Geometric control using the state-dependent Riccati equation: Application to aerial-acrobatic maneuvers

Saeed Rafee Nekoo*, José Ángel Acosta and Anibal Ollero
GRVC Robotics Lab., Depto de Ingeniería de Sistemas y Automática,
Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Seville, Spain
saerafee@yahoo.com, {jaar, aollero}@us.es
*Corresponding Author

Abstract

Acrobatic flip is one of the most challenging representatives of aggressive maneuvers to test the performance of an aerial system's capability or a controller. A variable-pitch rotor quadcopter generates thrust in both vertical directions for the special design of the rotor's actuation mechanism. Flip maneuver sets the rotation matrix of the copter in a singularity that makes the action impossible for common designs based on the hovering dynamics model. Sudden flip was a solution to pass the singularities though that imposes an agile maneuver and excessive energy consumption. This research proposes two possible solutions for the flip: a regulation solution based on the geometric control approach; and tracking a predefined optimal smooth trajectory covering a turnover. The first solution uses a geometric control approach that is immune to singular points since the rotation matrix is integrated into another manifold on $\mathrm{SO}(3)$. The second solution proposes an optimal trajectory generation for flip maneuver using open-loop optimal control, two-point boundary value problem (TPBVP) approach. Since generated open-loop state information is not applicable without a controller, the state-dependent differential Riccati equation (SDDRE) is chosen for trajectory tracking. The backward integration method is employed to provide a solution to the SDDRE with a final boundary condition. The dynamics equation of the variable pitch rotor was considered in the simulation to compute the angle of the blades and thrust coefficients. The flip maneuver has been successfully implemented and the optimal trajectory for the maneuver has been obtained and tracked by the nonlinear closed-loop optimal controller. For the regulation case, the geometric control resulted in a smoother flip rather than the abrupt turning of the conventional design.

Keywords: Quadcopter; variable-pitch; geometric control; open-loop optimal control; flip maneuver; SDDRE.

1. Introduction

Inspection and maintenance is the key point for the safety of industrial plants and refineries. The size of the refineries forces the designers to keep the infrastructure outdoor exposed to sun and rain, or in other words, degradation and corrosion. HYbrid FLying-rollIng with-snakE-aRm robot for contact inSpection (HYFLIERS) is an innovation and research action of EU Horizon 2020 program to inspect and perform maintenance via advanced robot capabilities (https://www.oulu.fi/hyfliers/). One branch of the research is devoted to an inspection task of the pipes by rotating around it with a multi-rotor drone. The drone keeps the inspection device near the pipe during the rotary motion. The common multi-rotor drones are working in hovering conditions due to the singularity of the rotation matrix for a large role angle. This current research presents a geometric design to control the system in an aggressive role angle, without the problem of singularity. The representative of this aggressive motion is a flip maneuver. So, the elementary step for solving the singularity problem for rotating around the pipe is followed in this current research. This work contributes to 1) setting up and formalizing the state-dependent Riccati equation equipped by geometric control; 2) presenting a TPBVP for quadrotor path-planning including flip maneuver. A comparative study analyzes the proposed methods in this aggressive flight.

The use of variable-pitch quadrotors has been increased in the research due to their capability in flight such as agile and aggressive maneuver, flip, and inverted flight (Pretorius & Boje, 2014). Agile and aggressive maneuvers always enforce the quadrotor to undergo sudden motions and excessive energy consumption; moreover, such maneuvers are not usually optimized with smooth and practical trajectory. Constrained input path generation and control for variable-pitch UAVs were presented (Cutler & How, 2012). An algorithm was designed to assign the flip in minimum time when the time was not mentioned in the acrobatic motion, and polynomials were designed to perform the flip in predefined trajectories. The flip maneuver was simulated for attitude control (Pretorius & Boje, 2014). Adaptive control was investigated based on quaternion design for package delivery by a variable-pitch unmanned system (Shastry, Bhargavapuri, Kothari, & Sahoo, 2018). A nonlinear adaptive controller was used to control the load uncertainty and agile flight. During the trajectory tracking in a circular motion, a flip was performed to illustrate the performance of the variable-pitch rotors. The flip was done fast to reduce the unstable situation during the acrobatic motion.

The sudden flip during the flight highlights the potential of the variable-pitch rotor unmanned aerial vehicle (UAV); however, unstable situations, power consumption, and possible actuator saturation are the disadvantages of the fast acrobatic maneuver which is not optimized. Moreover, the trajectory during the flip is unpredictable. This current research proposes two methods to solve this issue: a regulation method based on a geometric control approach and trajectory tracking of an optimal predefined path for the flip.

The first approach avoids singularity via geometric control, moving to another manifold in a special group in $SO(3)$. Although the geometric control was used for different cases such as airplane flight dynamics (Hassan & Taha, 2017), tilted-rotor UAV (Invernizzi & Lovera, 2017), haptic-virtual control (Hernández-Díez, Méndez-Barrios, Mondié, Niculescu, & González-Galván, 2018), UAV trajectory tracking (Shi, Zhang, & Zhou, 2015), and control of a UAV transporting a payload (F. A. Goodarzi, Lee, & Lee, 2015); the power of the method is more visible in performing acrobatic motions (F. Goodarzi, Lee, & Lee, 2013; Loianno, Brunner, McGrath, & Kumar, 2016; Raj, Banavar, & Kothari, 2017). The common form of the controller for the geometric approach is PD/PID plus a nonlinear part for compensating the rotational dynamics (Fernando, Chandiramani, Lee, & Gutierrez, 2011; Lee, 2012). An attitude tracking control was presented with the exponential stability on $SO(3)$ for large-scale rotational movements (Lee, 2012). The robustness and adaptive characteristics were mainly involved in the controllers though the dominating parts were the PD or PID terms (Gamagedara, Bisheban, Kaufman, & Lee, 2019). Here we introduce the state-dependent Riccati equation equipped with the geometric control to achieve a singularity-free nonlinear optimal controller for aggressive maneuvers.

For the second solution, open-loop optimal control is expressed for trajectory generation for a six-DoF fully coupled quadrotor considering flip maneuver. The flip time, initial, and desired position of the UAV are adjustable. The generated motion is smooth and admissible; also better to emphasize that the trajectory is optimal, a consequence of a quadratic performance criterion. The solution is found by solving a two-point boundary value problem. The trajectory generation, in the optimal domain, was investigated in different fields such as under-actuated pendulum (Boscariol & Richiedei, 2018), robotic manipulators for load-carrying (M. H. Korayem & Nikoobin, 2008), and hovercraft motion planning (Seguchi &

Ohtsuka, 2003); although, for the best knowledge of authors, optimal trajectory generation by TPBVP for flip maneuver of a quadrotor has not been studied.

Finding the optimal trajectory for the flip is half of the design and trajectory tracking by a closed-loop controller is essential to complete the task. The tracking controller must be effective to reduce the error for both the orientation and position of the system. The position and attitude of the system are coupled and this specific design removes the cascade control in closed-loop tracking. So, the incompatibility or lack of precision in one part (attitude or position) imposes a large error on the output. The state-dependent differential Riccati equation has been selected to track the trajectory. There are several methods to solve the SDDRE tracking problem though backward integration is the most precise method among them (M. H. Korayem & Nekoo, 2015a, 2015b). Backward integration provides a two-round solution; in the first round, the system sweeps the trajectory backward to generate the gains, and in the forward one, the system tracks the system with the obtained gain, forward in time (Beeler, Tran, & Banks, 2003; Nekoo, 2020). Approximate SDRE tracking control was presented considering the feed-forward tracking vector (Cimen, 2007). The design of a suboptimal tracking controller was studied for a class of nonlinear systems (Batmani, Davoodi, & Meskin, 2016). Backward integration might seem impractical because of the two-round solution; however, the different applications showed otherwise such as satellite control (Abdelrahman & Park, 2013), robotics (A. H. Korayem, Nekoo, & Korayem, 2019), multi-motor driving unit (Wang, Dong, Ren, & Chen, 2020), and hardware-in-the-loop satellite control (Jung, Park, Kim, Eun, & Chang, 2013).

The main contributions: (a) introducing the geometric control embedded within the nonlinear optimal control problem, the SDRE, to release a singularity-free method for acrobatic and aggressive maneuver (including flip) for variable-pitch quadcopters. (b) Proposing an open-loop optimal control approach for path planning of a quadrotor in six-DoF considering flip maneuver with adjustable maneuver time. Then tracking the optimal trajectory with the SDDRE controller using the backward integration solution. The sudden flip (in regulation) for a quadrotor using SDRE was reported (Nekoo, Acosta, Gomez-Tamm, & Ollero, 2019), though the novelty of this current work is highlighted by the smoothness of the optimal trajectory for the flip, defining the desired time of the flip, and the differential form of the tracking controller.

