multicopters in complex environments involves two hierarchical steps. The first step is the planning step that considers environmental and dynamical constraints to generate a safe reference trajectory for the second step, which accurately tracks it in the presence of external disturbances such as wind, aerodynamic drag, *etc*. In most cases, the planning step is executed at a significantly slower rate than the trajectory tracking step. The focus of this work lies in the trajectory tracking step.

Non-linear robust control approaches such as sliding mode controller design have been used to compensate for external disturbances as in [1, 2]. In [1], the authors have proposed an integral sliding mode controller, while [2] achieves robust tracking by combining a sliding mode disturbance compensator with a proportional-derivative (PD) controller. [3, 4] uses a backstepping controller to achieve robust trajectory tracking. Some other approaches linearize the multicopter dynamics about its hover point using small-angle approximation. This paves the way for the use of linear control theory. The approach in [5] minimizes the tracking error and control effort using optimal linear quadratic regulator (LQR) which leads to a static-gain state feedback controller. The problem is formulated as an infinite-horizon optimal control problem, therefore the static gain is computed efficiently by solving an algebraic Ricatti equation. Similarly, by considering small-angle approximations, [6] proposed an H_∞ robust controller for tracking. Compensating for the aerodynamic effects and avoiding the small-angle approximations, the authors in [7] have used feedback linearisation along with cascaded proportional-integral-derivative (PID) controllers to design separate control loops for position and attitude control.

*Doctoral Student, Aerospace Engineering, Technion-Israel Institute of Technology, jhabhargav@campus.technion.ac.il
†Research Assistant, Aerospace Engineering, Technion-Israel Institute of Technology, ujjwalgupta.iitr@gmail.com
‡Professor, Department of Mathematics, Ort Braude College of Engineering, turetsky1@braude.ac.il
§Professor, Aerospace Engineering, Technion-Israel Institute of Technology, tal.shima@technion.ac.il

In [8], it is shown that for a careful choice of outputs containing positional and yaw constraints, the quadcopter dynamics is differentially flat. This enables the reconstruction of the reference control inputs and the rest of the states in terms of higher derivatives of position and yaw angle. The same choice of outputs has been shown to be differentially flat for the case of quadcopter dynamics with drag [9] and also for a hexacopter dynamics [10]. Using this idea of differential flatness, works such as [8–11] have demonstrated precise trajectory tracking. In [12], the trajectory tracking controller is decomposed into two cascaded loops corresponding to attitude control and position control. A PD controller and a method similar to backstepping control is used for postion and attitude control, respectively. [9] achieves robust tracking in the presence of aerodynamic drag by modeling and incorporating a drag term in the state dynamics. Thereafter, a PD controller with a feed-forward acceleration term is used for the position control loop and a Lyapunov based controller [13, 14] is applied for the attitude control loop. The authors in [11] use incremental non-linear dynamic inversion in conjunction with PD control for both position and attitude control loops. For hexacopters, the authors in [10], have proposed the use of differential flatness together with an infinite horizon optimal LQR controller. Here the output of the LQR controller is mapped to the input of a hexacopter by using a state-dependent inverse function which can lead to singularities.

The non-linear control approaches can offer the benefit of robustness and finite time global convergence but do not consider the optimality of some performance index. Even if formulated, solving a non-linear optimal control problem for robust trajectory tracking remains intractable. Additionally, the non-linear controllers along with some linear controllers such as cascaded PD, need careful tuning of several gain parameters. The optimal LQR and robust H_∞ controllers make controller tuning simpler, but their performance is limited either by the validity of small-angle approximations near the hover point or by singularities that arise due to feedback linearisation.

In this paper, we formulate the trajectory tracking problem as a linear-quadratic differential game against an unknown disturbance. Both position and attitude loops are integrated into a single control loop. Unlike the aforementioned works involving linearisation, we perform the linearisation in the vicinity of the given nominal trajectory, which avoids singularities and is valid for cases where small-angle approximations do not hold. In addition to it, we assume the presence of unknown bounded additive control disturbances, which has two-fold advantages: first, it accounts for modeling uncertainties that may arise due to linearisation, and second, it can compensate for external disturbances. Moreover, for any admissible trajectory, the controller gains can be efficiently calculated offline, and there are only two control parameters whose tuning can be done intuitively. The proposed controller is developed to control any multirotor platform in which the thrust vector is perpendicular to the body plane and it accepts body rates and thrust command as inputs. This increases the applicability of the method to many off-the-shelf multicopters irrespective of the number of rotors. Similar approach has been previously explored for trajectory tracking by a Dubins vehicle in [15, 16].

II. Preliminaries and Problem Statement

The objective of our work is to track a desired trajectory by a multicopter platform in the presence of external disturbances. The desired trajectory $\Gamma(t) \triangleq \begin{bmatrix} x^*(t) & y^*(t) & z^*(t) & \psi^*(t) \end{bmatrix}$ is composed of position reference $\begin{bmatrix} x^*(t) & y^*(t) & z^*(t) \end{bmatrix}$ and yaw reference $\psi^*(t)$, for all $t \in [0, t_f]$, where t_f denotes the final time. Consider an inertial frame $(\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W)$ and the frame attached to the center of mass of the multicopter platform as $(\mathbf{x}_B(t), \mathbf{y}_B(t), \mathbf{z}_B(t))$. The thrust applied by the rotors of the multipcopter platform is along the vector $\mathbf{z}_B(t)$ and the acceleration due to the gravity (g) is in $-\mathbf{z_W}$ direction. The rotation matrix from the body frame to the inertial frame is denoted by $\mathbf{R}(t)$.