Section 2 expresses the dynamics of a variable-pitch rotor quadcopter. Section 3 presents the geometric approach for the flip maneuver using the SDRE design. Section 4 presents the open-loop optimal control trajectory generation by TPBVP. Section 5 is dedicated to the SDDRE tracking control design. Section 6 illustrates the simulations and Section 7 states the conclusion.

2. Variable-pitch quadrotor dynamics

The fixed (inertial or Earth) frame is the first reference frame. The Z axis points from the center of Earth towards the surface, Y pointing to North and X pointing to East, see Fig. 1. The body frame is set on to the center-of-mass (CoM) of a UAV, presented in Fig. 2; the position variables are $\boldsymbol{\xi}_1 = [x_\mathrm{c}, y_\mathrm{c}, z_\mathrm{c}]^T$(m), and Euler angles in the inertial frame, roll-pitch-yaw, are $\boldsymbol{\xi}_2 = [\phi, \theta, \psi]^T$(rad). The derivatives of generalized coordinates $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are named as linear velocity vector in the inertial frame $\dot{\boldsymbol{\xi}}_1 = [\dot{x}_\mathrm{c}, \dot{y}_\mathrm{c}, \dot{z}_\mathrm{c}]^T$(m/s), and angular velocity vector in the inertial frame $\dot{\boldsymbol{\xi}}_2 = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$(rad/s).

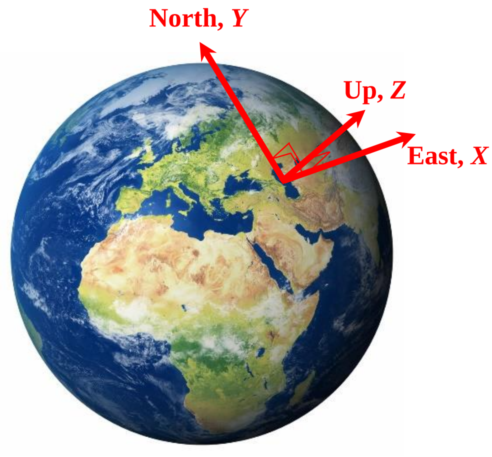

Fig. 1. Fixed reference coordinate on earth.

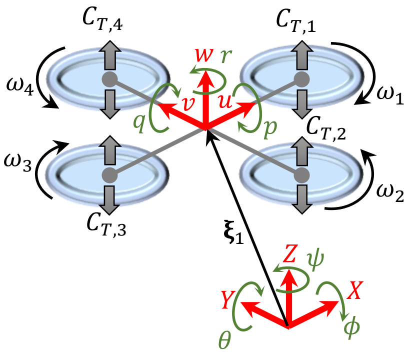

Fig. 2. The schematic view of the variable-pitch quadcopter.

The total thrust T_{B} (N) is aligned with z_{c} on the CoM of the UAV. The input torque vector is $\boldsymbol{\tau}_{\mathrm{B}} = [\tau_{\phi} \quad \tau_{\theta} \quad \tau_{\psi}]^T$, (Nm); its direction is against Euler angles $\{\phi, \theta, \psi\}$. T_{B} is set on body frame, and $\boldsymbol{\tau}_{\mathrm{B}}$ acts on the inertial frame. To make the coordinates identical, the input force is transformed from the body coordinate to the inertial one:

$$\mathbf{F}(\boldsymbol{\xi}_2) = \mathbf{R}_{ZYX}(\boldsymbol{\xi}_2)\begin{bmatrix} 0 \\ 0 \\ T_{\mathrm{B}} \end{bmatrix}, \tag{1}$$

where

$$\mathbf{R}_{ZYX}(\boldsymbol{\xi}_2) = \begin{bmatrix} c_{\psi}c_{\theta} & -c_{\phi}s_{\psi} + s_{\phi}s_{\theta}c_{\psi} & s_{\phi}s_{\psi} + c_{\phi}s_{\theta}c_{\psi} \\ s_{\psi}c_{\theta} & c_{\phi}c_{\psi} + s_{\phi}s_{\theta}s_{\psi} & -s_{\phi}c_{\psi} + c_{\phi}s_{\theta}s_{\psi} \\ -s_{\theta} & c_{\theta}s_{\phi} & c_{\theta}c_{\phi} \end{bmatrix}.$$

The purpose of using two rotational matrices in this work is to compare the geometric control with conventional methods for aggressive maneuvers. It should be noted that $\mathbf{R}_{ZYX}(\boldsymbol{\xi}_2(t))\colon \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ in Eq. (1) is only used for TPBVP (Section 4 and 5). For the geometric control (Section 3), a new numerical $\mathbf{R}(t) \in \mathrm{SO}(3)$ is used:

$$\begin{gathered} \mathrm{SO}(3) = \{\mathbf{R}(t) \in \mathbb{R}^{3\times 3} | \mathbf{R}^T(t)\mathbf{R}(t) = \mathbf{I}_{3\times 3}, \det[\mathbf{R}(t)] = 1\}, \\ \dot{\mathbf{R}}(t) = \mathbf{R}(t)\widehat{\boldsymbol{\Omega}}(t), \end{gathered} \tag{2}$$

where [*hat*] map $\widehat{(\cdot)}\colon \mathbb{R}^3 \to \mathfrak{so}(3)$ changes any vector to a skew-symmetric matrix:

$$\widehat{\boldsymbol{\Omega}}(t) = \begin{bmatrix} 0 & -\dot{\psi}(t) & \dot{\theta}(t) \\ \dot{\psi}(t) & 0 & -\dot{\phi}(t) \\ -\dot{\theta}(t) & \dot{\phi}(t) & 0 \end{bmatrix}.$$

$\mathbf{R}_{ZYX}(\boldsymbol{\xi}_2(t))$, in Eq. (1), is the common form of a rotation matrix used in kinematics and dynamics of the flying systems, found by three consecutive rotations on an orthogonal coordinate. To avoid the singularities in $\mathbf{R}_{ZYX}(\boldsymbol{\xi}_2(t))$, a new rotation matrix is used $\mathbf{R}(t)$, with specifications in Eq. (2), which results in geometric control.

Using the Euler-Lagrange equation, the dynamics equation of the system is found, which the translation section provides:

$$m\ddot{\boldsymbol{\xi}}_1 + m\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} s_{\phi}s_{\psi} + c_{\phi}s_{\theta}c_{\psi} \\ -s_{\phi}c_{\psi} + c_{\phi}s_{\theta}s_{\psi} \\ c_{\theta}c_{\phi} \end{bmatrix} T_{\mathrm{B}}, \tag{3}$$

where m(kg) represents the total mass of the system and g(m/s^2) indicates gravity acceleration. The orientation dynamic results in:

$$\mathbf{J}(\boldsymbol{\xi}_2)\ddot{\boldsymbol{\xi}}_2 + \mathbf{C}(\boldsymbol{\xi}_2, \dot{\boldsymbol{\xi}}_2)\dot{\boldsymbol{\xi}}_2 = \boldsymbol{\tau}_{\mathrm{B}}. \tag{4}$$

Generalized coordinates and their velocities are set in the state vector

$$\mathbf{x} = [\boldsymbol{\xi}_1^T, \boldsymbol{\xi}_2^T, \dot{\boldsymbol{\xi}}_1^T, \dot{\boldsymbol{\xi}}_2^T]^T = [x_{\mathrm{c}}, y_{\mathrm{c}}, z_{\mathrm{c}}, \phi, \theta, \psi, \dot{x}_{\mathrm{c}}, \dot{y}_{\mathrm{c}}, \dot{z}_{\mathrm{c}}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T. \tag{5}$$

Considering state-vector (5), and equation of motion (3) and (4), the state-space equation of the UAV is found:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\boldsymbol{\xi}}_1 \\ \dot{\boldsymbol{\xi}}_2 \\ \ddot{\boldsymbol{\xi}}_1 \\ \ddot{\boldsymbol{\xi}}_2 \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{\xi}}_1 \\ \dot{\boldsymbol{\xi}}_2 \\ 1/m\mathbf{I}_{3\times3}\left[\mathbf{R}_{ZYX,3}(\boldsymbol{\xi}_2)T_{\mathrm{B}} - mg\mathbf{e}_3 - \mathbf{D}\dot{\boldsymbol{\xi}}_1\right] \\ \mathbf{J}^{-1}(\boldsymbol{\xi}_2)\left[\boldsymbol{\tau}_{\mathrm{B}} - \mathbf{C}(\boldsymbol{\xi}_2, \dot{\boldsymbol{\xi}}_2)\dot{\boldsymbol{\xi}}_2\right] \end{bmatrix}, \tag{6}$$

in which $\mathbf{e}_3 = [0,0,1]^T$, $\mathbf{R}_{ZYX,3}(\boldsymbol{\xi}_2)$ represents the last column of $\mathbf{R}_{ZYX}(\boldsymbol{\xi}_2)$, and details of $\mathbf{C}(\boldsymbol{\xi}_2, \dot{\boldsymbol{\xi}}_2)$ could be found in Ref. (Nekoo, Acosta, Gomez-Tamm, et al., 2019).