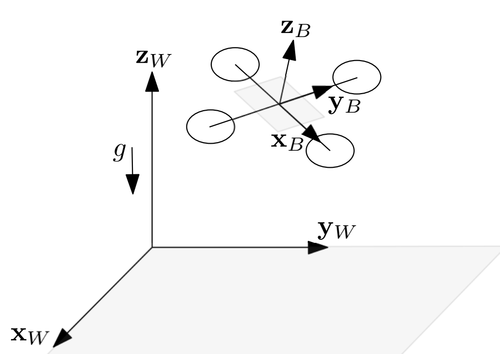


Fig. 1 Inertial and Body Frames

The state vector is defined as $\mathbf{X}(t) = \begin{bmatrix}\mathbf{p}(t) & \mathbf{v}(t) & \mathbf{q}(t)\end{bmatrix}^\top$. Here $\mathbf{p}(t) = \begin{bmatrix}x(t) & y(t) & z(t)\end{bmatrix}^\top \in \mathbb{R}^3$, $\mathbf{v}(t) \in \mathbb{R}^3$, and $\mathbf{q}(t) = \begin{bmatrix}q_0(t) & q_1(t) & q_2(t) & q_3(t)\end{bmatrix}^\top \in SO(3)$ denote its position, velocity and rotation as a unit-normed quaternion, respectively, in the inertial frame. We assume that the mass of the platform is known. Now denote the control input vector as $\mathbf{U}(t) = \begin{bmatrix}\tau(t) & \boldsymbol{\omega}(t)^\top\end{bmatrix}^\top$, where $\tau(t) \in \mathbb{R}$ and $\boldsymbol{\omega}(t) = \begin{bmatrix}\omega_x(t) & \omega_y(t) & \omega_z(t)\end{bmatrix}^\top \in \mathbb{R}^3$ correspond to the specific thrust and the body rate commands, respectively. In order to account for unknown exogenous forces such as model uncertainties, aerodynamic drag *etc*, additive input disturbances $\tau_\Delta(t) \in L_2[0,t_f]$ and $\boldsymbol{\omega}_\Delta(t) \in L_2^3[0,t_f]$ are introduced in the state dynamics of the multicopter. These disturbances are assumed to be bounded such that $\|\tau_\Delta\| \leq \mu$ and $\|\boldsymbol{\omega}_\Delta\| \leq \nu$.

The state dynamics of the platform is given by,

$$\dot{\mathbf{p}}(t) = \mathbf{v}(t) \tag{1a}$$

$$\dot{\mathbf{v}}(t) = -g\mathbf{z}_w + (\tau(t) + \tau_\Delta(t))\,\mathbf{z}_B(t) \tag{1b}$$

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\begin{bmatrix}Q(t) & W(t)^\top\end{bmatrix}^\top (\boldsymbol{\omega}(t) + \boldsymbol{\omega}_\Delta(t)) \tag{1c}$$

The matrices $\mathrm{Q}(t) \in \mathbb{R}^{3\times 1}$ and $\mathrm{W}(t) \in \mathbb{R}^{3\times 3}$ are defined as,

$$Q(t) = \begin{bmatrix}-q_1(t)\\ -q_2(t)\\ -q_3(t)\end{bmatrix}, \qquad W(t) = \begin{bmatrix}q_0(t) & -q_3(t) & q_2(t)\\ q_3(t) & q_0(t) & -q_1(t)\\ -q_2(t) & q_1(t) & q_0(t)\end{bmatrix} \tag{2}$$

A concise introduction to quaternion and its relation to rotation matrix can be found in [17]. Assume zero disturbances, Eq. (1c) can be equivalently represented as in terms of rotation matrix $\mathbf{R}(t)$ as,

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t)\hat{\omega}(t), \qquad \hat{\omega}(t) \triangleq \begin{bmatrix}0 & -\omega_z(t) & \omega_y(t)\\ \omega_z(t) & 0 & -\omega_x(t)\\ -\omega_y(t) & \omega_x(t) & 0\end{bmatrix} \tag{3}$$

This representation is sometimes more familiar and easy to use. However, the use of the quaternion representation avoids the problem of gimbal lock. Moreover, it leads to a concise representation of the linearized error dynamics of Eq. (1), as we will discuss in III.B.

III. Linearization

In [8, 9], for a quadcopter dynamics, the choice of the reference $\Gamma(t)$ is shown to be differentially flat. In the context of our problem, it means that if $\Gamma(t)$ is specified such that the first order derivative of $\psi^*(t)$ and third order derivatives of $x^*(t), y^*(t)$, and $z^*(t)$ exist, then the nominal control input $\mathbf{U}^*(t) = \begin{bmatrix}\tau^*(t) & \boldsymbol{\omega}^*(t)^\top\end{bmatrix}^\top$ and thus the nominal states $\mathbf{X}^*(t) = \begin{bmatrix}\mathbf{x}^*(t) & \mathbf{v}^*(t) & \mathbf{q}^*(t)\end{bmatrix}^\top$ can be reconstructed. This will facilitate the linearization of the state dynamics (Eq. (1)) in the vicinity of the nominal states $\mathbf{X}^*(t)$, which will be discussed in this section.