The aerodynamics effect is regarded in UAV dynamics (6) by definition of $\mathbf{D} = \mathrm{diag}(D_x, D_y, D_z)$ (kg/s) (Luukkonen, 2011). The corresponding state-space system for geometric control is also presented with a different rotation matrix:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\boldsymbol{\xi}}_1 \\ \dot{\boldsymbol{\xi}}_2 \\ \ddot{\boldsymbol{\xi}}_1 \\ \ddot{\boldsymbol{\xi}}_2 \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{\xi}}_1 \\ \dot{\boldsymbol{\xi}}_2 \\ 1/m\mathbf{I}_{3\times3}\left[\mathbf{R}_3 T_{\mathrm{B}} - mg\mathbf{e}_3 - \mathbf{D}\dot{\boldsymbol{\xi}}_1\right] \\ \mathbf{J}^{-1}(\boldsymbol{\xi}_2)\left[\boldsymbol{\tau}_{\mathrm{B}} - \mathbf{C}(\boldsymbol{\xi}_2, \dot{\boldsymbol{\xi}}_2)\dot{\boldsymbol{\xi}}_2\right] \end{bmatrix}, \tag{7}$$

in which $\mathbf{R}_3(t)$ presents the last column of $\mathbf{R}(t)$.

The dynamics of the quadrotor is balanced by generalized force including total thrust T_{B} and moments $\boldsymbol{\tau}_{\mathrm{B}}$. The state-space representation of the system is nonlinear (either of Eqs. (6) or (7)), coupled and under-actuated. The computation of dynamics and control is done by the generalized force-moments. The blade angle and angular velocity of four rotors of the quadcopter are related to the generalized force-moment through a kinematic relation. Following definitions are considered to find the thrust factors $\mathbf{C}_{\mathrm{T}}(t)$:

1) **Thrust** $T_{\mathrm{B}}(t)$: If all the blades have the same angle, the force $T_{\mathrm{B}}(t)$ (in body frame) is generated. An increase in blade angle (positive direction) moves the quadrotor up and a decrease towards down. If the blade angle sets in the negative direction, the negative thrust is generated. A specific constant blade angle holds the quadrotor steady in the air.
2) **Roll** $\tau_{\phi}(t)$: If the blade angles of rotor 2 and 4 differ, a rotation around x_{c} coupled with lateral motion occurs, represented by the reactive moment $\tau_{\phi}(t)$.

3) **Pitch** $\tau_\theta(t)$: If the blade angles of rotor 1 and 3 differ, a rotation around y_{c} coupled with lateral motion occurs, represented by the reactive moment $\tau_\theta(t)$.
4) **Yaw** $\tau_\psi(t)$: If a differential blade angle sets between rotors (1,3) and (2,4), a rotation around z_{c} happens, represented by the reactive moment $\tau_\psi(t)$. The pair of blade angles (1,3) changes with the same angle and the pair of blade angles (2,4) changes with the same one in the opposite direction. The difference between the two pairs provides yaw.

The total thrust is (Bhargavapuri, Sahoo, & Kothari, 2019):

$$T_{\mathrm{B}} = K\sum_{i=1}^{4} C_{T_i}, \tag{8}$$

where $K = \rho\pi R^4\omega_{\mathrm{ss}}^2$, $\rho(\mathrm{kg/m^3})$ is the air density, $R(\mathrm{m})$ is the tip radius of the blade, $\omega_{\mathrm{ss}}(\mathrm{rad/s})$ is the steady-state angular velocity of the blades, and $C_{T_i}(t)$ indicates a coefficient of the thrust of i-th variable-pitch blade. ω_{ss} can be defined more than the minimum angular velocity of the quadcopter; the minimum value for flying is $\omega_{\mathrm{min}} = \sqrt{mg/(4k)}$ $(\mathrm{rad/s})$, in which $k(\mathrm{Ns^2/rad^2})$ is the lift constant for a special steady-state blade angle. Roll and pitch moments are

$$\tau_\phi = KL\left[-C_{T_2} + C_{T_4}\right], \tag{9}$$

$$\tau_\theta = KL\left[-C_{T_1} + C_{T_3}\right], \tag{10}$$

where L is the distance between a rotor and CoM of the quadrotor. Yaw moment is

$$\tau_\psi = \frac{RK}{\sqrt{2}}\left[-\left|C_{T_1}\right|^{3/2} + \left|C_{T_2}\right|^{3/2} - \left|C_{T_3}\right|^{3/2} + \left|C_{T_4}\right|^{3/2}\right]. \tag{11}$$

Equations (8)-(11) form a nonlinear relation between input force/moments and thrust coefficients:

$$\begin{bmatrix} T_{\mathrm{B}} \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} K & K & K & K \\ 0 & -LK & 0 & LK \\ -LK & 0 & LK & 0 \\ -\frac{RK}{\sqrt{2}}\sqrt{\left|C_{T_1}\right|} & \frac{RK}{\sqrt{2}}\sqrt{\left|C_{T_2}\right|} & -\frac{RK}{\sqrt{2}}\sqrt{\left|C_{T_3}\right|} & \frac{RK}{\sqrt{2}}\sqrt{\left|C_{T_4}\right|} \end{bmatrix} \begin{bmatrix} C_{T_1} \\ C_{T_2} \\ C_{T_3} \\ C_{T_4} \end{bmatrix}. \tag{12}$$

Several methods were presented to solve this nonlinear equation such as using first-order dynamics (Bhargavapuri et al., 2019), Mean Value Theorem, Pseudo, augmented Pseudo inverse approach, and null-space of Pseudo inverse approach (Nekoo, Acosta, Gomez-Tamm, et al., 2019). In this work, to solve the nonlinear relation in thrust allocation which leads to a system with control nonlinearity, the augmented Pseudo inverse approach is used (Section VII-C, Ref. (Nekoo, Acosta, Gomez-Tamm, et al., 2019)). First, the linear and nonlinear parts of (12) are separated:

$$\boldsymbol{\zeta} = \begin{bmatrix} T_{\mathrm{B}} \\ \tau_{\phi} \\ \tau_{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} K & K & K & K \\ 0 & -LK & 0 & LK \\ -LK & 0 & LK & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} C_{T_1} \\ C_{T_2} \\ C_{T_3} \\ C_{T_4} \end{bmatrix}, \tag{13}$$

$$\frac{\sqrt{2}\tau_{\psi}}{RK} = \underbrace{\left[-\sqrt{|C_{T_1}|} \quad \sqrt{|C_{T_2}|} \quad -\sqrt{|C_{T_3}|} \quad \sqrt{|C_{T_4}|}\right]}_{\mathbf{a}^{T}(\mathbf{C}_{\mathrm{T}})} \mathbf{C}_{\mathrm{T}} = \mathbf{f}(\mathbf{C}_{\mathrm{T}}). \tag{14}$$

Next, the optimization problem is defined by:

$$H_{\mathrm{F}}(\mathbf{C}_{\mathrm{T}}, \lambda) = \frac{1}{2}\left(\mathbf{C}_{\mathrm{T}}^{T}\mathbf{W}\mathbf{C}_{\mathrm{T}} + W_{\mathrm{Yaw}}\left\{\frac{\sqrt{2}\tau_{\psi}}{RK} - \mathbf{f}(\mathbf{C}_{\mathrm{T}})\right\}^{2}\right) - \boldsymbol{\lambda}^{T}[\mathbf{A}\mathbf{C}_{\mathrm{T}} - \boldsymbol{\zeta}], \tag{15}$$

where $W_{\mathrm{Yaw}}\frac{\sqrt{2}\tau_{\psi}}{RK} - \mathbf{f}(\mathbf{C}_{\mathrm{T}})$ is a mild constraint in the optimization, $\mathbf{W}$ is a weighting matrix, W_{Yaw} is a weighting parameter for highlighting/weakening the effect of nonlinearity in the optimization process, and the aim is to find the best value of $\mathbf{C}_{\mathrm{T}}$. Since all the elements of $\mathbf{C}_{\mathrm{T}}$ have the same importance, $\mathbf{W}$ is set as an identity matrix. W_{Yaw} is also set 1 to share the same weight for linear and nonlinear parts Eqs. (13) and (14). The time derivative of the cost function (15) with respect to $\mathbf{C}_{\mathrm{T}}$ and equating that with zero, result in

$$\frac{\partial H_{\mathrm{F}}(\mathbf{C}_{\mathrm{T}}, \lambda)}{\partial \mathbf{C}_{\mathrm{T}}} = \mathbf{W}\mathbf{C}_{\mathrm{T}} - W_{\mathrm{Yaw}}\left\{\frac{\sqrt{2}\tau_{\psi}}{RK} - \mathbf{f}(\mathbf{C}_{\mathrm{T}})\right\}\frac{\partial \mathbf{f}(\mathbf{C}_{\mathrm{T}})}{\partial \mathbf{C}_{\mathrm{T}}} - \mathbf{A}^{T}\boldsymbol{\lambda} = \mathbf{0}. \tag{16}$$

We rewrite the term

$$\frac{\partial \mathbf{f}(\mathbf{C}_{\mathrm{T}})}{\partial \mathbf{C}_{\mathrm{T}}} = \mathbf{F}(\mathbf{C}_{\mathrm{T}})\mathbf{C}_{\mathrm{T}}, \tag{17}$$

in which

$$\mathbf{F}(\mathbf{C}_{\mathrm{T}}) = \frac{3}{2}\mathrm{diag}\left(-\frac{\mathrm{sign}C_{T_1}}{\sqrt{|C_{T_1}|}}, \frac{\mathrm{sign}C_{T_2}}{\sqrt{|C_{T_2}|}}, -\frac{\mathrm{sign}C_{T_3}}{\sqrt{|C_{T_3}|}}, \frac{\mathrm{sign}C_{T_4}}{\sqrt{|C_{T_4}|}}\right).$$

Substituting (17) into (16) and extracting $\mathbf{C}_{\mathrm{T}}$, result in

$$\mathbf{C}_{\mathrm{T}} = \left(\mathbf{W} - W_{\mathrm{Yaw}}\left\{\frac{\sqrt{2}\tau_{\psi}}{RK} - \mathbf{f}(\mathbf{C}_{\mathrm{T}})\right\}\mathbf{F}(\mathbf{C}_{\mathrm{T}})\right)^{-1}\mathbf{A}^{T}\boldsymbol{\lambda}. \tag{18}$$

Substituting (18) into (13) also provides

$$\boldsymbol{\zeta} = \mathbf{A}\boldsymbol{\Lambda}^{-1}(\mathbf{C}_\mathrm{T})\mathbf{A}^T\boldsymbol{\lambda} \Rightarrow \boldsymbol{\lambda} = (\mathbf{A}\boldsymbol{\Lambda}^{-1}(\mathbf{C}_\mathrm{T})\mathbf{A}^T)^{-1}\boldsymbol{\zeta}, \tag{19}$$

where $\boldsymbol{\Lambda}(\mathbf{C}_\mathrm{T}) = \mathbf{W} - W_\mathrm{Yaw}\left\{\frac{\sqrt{2}\tau_\psi}{RK} - \mathbf{f}(\mathbf{C}_\mathrm{T})\right\}\mathbf{F}(\mathbf{C}_\mathrm{T})$. Finally, inserting $\boldsymbol{\lambda}$, (19), into (18) results in:

$$\mathbf{C}_\mathrm{T} = \boldsymbol{\Lambda}^{-1}(\mathbf{C}_\mathrm{T})\mathbf{A}^T(\mathbf{A}\boldsymbol{\Lambda}^{-1}(\mathbf{C}_\mathrm{T})\mathbf{A}^T)^{-1}\boldsymbol{\zeta}. \tag{20}$$

In simulation or experiment, each time step is represented by i within the timespan $t(i) \in [0, t_\mathrm{f}]$ in which $i \in [1, N], N \in \mathbb{Z}$. Concerning the time steps and applying Mean Value Theorem, the thrust coefficients are (Nekoo, Acosta, Gomez-Tamm, et al., 2019):

$$\mathbf{C}_\mathrm{T}(i) = \boldsymbol{\Lambda}^{-1}\big(\mathbf{C}_\mathrm{T}(i-1)\big)\mathbf{A}^T\big(\mathbf{A}\boldsymbol{\Lambda}^{-1}\big(\mathbf{C}_\mathrm{T}(i-1)\big)\mathbf{A}^T\big)^{-1}\boldsymbol{\zeta}(i).$$

So, the thrust coefficients are updated based on the previous loop in the implementation process. The blade angles in terms of thrust coefficients are defined (Bhargavapuri et al., 2019):

$$\boldsymbol{\alpha}(t) = \frac{3}{2}\lambda + \frac{6}{\sigma C_{l_\alpha}}\mathbf{C}_\mathrm{T}(t), \tag{21}$$

where (for each rotor and its blade) $\sigma = N_b c/(\pi R)$, N_b is the number of blades, C_{l_α} is airfoil lift curve slope, $c(\mathrm{m})$ is chord length, and $\lambda = v_0/(\omega_\mathrm{ss}R)$ is the inflow ratio and $v_0(\mathrm{m/s})$ is airflow velocity. Substituting (12) in (21) and discretization result in:

$$\boldsymbol{\alpha}(i) = \frac{6}{\sigma C_{l_\alpha}K}\Big(\boldsymbol{\Lambda}^{-1}\big(\mathbf{C}_\mathrm{T}(i-1)\big)\mathbf{A}^T\big(\mathbf{A}\boldsymbol{\Lambda}^{-1}\big(\mathbf{C}_\mathrm{T}(i-1)\big)\mathbf{A}^T\big)^{-1}\boldsymbol{\zeta}(i)\Big) + \frac{3}{2}\lambda,$$

which is limited by the input constraints $\alpha_\mathrm{min} \leq \alpha_i(t) \leq \alpha_\mathrm{max}$ for $i = 1, \dots, 4$ where α_min and α_max are the minimum and maximum blade angles. In practical implementation, $\alpha_i(t)$ will be commanded to the servomotors for changing the blade angles. In simulation, $\alpha_i(t)$ will reconstruct $\mathbf{C}_\mathrm{T}(i) = \frac{\sigma C_{l_\alpha}}{2}\left(\frac{\boldsymbol{\alpha}(i)}{3} - \frac{\lambda}{2}\right)$, then it will be substituted into (12) for rebuilding the input force/moment vector for solving the state-space model (6) or (7), at time-step i.

3. Geometric control approach: A regulation problem for flip

The geometric control is applied for a nonlinear system (7), divided into two translation and orientation parts:

$$\dot{\mathbf{x}}_\mathrm{t} = \underbrace{\begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & -m\mathbf{I}_{3\times3}\mathbf{D}\end{bmatrix}}_{\mathbf{A}_\mathrm{t}}\begin{bmatrix}\boldsymbol{\xi}_1 \\ \dot{\boldsymbol{\xi}}_1\end{bmatrix} + \underbrace{\begin{bmatrix}\mathbf{0}_{3\times3} \\ m\mathbf{I}_{3\times3}\end{bmatrix}}_{\mathbf{B}_\mathrm{t}}\mathbf{u}_\mathrm{t}, \tag{22}$$

$$\dot{\mathbf{x}}_{\mathrm{o}} = \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \\ \mathbf{0}_{3\times 3} & -\mathbf{J}^{-1}(\mathbf{x}_{\mathrm{o}})\mathbf{C}(\mathbf{x}_{\mathrm{o}}) \end{bmatrix}}_{\mathbf{A}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}(t))} \begin{bmatrix} \boldsymbol{\xi}_2 \\ \dot{\boldsymbol{\xi}}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{J}^{-1}(\mathbf{x}_{\mathrm{o}}) \end{bmatrix}}_{\mathbf{B}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}(t))} \mathbf{u}_{\mathrm{o}}, \qquad (23)$$

in which control laws are defined

$$\mathbf{u}_{\mathrm{t}} = -\mathbf{R}_{\mathrm{t}}^{-1}\mathbf{B}_{\mathrm{t}}^{T}\mathbf{K}_{\mathrm{t}} \begin{bmatrix} \boldsymbol{\xi}_1 - \boldsymbol{\xi}_{1,\mathrm{des}} \\ \dot{\boldsymbol{\xi}}_1 - \dot{\boldsymbol{\xi}}_{1,\mathrm{des}} \end{bmatrix}, \qquad (24)$$

$$\mathbf{u}_{\mathrm{o}} = -\mathbf{R}_{\mathrm{o}}^{-1}(\mathbf{x}_{\mathrm{o}})\mathbf{B}_{\mathrm{o}}^{T}(\mathbf{x}_{\mathrm{o}})\mathbf{K}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}) \begin{bmatrix} \mathbf{e}_{\mathrm{R}} \\ \mathbf{e}_{\Omega} \end{bmatrix}. \qquad (25)$$