A. Reconstruction of Nominal States and Control Inputs

Now proceeding on similar lines as [18], we can reconstruct the nominal states and the control inputs. The nominal orientation $\mathbf{R}^*(t) = \begin{bmatrix}\mathbf{x}_B^*(t) & \mathbf{y}_B^*(t) & \mathbf{z}_B^*(t)\end{bmatrix}$ can be reconstructed given the nominal yaw angle ($\psi^*(t)$) and reference acceleration $\dot{\mathbf{v}}^*(t)$. At each time instant (t), consider a unit vector along the projection of $\mathbf{x}_B^*(t)$ on the $\mathbf{x}_W - \mathbf{y}_W$ plane and denote it as $\mathbf{x}_C(t)$. Denote another vector $\mathbf{y}_C(t)$ that is orthogonal to $\mathbf{x}_C(t)$ and lies on the plane $\mathbf{x}_W - \mathbf{y}_W$. Note that there are two possible vectors for $\mathbf{y}_C(t)$. In this work, we do not consider inverted flights, therefore, we consider $\mathbf{y}_C(t)$ such that $\mathbf{z}_W \cdot (\mathbf{x}_C(t) \times \mathbf{y}_C(t)) = 1$. Hence, the vectors $\mathbf{x}_C(t)$ and $\mathbf{y}_C(t)$ can be represented as,

$$\mathbf{x}_C(t) = \begin{bmatrix}\cos\psi^*(t) & \sin\psi^*(t) & 0\end{bmatrix}^\top \qquad \mathbf{y}_C(t) = \begin{bmatrix}-\sin\psi^*(t) & \cos\psi^*(t) & 0\end{bmatrix}^\top \tag{4}$$

As the nominal trajectory does not include the disturbances ($\tau_\Delta(t) = 0$, $\omega_\Delta(\mathbf{t})$), from Eq. 1b we obtain,

$$\frac{1}{\tau^*(t)}\left(\dot{\mathbf{v}}^*(t) + g\mathbf{z}_W\right) = \mathbf{z}_B^*(t) \tag{5}$$

Now consider a plane passing through $\mathbf{x}_C(t)$ and normal to $\mathbf{y}_C(t)$. Note that this plane contains $\mathbf{x}^*_B(t)$. Consider another plane with the normal vector $\mathbf{y}_C(t) \times \mathbf{z}^*_B(t)$. As $\mathbf{x}^*_B(t) \perp \mathbf{y}_C(t)$ and $\mathbf{x}^*_B(t) \perp \mathbf{z}^*_B(t)$, we have

$$\mathbf{x}^*_B(t) = \frac{\mathbf{y}_C(t) \times \mathbf{z}^*_B(t)}{\|\mathbf{y}_C(t) \times \mathbf{z}^*_B(t)\|} = \frac{\mathbf{y}_C(t) \times (\dot{\mathbf{v}}^*(t) + g\mathbf{z}_W)}{\|\mathbf{y}_C(t) \times (\dot{\mathbf{v}}^*(t) + g\mathbf{z}_W)\|} \tag{6}$$

Similarly,

$$\mathbf{y}^*_B(t) = \frac{\mathbf{z}^*_B(t) \times \mathbf{x}^*_B(t)}{\|\mathbf{z}^*_B(t) \times \mathbf{x}^*_B(t)\|} = \frac{(\dot{\mathbf{v}}^*(t) + g\mathbf{z}_W) \times \mathbf{x}^*_B(t)}{\|(\dot{\mathbf{v}}^*(t) + g\mathbf{z}_W) \times \mathbf{x}^*_B(t)\|} \tag{7}$$

$$\mathbf{z}^*_B(t) = \mathbf{x}^*_B(t) \times \mathbf{y}^*_B(t) \tag{8}$$

As $\dot{\mathbf{v}}^*(t)$ is known, we can compute $\mathbf{R}^*(t)$ from the above equations. Now $\mathbf{q}^*(t)$ can be obtained by using the relation between rotation matrices and quaternions. We have $\mathbf{z}^*_B \cdot \mathbf{z}^*_B = 1$, hence from Eq. (5), the nominal specific thrust input is given by,

$$\tau^*(t) = \|\dot{\mathbf{v}}^*(t) + g\mathbf{z}_W\| \tag{9}$$

Now we will calculate the nominal body rate command $\boldsymbol{\omega}^*(t)$. Obtaining the time derivative of Eq. (1b), we have

$$\ddot{\mathbf{v}}^*(t) = \tau^*(t)\mathbf{R}^*(t)\hat{\omega}^*(t)\mathbf{z}_w + \dot{\tau}^*(t)\mathbf{z}^*_B(t). \tag{10}$$

This can be rewritten as,

$$\ddot{\mathbf{v}}^*(t) = \tau^*(t)\begin{bmatrix}\mathbf{x}^*_B(t) & \mathbf{y}^*_B(t) & \mathbf{z}^*_B(t)\end{bmatrix}\begin{bmatrix}0 & -\omega^*_z(t) & \omega^*_y(t)\\ \omega^*_z(t) & 0 & -\omega^*_x(t)\\ -\omega^*_y(t) & \omega^*_x(t) & 0\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix} + \dot{\tau}^*(t)\mathbf{z}^*_B(t). \tag{11}$$

Taking dot product on both sides of Eq. (11) with the vector $\mathbf{x}^*_B(t)$ we have,

$$\omega^*_y(t) = \frac{1}{\tau^*(t)}(\mathbf{x}^*_B)^\top(t)\ddot{\mathbf{v}}^*(t) \tag{12}$$