The feedback vector of the geometric control approach in Eq. (25) is $\begin{bmatrix} \mathbf{e}_{\mathrm{R}} \\ \mathbf{e}_{\Omega} \end{bmatrix}$, different from conventional form $\begin{bmatrix} \boldsymbol{\xi}_2 \\ \dot{\boldsymbol{\xi}}_2 \end{bmatrix}$. $\mathbf{e}_{\mathrm{R}}$ is derived from geometric control that includes $\boldsymbol{\xi}_2$ within rotation matrix $\mathbf{R}(t)$. In other words, the new feedback is forcing the generalized coordinates of orientation $\boldsymbol{\xi}_2$ to desired values in another geometry or manifold. The new concept prefers to regulate the total matrix to the desired matrix rather than the regulation of $\boldsymbol{\xi}_2$ directly. $\mathbf{e}_{\Omega}$ also compares the angular velocity $\dot{\boldsymbol{\xi}}_2$ with desired ones in another geometry. The detail is presented in this section.

$\boldsymbol{\xi}_{1,\mathrm{des}}$ and $\dot{\boldsymbol{\xi}}_{1,\mathrm{des}}$, in Eq. (24), are the desired linear position and velocity of the quadcopter. The weighting matrices are set in the cost functional integrals

$$J_{\mathrm{t}} = \frac{1}{2}\int_{0}^{\infty} \{\mathbf{u}_{\mathrm{t}}^{T}\mathbf{R}_{\mathrm{t}}\mathbf{u}_{\mathrm{t}} + \mathbf{x}_{\mathrm{t}}^{T}\mathbf{Q}_{\mathrm{t}}\mathbf{x}_{\mathrm{t}}\}\mathrm{d}t,$$

$$J_{\mathrm{o}} = \frac{1}{2}\int_{0}^{\infty} \{\mathbf{u}_{\mathrm{o}}^{T}\mathbf{R}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}})\mathbf{u}_{\mathrm{o}} + \mathbf{x}_{\mathrm{o}}^{T}\mathbf{Q}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}})\mathbf{x}_{\mathrm{o}}\}\mathrm{d}t,$$

where weighting matrices for states $\mathbf{Q}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}): \mathbb{R}^{6} \to \mathbb{R}^{6\times 6}$ and $\mathbf{Q}_{\mathrm{t}} \in \mathbb{R}^{6\times 6}$ are symmetric positive semi-definite and $\mathbf{R}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}): \mathbb{R}^{3} \to \mathbb{R}^{3\times 3}$ and $\mathbf{R}_{\mathrm{t}} \in \mathbb{R}^{3\times 3}$ are the ones for inputs, symmetric positive definite. The suboptimal gains $\mathbf{K}_{\mathrm{t}} \in \mathbb{R}^{6\times 6}$ and $\mathbf{K}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}): \mathbb{R}^{6} \to \mathbb{R}^{6\times 6}$ are solutions to the linear quadratic regulator and the state-dependent Riccati equation

$$\mathbf{K}_{\mathrm{t}}\mathbf{A}_{\mathrm{t}} + \mathbf{A}_{\mathrm{t}}^{T}\mathbf{K}_{\mathrm{t}} + \mathbf{Q}_{\mathrm{t}} - \mathbf{K}_{\mathrm{t}}\mathbf{B}_{\mathrm{t}}\mathbf{R}_{\mathrm{t}}^{-1}\mathbf{B}_{\mathrm{t}}^{T}\mathbf{K}_{\mathrm{t}} = \mathbf{0}, \qquad (26)$$

$$\begin{aligned} &\mathbf{K}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}})\mathbf{A}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}) + \mathbf{A}_{\mathrm{o}}^{T}(\mathbf{x}_{\mathrm{o}})\mathbf{K}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}) + \mathbf{Q}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}) \\ &\quad - \mathbf{K}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}})\mathbf{B}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}})\mathbf{R}_{\mathrm{o}}^{-1}(\mathbf{x}_{\mathrm{o}})\mathbf{B}_{\mathrm{o}}^{T}(\mathbf{x}_{\mathrm{o}})\mathbf{K}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}}) = \mathbf{0}. \end{aligned} \qquad (27)$$

Since the translation and orientation dynamics are controlled separately (in a cascade design) two Riccati equations must be solved. All the matrices of the first one, Eq. (26), are constant which leads to a linear quadratic regulator with constant gain $\mathbf{K}_{\mathrm{t}}$. The second part includes nonlinear matrices that result in the SDRE (27) and the nonlinear gain $\mathbf{K}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}})$.

Since the state-dependent coefficient (SDC) system design is a linear parameterization in translation dynamics (22), the SDRE is turned into a linear quadratic regulator design, see Eq. (26). Nonlinear matrices in (23) preserve the nonlinear shape of the SDRE (27) and provide a nonlinear optimal gain $\mathbf{K}_{\mathrm{o}}(\mathbf{x}_{\mathrm{o}})$. Control law (24) builds the total thrust for system equation (7) (Nekoo, Acosta, Gomez-Tamm, et al., 2019):

$$T_{\mathrm{B}}(t)=m\{[\mathbf{R}(t)]_{3,1}u_{\mathrm{t},1}(t)+[\mathbf{R}(t)]_{3,2}u_{\mathrm{t},2}(t)+[\mathbf{R}(t)]_{3,3}\big(u_{\mathrm{t},3}(t)+g\big)\}.$$

To define the error vector for rotational dynamics in (25), an error function is chosen (Lee, 2012):

$$\Psi\big(\mathbf{R}(t),\mathbf{R}_{\mathrm{d}}(t)\big)=2-\sqrt{\mathrm{tr}\big[\mathbf{R}_{\mathrm{d}}^{T}(t)\mathbf{R}(t)\big]+1}, \tag{28}$$

where tr[·] is trace operator, $\mathbf{R}(t)$ is the rotation matrix, which has been defined in Eq. (2), to be computed by numerical integration, and the desired one is $\mathbf{R}_{\mathrm{d}}(t)$. Variation of error function (28) is:

$$\delta_{\mathbf{R}(t)}\Psi\big(\mathbf{R}(t),\mathbf{R}_{\mathrm{d}}(t)\big)=\frac{\partial\Psi\big(\mathbf{R}(t),\mathbf{R}_{\mathrm{d}}(t)\big)}{\partial\mathbf{R}(t)}\delta\mathbf{R}(t),$$

where $\delta\mathbf{R}(t)=\dot{\mathbf{R}}(t)=\mathbf{R}(t)\widehat{\mathbf{\Omega}}(t)$ holds, that provides the derivative of the error function

$$\delta_{\mathbf{R}(t)}\Psi\big(\mathbf{R}(t),\mathbf{R}_{\mathrm{d}}(t)\big)=-\frac{\mathrm{tr}\big[\mathbf{R}_{\mathrm{d}}^{T}(t)\mathbf{R}(t)\widehat{\mathbf{\Omega}}(t)\big]}{2\sqrt{\mathrm{tr}\big[\mathbf{R}_{\mathrm{d}}^{T}(t)\mathbf{R}(t)\big]+1}}. \tag{29}$$

Considering that $\mathrm{tr}[\mathbf{A}\hat{\mathbf{x}}]=-(\mathbf{A}-\mathbf{A}^{T})^{\vee}\mathbf{x}$ holds (Zhu, Chen, & Li, 2017); for arbitrary $\mathbf{A}$ and $\mathbf{x}$ where *vee* map is defined as $(\cdot)^{\vee}:\mathfrak{so}(3)\rightarrow\mathbb{R}^{3}$, Eq. (29) is rewritten as:

$$\delta_{\mathbf{R}(t)}\Psi\big(\mathbf{R}(t),\mathbf{R}_{\mathrm{d}}(t)\big)=\frac{\big[\mathbf{R}_{\mathrm{d}}^{T}(t)\mathbf{R}(t)-\mathbf{R}^{T}(t)\mathbf{R}_{\mathrm{d}}(t)\big]^{\vee}}{2\sqrt{\mathrm{tr}\big[\mathbf{R}_{\mathrm{d}}^{T}(t)\mathbf{R}(t)\big]+1}}\mathbf{\Omega}(t).$$

Consequently, the orientation error vector is selected as

$$\mathbf{e}_{\mathrm{R}}(t)=\frac{\big[\mathbf{R}_{\mathrm{d}}^{T}(t)\mathbf{R}(t)-\mathbf{R}^{T}(t)\mathbf{R}_{\mathrm{d}}(t)\big]^{\vee}}{2\sqrt{\mathrm{tr}\big[\mathbf{R}_{\mathrm{d}}^{T}(t)\mathbf{R}(t)\big]+1}}.$$

The error vector has a second part, $\mathbf{e}_{\Omega}(t)$. To find that part, the time derivative of the rotation matrix $\mathbf{R}(t)$ should be compared with the desired rotation matrix $\mathbf{R}_{\mathrm{d}}(t)$, though they should be lying in the same tangent space (Lee, 2012):

$$\dot{\mathbf{R}}(t)-\dot{\mathbf{R}}_{\mathrm{d}}^{T}(t)\left[\mathbf{R}_{\mathrm{d}}^{T}(t)\mathbf{R}(t)\right]=\mathbf{R}(t)[\boldsymbol{\Omega}(t)-\mathbf{R}^{T}(t)\mathbf{R}_{\mathrm{d}}(t)\boldsymbol{\Omega}_{\mathrm{d}}(t)]^{\wedge}.$$

The velocity section of error of the orientation dynamics in geometric form is also selected as:

$$\mathbf{e}_{\Omega}(t)=\boldsymbol{\Omega}(t)-\mathbf{R}^{T}(t)\mathbf{R}_{\mathrm{d}}(t)\boldsymbol{\Omega}_{\mathrm{d}}(t).$$

The desired rotation matrix is

$$\mathbf{R}_{\mathrm{d}}(t)=\begin{bmatrix} c_{\psi_{\mathrm{des}}}c_{\theta_{\mathrm{des}}} & -c_{\phi_{\mathrm{des}}}s_{\psi_{\mathrm{des}}}+s_{\phi_{\mathrm{des}}}s_{\theta_{\mathrm{des}}}c_{\psi_{\mathrm{des}}} & s_{\phi_{\mathrm{des}}}s_{\psi_{\mathrm{des}}}+c_{\phi_{\mathrm{des}}}s_{\theta_{\mathrm{des}}}c_{\psi_{\mathrm{des}}} \\ s_{\psi_{\mathrm{des}}}c_{\theta_{\mathrm{des}}} & c_{\phi_{\mathrm{des}}}c_{\psi_{\mathrm{des}}}+s_{\phi_{\mathrm{des}}}s_{\theta_{\mathrm{des}}}s_{\psi_{\mathrm{des}}} & -s_{\phi_{\mathrm{des}}}c_{\psi_{\mathrm{des}}}+c_{\phi_{\mathrm{des}}}s_{\theta_{\mathrm{des}}}s_{\psi_{\mathrm{des}}} \\ -s_{\theta_{\mathrm{des}}} & c_{\theta_{\mathrm{des}}}s_{\phi_{\mathrm{des}}} & c_{\theta_{\mathrm{des}}}c_{\phi_{\mathrm{des}}} \end{bmatrix},$$

where ϕ_{des} and θ_{des} are as follows and ψ_{des} could be set independently (Nekoo, Acosta, & Ollero, 2019):

$$\theta_{\mathrm{des}}(t)=\tan^{-1}\left(\frac{u_{\mathrm{t},1}\cos\psi_{\mathrm{des}}+u_{\mathrm{t},2}\sin\psi_{\mathrm{des}}}{u_{\mathrm{t},3}+g}\right),$$

$$\phi_{\mathrm{des,f}}(t)=\sin^{-1}\left(\frac{u_{\mathrm{t},1}\sin\psi_{\mathrm{des}}-u_{\mathrm{t},2}\cos\psi_{\mathrm{des}}}{\sqrt{u_{\mathrm{t},1}^{2}+u_{\mathrm{t},2}^{2}+\left(u_{\mathrm{t},3}+g\right)^{2}}}\right),$$

where $u_{\mathrm{t},i}$ is found from Eq. (24). For performing the flip maneuver at the time $0<t_1<t_{\mathrm{f}}$ with the desired time-span of the flip $t_{\mathrm{flip}}=t_2-t_1$, for example in ϕ direction, the following condition must be applied:

$$\begin{aligned} &\text{if } 0\le t<t_1, && \phi_{\mathrm{des}}(t)=\phi_{\mathrm{des,f}}(t),\\ &\text{if } t_1\le t<t_2, && \phi_{\mathrm{des}}(t)=\text{flip value},\\ &\text{if } t_2<t\le t_{\mathrm{f}}, && \phi_{\mathrm{des}}(t)=\phi_{\mathrm{des,f}}(t)+\text{flip value}, \end{aligned} \tag{30}$$

in which flip value coupled be chosen π for flip and 2π for a complete rotation.

4. A two-point boundary value problem: Optimal path planning

Consider a nonlinear time-invariant affine-in-control system

$$\dot{\mathbf{x}}(t)=\mathbf{f}(\mathbf{x}(t))+\mathbf{g}(\mathbf{x}(t),\mathbf{u}(t)), \tag{31}$$

where $\mathbf{u}(t)\in\mathbb{R}^{m}$ is an input vector and $\mathbf{x}(t)\in\mathbb{R}^{n}$ is a state vector. $\mathbf{g}(\mathbf{x}(t),\mathbf{u}(t)):\mathbb{R}^{n}\times\mathbb{R}^{m}\to\mathbb{R}^{n}$ and $\mathbf{f}(\mathbf{x}(t)):\mathbb{R}^{n}\to\mathbb{R}^{n}$ are smooth vector-valued functions; they satisfy the Lipschitz condition. An equilibrium point of the system (31) sets on $\mathbf{f}(\mathbf{0})=\mathbf{0}$. The goal is to find a set of admissible trajectories, assembled in the vector $\mathbf{x}^{*}(t)\in\mathbb{R}^{n}$, and a set of admissible inputs $\mathbf{u}^{*}(t)\in\mathbb{R}^{m}$, to satisfy the optimal quadratic cost functional integral (Kirk, 2012):

$$J_{\mathrm{o}}=\frac{1}{2}\left[\mathbf{x}^{T}(t_{\mathrm{f}})\mathbf{F}\mathbf{x}(t_{\mathrm{f}})+\int_{0}^{t_{\mathrm{f}}}\{\mathbf{u}^{T}(t)\mathbf{R}\mathbf{u}(t)+\mathbf{x}^{T}(t)\mathbf{Q}\mathbf{x}(t)\}\mathrm{d}t\right],$$

where weighting matrices for states are $\mathbf{Q}\in\mathbb{R}^{n\times n}$ and $\mathbf{F}\in\mathbb{R}^{n\times n}$ in $t\in[0,t_{\mathrm{f}})$ and at t_{f}. Both are symmetric positive semi-definite; the weighting matrix for inputs is $\mathbf{R}\in\mathbb{R}^{m\times m}$, symmetric positive definite.