Similarly, taking dot product with the vector $\mathbf{y}^*_B(t)$ we have,

$$\omega^*_x(t) = -\frac{1}{\tau^*(t)}(\mathbf{y}^*_B)^\top(t)\ddot{\mathbf{v}}^*(t) \tag{13}$$

To calculate ω_z we can use the relation from Eq. (3) to obtain

$$(\mathbf{y}^*_B)^\top(t)\dot{\mathbf{R}}^*(t)\begin{bmatrix}1\\0\\0\end{bmatrix} = (\mathbf{y}^*_B)^\top(t)\mathbf{R}^*(t)\begin{bmatrix}0 & -\omega^*_z(t) & \omega^*_y(t)\\ \omega^*_z(t) & 0 & -\omega^*_x(t)\\ -\omega^*_y(t) & \omega^*_x(t) & 0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} \tag{14}$$

This implies,

$$\omega^*_z(t) = (y^*_B)^\top(t)\dot{\mathbf{x}}^*_B(t) = \frac{(y^*_B)^\top(t)\left(\dot{\mathbf{y}}_C(t) \times \mathbf{z}^*_B(t) + \mathbf{y}_C(t) \times \dot{\mathbf{z}}^*_B(t)\right)}{\|\mathbf{y}_C(t) \times \mathbf{z}^*_B(t)\|} \tag{15}$$

By some more manipulation with the vectors, we obtain

$$\omega^*_z(t) = \frac{\dot{\psi}^*(t)\mathbf{x}_C^\top(t)\mathbf{x}^*_B(t) + \omega_y\mathbf{y}_C^\top(t)\mathbf{z}^*_B(t)}{\|\mathbf{y}_C(t) \times \mathbf{z}^*_B(t)\|} \tag{16}$$

B. Linearized Error Dynamics

After computing the nominal control inputs and states, the system dynamics are linearized in the vicinity of the nominal trajectory. Construct an error vector $\mathbf{e}(t) \triangleq \mathbf{X}(t) - \mathbf{X}^*(t)$ and a vector $\mathbf{u}(t) \triangleq \mathbf{U}(t) - \mathbf{U}^*(t)$. By obtaining the Jacobian linearization of the system dynamics, the error dynamics can be obtained as,

$$\dot{\mathbf{e}}(t) = A(t)\,\mathbf{e}(t) + B(t)\,\mathbf{u}(t) + B(t)\,\mathbf{u}_\Delta(t) \tag{17}$$

The matrices $A(t)$ and $B(t)$ are defined as,

$$A(t) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times4} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & 2\tau A_v(t) \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \frac{1}{2}A_q(t) \end{bmatrix}, \quad B(t) = \begin{bmatrix} \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ B_\tau(t) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times1} & \frac{1}{2}B_\omega(t) \end{bmatrix}$$

where

$$A_v(t) = \begin{bmatrix} W(t)\mathbf{J}_3 & -Q(t) \end{bmatrix}, \; A_q = \begin{bmatrix} 0 & -\omega_x^*(t) & -\omega_y^*(t) & -\omega_z^*(t) \\ \omega_x^*(t) & 0 & \omega_z^*(t) & -\omega_y^*(t) \\ \omega_y^*(t) & -\omega_z^*(t) & 0 & \omega_x^*(t) \\ \omega_z^*(t) & \omega_y^*(t) & -\omega_x^*(t) & 0 \end{bmatrix},$$

$$B_\tau(t) = \begin{bmatrix} 2\left(q_0^*(t)q_2^*(t) + q_1^*(t)q_3^*(t)\right) \\ 2\left(q_2^*(t)q_3^*(t) - q_0^*(t)q_1^*(t)\right) \\ (q_0^*(t))^2 - (q_1^*(t))^2 - (q_2^*(t))^2 + (q_3^*(t))^2 \end{bmatrix}, \; B_\omega(t) = \begin{bmatrix} Q(t)^\top \\ W(t) \end{bmatrix}$$

and $\mathbf{J}_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

IV. Linear Quadratic Differential Game

Using the error dynamics (Eq. (17)), the problem of robust trajectory tracking can be posed as a two-player linear-quadratic differential game where the control input $\mathbf{u}(t)$ is the minimizer, and the disturbance $\mathbf{u}_\Delta(t)$ is the maximizer of the following cost function,

$$J = \int_0^{t_f} \mathbf{e}^\top(t)\mathbf{e}(t)\mathrm{d}t + \alpha\int_0^{t_f} \mathbf{u}^\top(t)\mathbf{u}(t)\mathrm{d}t - \beta\int_0^{t_f} \mathbf{u}_\Delta^\top(t)\mathbf{u}_\Delta(t)\mathrm{d}t \tag{18}$$

The parameters $\alpha > 0$ and $\beta > 0$ are the control and disturbance penalties, respectively. The game is formulated as

$$\max_{\mathbf{u}_\Delta(t)} \min_{\mathbf{u}(t)} J \tag{19}$$

Note that decreasing β increases the robustness against external disturbances and decreasing α increases the tracking accuracy at the cost of increasing the control effort. A major advantage of our proposed approach lies in the fact that there are only two tuning parameters for the controller compared to the cascaded PD controllers or non-linear controllers, where the number of tuning parameters can be significantly more.