The system and objective function construct the Hamiltonian

$$H_{\mathrm{o}}(\mathbf{x}(t),\boldsymbol{\lambda}(t),\mathbf{u}(t),t)=\frac{1}{2}\{\mathbf{u}^{T}(t)\mathbf{R}\mathbf{u}(t)+\mathbf{x}^{T}(t)\mathbf{Q}\mathbf{x}(t)\}+\boldsymbol{\lambda}^{T}(t)[\mathbf{f}(\mathbf{x}(t))+\mathbf{g}(\mathbf{x}(t),\mathbf{u}(t))],$$

where the co-state vector is $\boldsymbol{\lambda}(t)$. The conditions for optimality are expressed as:

$$\dot{\mathbf{x}}^{*}(t)=\frac{\partial H_{\mathrm{o}}(\mathbf{x}^{*}(t),\boldsymbol{\lambda}^{*}(t),\mathbf{u}^{*}(t),t)}{\partial\boldsymbol{\lambda}(t)},\tag{32}$$

$$\dot{\boldsymbol{\lambda}}^{*}(t)=-\frac{\partial H_{\mathrm{o}}(\mathbf{x}^{*}(t),\boldsymbol{\lambda}^{*}(t),\mathbf{u}^{*}(t),t)}{\partial\mathbf{x}(t)},\tag{33}$$

$$0=\frac{\partial H_{\mathrm{o}}(\mathbf{x}^{*}(t),\boldsymbol{\lambda}^{*}(t),\mathbf{u}^{*}(t),t)}{\partial\mathbf{u}(t)},\tag{34}$$

and since the final state and time are fixed, the boundary condition (Kirk, 2012):

$$\left[\frac{\partial h(\mathbf{x}^{*}(t_{\mathrm{f}}),t_{\mathrm{f}})}{\partial\mathbf{x}}-\boldsymbol{\lambda}^{*}(t_{\mathrm{f}})\right]^{T}\delta\mathbf{x}_{\mathrm{f}}+\left[\partial H_{\mathrm{o}}(\mathbf{x}^{*}(t_{\mathrm{f}}),\boldsymbol{\lambda}^{*}(t_{\mathrm{f}}),\mathbf{u}^{*}(t_{\mathrm{f}}),t_{\mathrm{f}})+\frac{\partial h(\mathbf{x}^{*}(t_{\mathrm{f}}),t_{\mathrm{f}})}{\partial t}\right]\delta t_{\mathrm{f}}=0,$$

releases $\mathbf{x}^{*}(t_{\mathrm{f}})=\mathbf{x}_{t_{\mathrm{f}}}$; where $h(\mathbf{x}(t_{\mathrm{f}}),t_{\mathrm{f}})=\mathbf{x}^{T}(t_{\mathrm{f}})\mathbf{F}\mathbf{x}(t_{\mathrm{f}})$. As a result, the final weighting matrix $\mathbf{F}$ is not used in tuning the optimal trajectory. In other words, error at the final time is zero in open-loop trajectory generation; therefore, the weighting matrix at the final time is not playing a role in tuning.

Equation (32) results in the system:

$$\dot{\mathbf{x}}^{*}(t)=\mathbf{f}(\mathbf{x}^{*}(t))+\mathbf{g}(\mathbf{x}^{*}(t),\mathbf{u}^{*}(t)),\tag{35}$$

and stationary condition (33) generates

$$\dot{\boldsymbol{\lambda}}^{*}(t)=-\mathbf{Q}\mathbf{x}^{*}(t)-\left(\left[\frac{\partial\mathbf{f}(\mathbf{x}^{*}(t))}{\partial\mathbf{x}(t)}\right]^{T}+\left[\frac{\partial\mathbf{g}(\mathbf{x}^{*}(t),\mathbf{u}^{*}(t))}{\partial\mathbf{x}(t)}\right]^{T}\right)\boldsymbol{\lambda}^{*}(t).\tag{36}$$

The necessary condition for optimality (34) presents the control law, an algebraic equation:

$$\mathbf{u}^{*}(t)=-\mathbf{R}^{-1}\mathbf{B}^{T}(\mathbf{x}^{*}(t))\boldsymbol{\lambda}^{*}(t).\tag{37}$$

in which $\mathbf{B}(\mathbf{x}^{*}(t))$ is $\frac{\partial\mathbf{g}(\mathbf{x}^{*}(t),\mathbf{u}^{*}(t))}{\partial\mathbf{u}(t)}$. Solving equations (35)-(37) numerically and simultaneously, provides the admissible control and trajectories for the optimal control problem. Setting the final ($\mathbf{x}^*(t_\mathrm{f}) = \boldsymbol{\lambda}^*(t_\mathrm{f}) = \mathbf{x}_{t_\mathrm{f}}$) and initial one ($\mathbf{x}^*(\mathbf{0}) = \boldsymbol{\lambda}^*(\mathbf{0}) = \mathbf{x}_0$) boundary conditions for (35) and (36), results in a two-point boundary value problem.

The presented model of the quadrotor defines a state-vector $\mathbf{x}(t) \in \mathbb{R}^{12}$ and a co-state vector $\boldsymbol{\lambda}(t) \in \mathbb{R}^{12}$; consequently, the system to be solved is a set of differential equation $\boldsymbol{\chi}(t) = [\mathbf{x}^T(t), \boldsymbol{\lambda}^T(t)]^T_{24\times 1}$. The control law is also $\mathbf{u}(t) = [T_\mathrm{B}(t) \quad \boldsymbol{\tau}_\mathrm{B}^T(t)]^T_{4\times 1}$. Point-to-point motion is the goal of this study, so the initial and final velocities of states are set zeros $\mathbf{x}^*(0) = \boldsymbol{\lambda}^*(0) = [\boldsymbol{\xi}_1^T(0), \boldsymbol{\xi}_2^T(0), \mathbf{0}, \mathbf{0}]^T$ and $\mathbf{x}^*(t_\mathrm{f}) = \boldsymbol{\lambda}^*(t_\mathrm{f}) = [\boldsymbol{\xi}_1^T(t_\mathrm{f}), \boldsymbol{\xi}_2^T(t_\mathrm{f}), \mathbf{0}, \mathbf{0}]^T$. $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ define the translation and rotation states of the quadrotor with respect. The orientation dynamics (4) is fully actuated and controllable; however, the translation dynamics (3) is under-actuated and common designs will not result in an admissible solution.

Suppose $\mathbf{U}_{3\times 1}$ is a stable control law. Replacing $\ddot{\boldsymbol{\xi}}_1$ from (3) with $\mathbf{U}$ results in (Zuo, 2010):

$$\mathbf{U} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} s_\phi s_\psi + c_\phi s_\theta c_\psi \\ -s_\phi c_\psi + c_\phi s_\theta s_\psi \\ c_\theta c_\phi \end{bmatrix} \frac{T_\mathrm{B}}{m}. \tag{38}$$

Changing (38) to

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 + g \end{bmatrix} = \begin{bmatrix} c_\psi c_\theta & -c_\phi s_\psi + s_\phi s_\theta c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\psi c_\theta & c_\phi c_\psi + s_\phi s_\theta s_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ T_\mathrm{B}/m \end{bmatrix},$$

and multiplying $\mathbf{R}_{ZYX}^T(\boldsymbol{\xi}_2)$ from the left side provides:

$$\begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ -c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & c_\theta s_\phi \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\theta c_\phi \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 + g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ T_\mathrm{B}/m \end{bmatrix}. \tag{39}$$

From the last row of (39), the final form of thrust is found (Zuo, 2010):

$$\begin{aligned} T_\mathrm{B}(t) &= m\left\{\left[\mathbf{R}_{ZYX,3}(\boldsymbol{\xi}_2)\right]_1 U_1 + \left[\mathbf{R}_{ZYX,3}(\boldsymbol{\xi}_2)\right]_2 U_2 + \left[\mathbf{R}_{ZYX,3}(\boldsymbol{\xi}_2)\right]_3 (U_3 + g)\right\} \\ &= m\big[(s_\phi s_\psi + c_\phi s_\theta c_\psi) U_1 + (-s_\phi c_\psi + c_\phi s_\theta s_\psi) U_2 + c_\theta c_\phi (U_3 \\ &\quad + g)\big]. \end{aligned} \tag{40}$$

For generating the co-state dynamics and control law in off-line (parametric) form, the control law $\mathbf{U}$ and T_B must be considered in the Hamiltonian before derivation of (36) and (37). Equation (36) generates 12 co-state equation and Eq. (37) results in 6 control signals, $\mathbf{U}$ and $\boldsymbol{\tau}_\mathrm{B}$, off-line and symbolic. Then, the main system of differential equations $\boldsymbol{\chi}(t) = [\mathbf{x}^T(t), \boldsymbol{\lambda}^T(t)]^T_{24\times 1}$, is solvable with cascaded control law (40) and input moments, $\mathbf{u}(t) = [T_\mathrm{B}(t) \quad \boldsymbol{\tau}_\mathrm{B}^T(t)]^T$.