It is shown in [19] that there exists a saddle point consisting of optimal strategies $\mathbf{u}^O(t)$ and $\mathbf{u}_\Delta^O(t)$, if α and β satisfy the following solvability condition of the game,

$$\alpha < \begin{cases} \frac{1}{\frac{1}{\beta} - C}, & \beta < \frac{1}{C} \\ \infty, & \beta \geq \frac{1}{C} \end{cases}, \quad C > 0 \tag{20}$$

The value of C can not be calculated *apriori*, therefore for all practical purposes, we consider $\alpha < \beta$ which satisfies the solvability condition. Hence, a precise tracking can be achieved when $\alpha \to 0$ and $\beta \to \alpha$. If the solvability condition is satisfied, then the minimizer's strategy is given by,

$$\mathbf{u}^O(t) = -\frac{1}{\alpha}B^\top(t)\Phi^\top(t_f,t)R(t)\Phi(t_f,t)\mathbf{e}(t) \tag{21}$$

where $\Phi(t_f, t)$ is the state transition matrix associated with the linearized error dynamics in Eq. (17) and $R(t)$ satisfies the following matrix Riccati differential equation,

$$\dot{R}(t) = \left(\frac{1}{\alpha} - \frac{1}{\beta}\right) R(t)\Phi(t_f,t)B(t)B^\top(t)\Phi^\top(t_f,t)R(t) - \Phi^\top(t,t_f)\Phi(t,t_f), \qquad R(t_f) = \mathbf{0} \tag{22}$$

See [20] for theoretical justifications. For a given reference $\Gamma(t)$, Eq. (22) can be solved offline by a numerical solver. Thereafter, $R(t)$ can be substituted to Eq. (21) to obtain the optimal command input $\mathbf{U}^O(t)$ to the multicopter platform as,

$$\mathbf{U}^O(t) = \mathbf{U}^*(t) - \frac{1}{\alpha}B^\top(t)\Phi^\top(t_f,t)R(t)\Phi(t_f,t)\mathbf{e}(t) \tag{23}$$

V. Numerical and Experimental Results

The performance and applicability of the controller were validated by numerical simulations and experiments. The simulations were conducted with Matlab 2020b installed on a system with Intel(R) Core(TM) i7-4790 CPU@3.60 GHz processor and 16.0 GB RAM. For numerical integration, ode45 was used. The experiments were conducted in the Cooperative Autonomous SYstem laboratory (CASY) in Technion. The lab is equipped with overhead motion capture (mocap) system for tracking 6-DOF (degrees of freedom) state of rigid bodies with sub-millimeter accuracy. The quadcopter used for our experiments is shown in Fig. 2. The frame of the quadcopter is a Diatone Taycan 3 inch Cinewhoop. The motors used are Cobra-1407, 4100 kV with 3 inch propellers mounted on them.

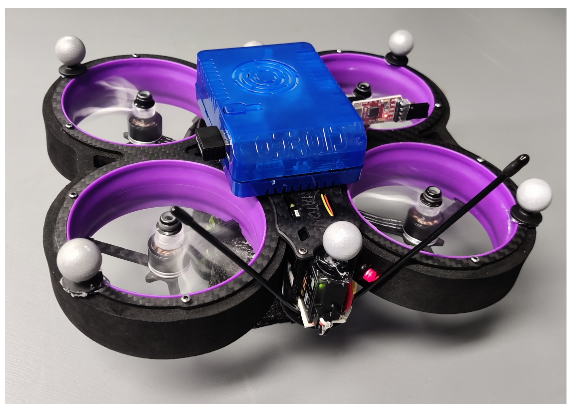


Fig. 2 Quadcopter used in the experiments

The position and attitude measurements are obtained from the mocap system at an update rate of 240 Hz. Moreover, the velocity measurement is estimated from the position data using Savitzky- Golay digital filter [21]. The feedback is directly fed to the proposed controller, which runs as a Robot Operating System (ROS) node on an Odroid XU4 onboard computer. The update rate of the controller is 200 Hz. A Pixracer flight controller is used as an interface between the onboard computer and the actuators. It is also responsible for tracking the body rates and thrust commands at an update rate of 1 kHz. The quadcopter together with all the components weighs 690 grams.

The position reference is a Viviani curve [22] and the yaw reference is zero. Table 1 summarizes the equation of the curve and initial parameters.

A. Simulation

In this section, we will present the performance of the controller for two cases. First, in the absence of external disturbances, and second, in the presence of external disturbances.

1. Case 1: No disturbance

Fig. 4 shows the tracking accuracy for the case of no external disturbances, *i.e.* $\tau_\Delta(t) \equiv 0$ and $\omega_\Delta(t) \equiv 0$. In this case, $\beta \to \infty$, so the problem is similar to a linear-quadratic regulator. The reference position at $t = 0$ is at $[1 \;\; 0 \;\; 1.5]$,

Parameter	Value
$x^*(t)\ (m)$	$a\left(1+\cos\left(\frac{\pi}{4}t\right)\right)$, $a=1$ (Simulations), $a=0.75$ (Experiments)
$y^*(t)\ (m)$	$a\sin\left(\frac{\pi}{4}t\right)$
$z^*(t)\ (m)$	$a\sin\left(\frac{\pi}{8}t\right)+1.5$
$\psi^*(t)\ (deg)$	0
$\mathbf{p}(0)\ (m)$	[0, 0, 0]
$\mathbf{v}(0)\ (m/s)$	[0, 0, 0]
$\mathbf{q}(0)$	[1, 0, 0, 0]

Table 1 Reference trajectory and initial parameters

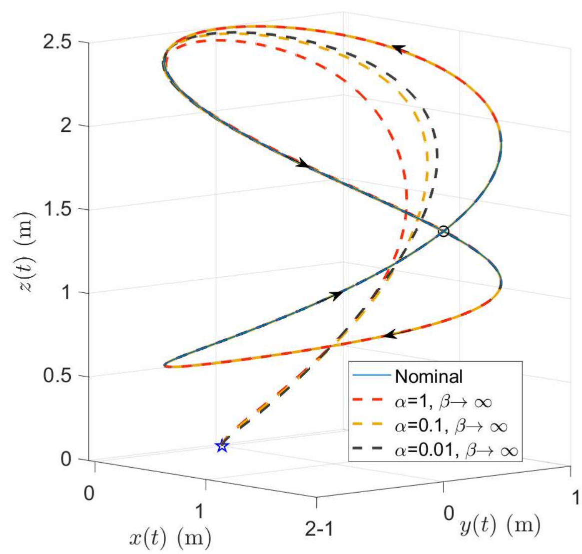
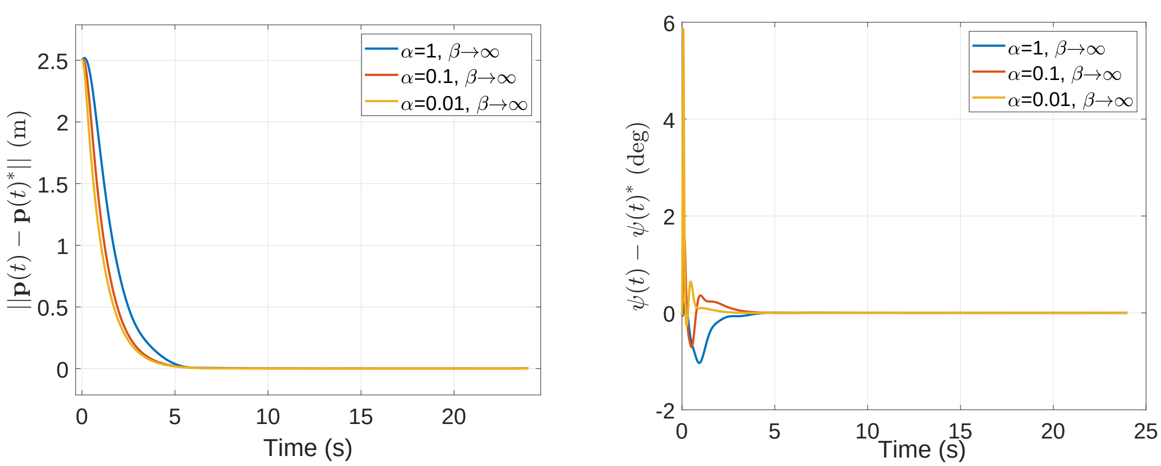


Fig. 3 Trajectory tracking with no disturbance in simulations



(a) Position Tracking (b) Yaw Tracking

Fig. 4 Effect of α on tracking accuracy

which is the intersection point of the two loops in the Viviani's curve. The time-varying reference point starts from this intersection point and moves in a counter-clockwise direction, as shown by the arrows in the figure. It can be seen that for $\alpha = 0.01$, the controller compensates more aggressively and minimizes the error. For $\alpha = 1$, the controller prioritizes the control effort and deviates more from the trajectory. This tradeoff between the control effort and the

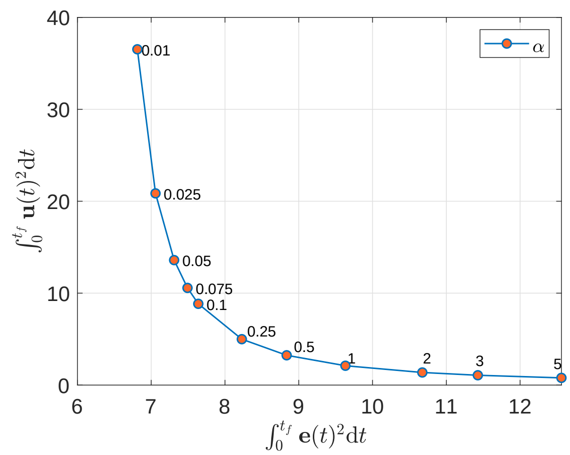


Fig. 5 Pareto front showing the tradeoff between tracking accuracy and control effort

integral square error can be seen in the Pareto front shown in Fig. 5. With increasing α, the control effort decreases and the error increases.

2. Case 2: External Drag

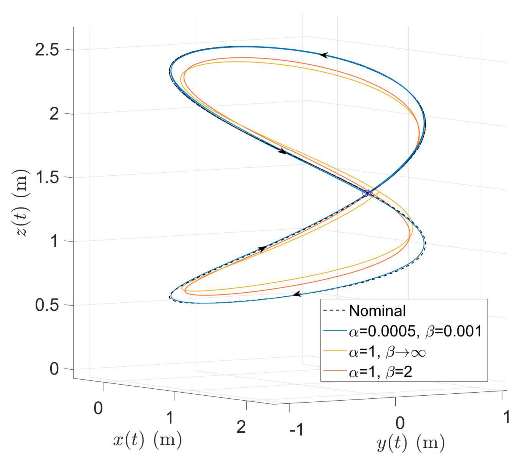
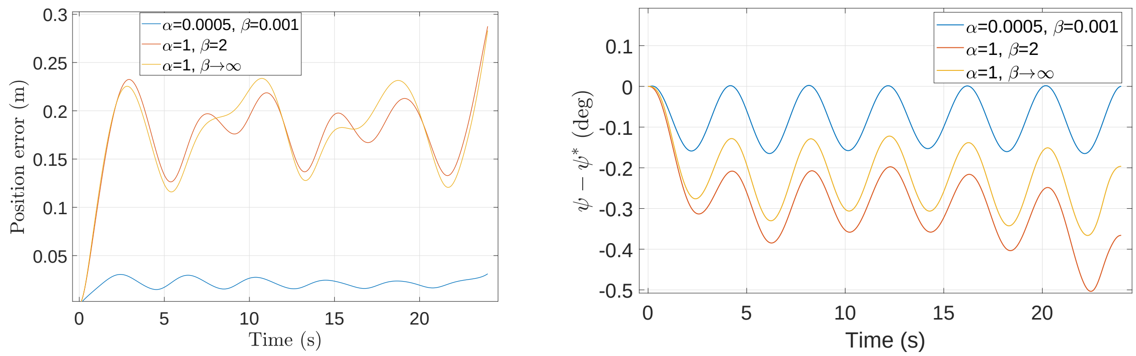


Fig. 6 Trajectory tracking in the presence of disturbance

To validate the robustness of the proposed controller in the presence of external disturbances, we present some simulation results with artificially induced disturbances. The disturbance model for τ_Δ is the same as the additional drag term modeled in [9]. In addition to it, we also add ω_Δ as the worst-case constant disturbance whose value is chosen as 10% of the peak values of ω_x, ω_y, and ω_z. The effect of this induced disturbance can be seen in Fig. 6. For $\alpha = 1, \beta \to \infty$, the tracking deteriorates significantly as compared to the case with no disturbance in Fig. 4. For the same α, as β is reduced, the tracking improves, and at the low value of $\alpha = 0.0005$ and $\beta = 0.001$, the effect to disturbance is attenuated and near-perfect tracking is achieved. This can also be seen from the position and yaw error plots shown in Fig. 7.



(a) Position Tracking

(b) Yaw Tracking

Fig. 7 Effect of α and β on tracking in the presence of external disturbance

B. Experiment

The tracking results from the experiments are shown in Fig. 8. It can be seen that robust trajectory tracking is achieved for very small values of α and β. At $t = 0$, we have introduced an initial position error of 2.1 m away from the starting point trajectory. This highlights the fact that, although the error dynamics is obtained from the linearization about the nominal trajectory, the proposed controller can compensate for significant errors. Once the quadcopter's trajectory converges to the desired trajectory (at $t \approx 4s$), the position tracking error is within 0.06 m (Fig. 9(a)), and the yaw tracking error (Fig. 9(b)) is less than 0.3 degrees.

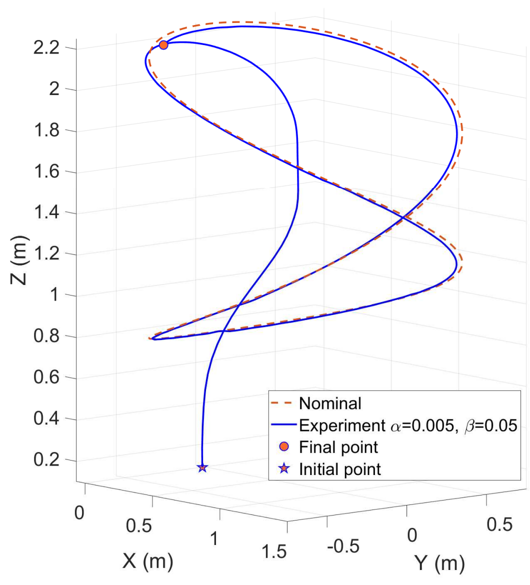
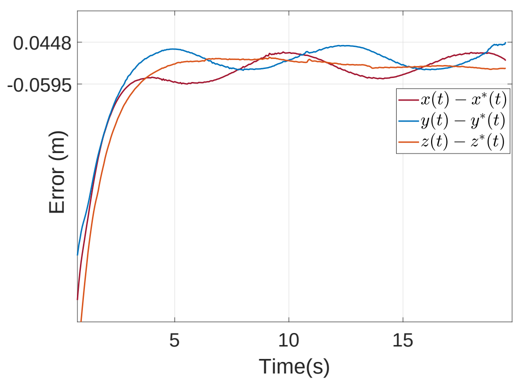
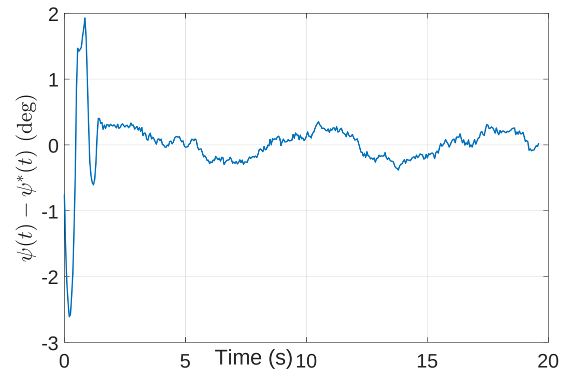


Fig. 8 Trajectory tracking in the experiment



(a) Position Error



(b) Yaw Error

Fig. 9 Position and yaw error in the experiment

VI. Conclusions

A robust trajectory tracking for a multicopter using a linear-quadratic differential game formulation is presented. The non-linear dynamics of the multicopter is represented using quaternions, which avoids singularities and also facilitates a compact representation of the linearized dynamics. Using Jacobian linearization about the nominal trajectory, a linear time-varying error dynamics is obtained. Using the error dynamics, a two-player differential game against nature is formulated. For the existence of the saddle point strategies, the conditions on the game parameters $(\alpha,\ \beta)$ are also discussed. The two game parameters are the only tuning parameters for calculating the controller gains, which can be efficiently calculated offline for a given reference trajectory. The effect of parameters on the tracking accuracy, control effort, and disturbance attenuation is presented using simulations. Results from the experiment show accurate tracking and exemplify the performance of the proposed controller under real-world disturbances. As a part of future goals, we aim to incorporate more real-world constraints such as measurement noise, internal dynamics for thrust and body rate tracking, and high speed reference trajectories. Work in this direction is currently in progress.

Acknowledgments

This research was partially supported by the A. Pazy Research Foundation.

References

[1] Ton, C. T., and Mackunis, W., "Robust attitude tracking control of a quadrotor helicopter in the presence of uncertainty," *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, IEEE, 2012, pp. 937–942.

[2] Liu, H., Bai, Y., Lu, G., Shi, Z., and Zhong, Y., "Robust tracking control of a quadrotor helicopter," *Journal of Intelligent & Robotic Systems*, Vol. 75, No. 3, 2014, pp. 595–608.

[3] Frazzoli, E., Dahleh, M. A., and Feron, E., "Trajectory tracking control design for autonomous helicopters using a backstepping algorithm," *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, Vol. 6, IEEE, 2000, pp. 4102–4107.

[4] de Vries, E., and Subbarao, K., "Backstepping based nested multi-loop control laws for a quadrotor," *2010 11th International Conference on Control Automation Robotics & Vision*, IEEE, 2010, pp. 1911–1916.

[5] Cowling, I. D., Yakimenko, O. A., Whidborne, J. F., and Cooke, A. K., "Direct method based control system for an autonomous quadrotor," *Journal of Intelligent & Robotic Systems*, Vol. 60, No. 2, 2010, pp. 285–316.

[6] Chen, M., and Huzmezan, M., "A combined mbpc/2 dof h infinity controller for a quad rotor uav," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003, p. 5520.

[7] Huang, H., Hoffmann, G. M., Waslander, S. L., and Tomlin, C. J., "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 3277–3282.

[8] Mellinger, D., and Kumar, V., "Minimum snap trajectory generation and control for quadrotors," *2011 IEEE International Conference On Robotics and Automation*, IEEE, 2011, pp. 2520–2525.

[9] Faessler, M., Franchi, A., and Scaramuzza, D., "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, Vol. 3, No. 2, 2017, pp. 620–626.

[10] Ferrin, J., Leishman, R., Beard, R., and McLain, T., "Differential flatness based control of a rotorcraft for aggressive maneuvers," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Ieee, 2011, pp. 2688–2693.

[11] Tal, E., and Karaman, S., "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Transactions on Control Systems Technology*, 2020.

[12] Mellinger, D., Michael, N., and Kumar, V., "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, Vol. 31, No. 5, 2012, pp. 664–674.

[13] Mayhew, C. G., Sanfelice, R. G., and Teel, A. R., "Quaternion-based hybrid control for robust global attitude tracking," *IEEE Transactions on Automatic control*, Vol. 56, No. 11, 2011, pp. 2555–2566.

[14] Brescianini, D., Hehn, M., and D'Andrea, R., "Nonlinear quadrocopter attitude control: Technical report," Tech. rep., ETH Zurich, 2013.

[15] Jha, B., Turetsky, V., and Shima, T., "Robust path tracking by a Dubins ground vehicle," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 6, 2018, pp. 2614–2621.

[16] Jha, B., Turetsky, V., and Shima, T., "Linear-Quadratic Robust Path Tracking for a Dubins Vehicle," *2018 European Control Conference (ECC)*, IEEE, 2018, pp. 2101–2106.

[17] Graf, B., "Quaternions and dynamics," *arXiv preprint arXiv:0811.2889*, 2008.

[18] Faessler, M., Franchi, A., and Scaramuzza, D., "Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories," *CoRR*, Vol. abs/1712.02402, 2017. URL `http://arxiv.org/abs/1712.02402`.

[19] Shinar, J., Turetsky, V., Glizer, V. Y., and Ianovsky, E., "Solvability of linear-quadratic differential games associated with pursuit-evasion problems," *International Game Theory Review*, Vol. 10, No. 04, 2008, pp. 481–515.

[20] Turetsky, V., Glizer, V. Y., and Shinar, J., "Robust trajectory tracking: differential game/cheap control approach," *International Journal of Systems Science*, Vol. 45, No. 11, 2014, pp. 2260–2274.

[21] Schafer, R. W., "What is a Savitzky-Golay filter?[lecture notes]," *IEEE Signal processing magazine*, Vol. 28, No. 4, 2011, pp. 111–117.

[22] Weisstein, E. W., "Viviani's Curve," , ???? URL `https://mathworld.wolfram.com/VivianisCurve.html`, champaign, IL, 2021.