5. The SDDRE tracking controller: Backward integration solution

Consider nonlinear system (31) in SDC parameterization form with output vector

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t), \tag{41}$$

$$\mathbf{y}(t) = \mathbf{C}(\mathbf{x}(t))\mathbf{x}(t), \tag{42}$$

where output vector is $\mathbf{y}(t) \in \mathbb{R}^p$; and $\mathbf{C}(\mathbf{x}(t)): \mathbb{R}^n \to \mathbb{R}^{p\times n}$, $\mathbf{B}(\mathbf{x}(t)): \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and $\mathbf{A}(\mathbf{x}(t)): \mathbb{R}^n \to \mathbb{R}^{n\times n}$ are held. For second-order Lagrangian systems, n is usually an even number greater than 1 and $n/2 \le p \le n$. The $\{\mathbf{A}(\mathbf{x}(t)), \mathbf{B}(\mathbf{x}(t))\}$ pair is a completely controllable SDC design and the $\{\mathbf{A}(\mathbf{x}(t)), \mathbf{C}(\mathbf{x}(t))\}$ pair is a completely observable design of the system (41) and output (42). The cost function of the SDDRE tracking is in the form of

$$J_{\mathrm{s}} = \frac{1}{2}\left\{\mathbf{e}^T(t_{\mathrm{f}})\mathbf{F}\mathbf{e}(t_{\mathrm{f}}) + \int_0^{t_{\mathrm{f}}} \big(\mathbf{u}^T(t)\mathbf{R}\big(\mathbf{x}(t)\big)\mathbf{u}(t) + \mathbf{e}^T(t)\mathbf{Q}(\mathbf{x}(t))\mathbf{e}(t)\big)\mathrm{d}t\right\},$$

where the error vector is $\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{r}(t)$, $\mathbf{r}(t) \in \mathbb{R}^p$ is desired trajectory, the weighting matrix for the error vector is $\mathbf{Q}(\mathbf{x}(t)): \mathbb{R}^n \to \mathbb{R}^{p\times p}$, and a weighting matrix for inputs is $\mathbf{R}(\mathbf{x}(t)): \mathbb{R}^n \to \mathbb{R}^{m\times m}$; both of them symmetric positive definite. The weighting matrix for states is $\mathbf{F} \in \mathbb{R}^{p\times p}$ imposing penalty at the final time.

The Hamiltonian is specified as

$$\begin{aligned} &H_{\mathrm{s}}(\mathbf{x}(t), \boldsymbol{\lambda}(t), \mathbf{u}(t), t) \\ &\qquad = \frac{1}{2}\{\mathbf{u}^T(t)\mathbf{R}\big(\mathbf{x}(t)\big)\mathbf{u}(t) + \mathbf{e}^T(t)\mathbf{Q}(\mathbf{x}(t))\mathbf{e}(t)\} \\ &\qquad + \boldsymbol{\lambda}^T(t)[\mathbf{A}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t)]. \end{aligned}$$

Using the optimality condition (34) for H_{s} results in control law

$$\mathbf{u}(t) = -\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^T(\mathbf{x}(t))\boldsymbol{\lambda}(t), \tag{43}$$

in which $\boldsymbol{\lambda}(t) = \mathbf{K}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{s}(\mathbf{x}(t))$ is the co-state equation where $\mathbf{K}(\mathbf{x}(t)): \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is optimal gain and the feed-forward vector is $\mathbf{s}(\mathbf{x}(t)): \mathbb{R}^n \to \mathbb{R}^n$. Importing co-state in (43) provides:

$$\mathbf{u}(t) = -\mathbf{R}^{-1}(\mathbf{x}(t))\mathbf{B}^T(\mathbf{x}(t))\{\mathbf{K}(\mathbf{x}(t))\mathbf{x}(t) + \mathbf{s}(\mathbf{x}(t))\}. \tag{44}$$

Computing derivative of co-state vector, $\dot{\boldsymbol{\lambda}}(t)$, and using condition (33), input (44), and mathematical operation, a system of differential equations is found (M. H. Korayem & Nekoo, 2015a):

$$\left\{\begin{matrix} \dot{\mathbf{K}} + \mathbf{KA} + \mathbf{A}^T\mathbf{K} + \mathbf{C}^T\mathbf{QC} - \mathbf{KBR}^{-1}\mathbf{B}^T\mathbf{K} + \\ \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{x}}\mathbf{x}\right]^T - \frac{1}{2}\left[\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\mathbf{u}\right]^T \mathbf{KBR}^{-1}\mathbf{B}^T + \left[\frac{\partial \mathbf{B}}{\partial \mathbf{x}}\mathbf{u}\right]^T\right)\mathbf{K} + \\ \left(\frac{1}{2}\left[\frac{\partial \mathbf{Q}}{\partial \mathbf{x}}[\mathbf{Cx}-\mathbf{r}]\right]^T + \left[\frac{\partial \mathbf{C}}{\partial \mathbf{x}}\mathbf{x}\right]^T \mathbf{Q}\right)\mathbf{C} \end{matrix}\right\}\mathbf{x}$$

$$+\left\{\begin{matrix} \dot{\mathbf{s}} + \left(\mathbf{A}^T + \left[\frac{\partial \mathbf{A}}{\partial \mathbf{x}}\mathbf{x}\right]^T - \frac{1}{2}\left[\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\mathbf{u}\right]^T \mathbf{KBR}^{-1}\mathbf{B}^T + \left[\frac{\partial \mathbf{B}}{\partial \mathbf{x}}\mathbf{u}\right]^T - \mathbf{KBR}^{-1}\mathbf{B}^T\right)\mathbf{s} - \\ \left(\mathbf{C}^T\mathbf{Q} + \frac{1}{2}\left[\frac{\partial \mathbf{Q}}{\partial \mathbf{x}}[\mathbf{Cx}-\mathbf{r}]\right]^T + \left[\frac{\partial \mathbf{C}}{\partial \mathbf{x}}\mathbf{x}\right]^T \mathbf{Q}\right)\mathbf{r} \end{matrix}\right\} \tag{45}$$

$$= \mathbf{0}.$$

The final boundary condition

$$\left[\frac{1}{2}\frac{\partial \mathbf{e}^T(t_\mathrm{f})\mathbf{Fe}(t_\mathrm{f})}{\partial \mathbf{x}} - \boldsymbol{\lambda}(t_\mathrm{f})\right]^T \delta\mathbf{x}_\mathrm{f} + \left[\partial H_\mathrm{s}(\mathbf{x}(t_\mathrm{f}), \boldsymbol{\lambda}(t_\mathrm{f}), \mathbf{u}(t_\mathrm{f}), t_\mathrm{f}) + \frac{1}{2}\frac{\partial \mathbf{e}^T(t_\mathrm{f})\mathbf{Fe}(t_\mathrm{f})}{\partial t}\right]\delta t_\mathrm{f} = 0,$$

concerning $\delta t_\mathrm{f} = 0$, also defines

$$\mathbf{K}(t_\mathrm{f}) = \left[\mathbf{C}(\mathbf{x}(t_\mathrm{f})) + \frac{\partial \mathbf{C}(\mathbf{x}(t_\mathrm{f}))}{\partial \mathbf{x}}\mathbf{x}(t_\mathrm{f})\right]^T \mathbf{FC}(\mathbf{x}(t_\mathrm{f})), \tag{46}$$

$$\mathbf{s}(t_\mathrm{f}) = -\left[\mathbf{C}(\mathbf{x}(t_\mathrm{f})) + \frac{\partial \mathbf{C}(\mathbf{x}(t_\mathrm{f}))}{\partial \mathbf{x}}\mathbf{x}(t_\mathrm{f})\right]^T \mathbf{Fr}(t_\mathrm{f}). \tag{47}$$

Excluding the derivative terms from (45)-(47), the SDDRE and feed-forward equation along with final boundary condition are found:

$$\dot{\mathbf{K}} = -\mathbf{KA} - \mathbf{A}^T\mathbf{K} - \mathbf{C}^T\mathbf{QC} + \mathbf{KBR}^{-1}\mathbf{B}^T\mathbf{K}, \tag{48}$$

$$\dot{\mathbf{s}} = -\mathbf{A}_\mathrm{s}^T\mathbf{s} + \mathbf{C}^T\mathbf{Qr}, \tag{49}$$

$$\mathbf{K}(t_\mathrm{f}) = \mathbf{C}^T(\mathbf{x}(t_\mathrm{f}))\mathbf{FC}(\mathbf{x}(t_\mathrm{f})), \tag{50}$$

$$\mathbf{s}(t_\mathrm{f}) = -\mathbf{C}^T\big(\mathbf{x}(t_\mathrm{f})\big)\mathbf{Fr}(t_\mathrm{f}), \tag{51}$$

where $\mathbf{A}_\mathrm{s} = \mathbf{A} - \mathbf{BR}^{-1}\mathbf{B}^T\mathbf{K}$ is a stable closed-loop matrix. Equation (50) is a boundary condition for the SDDRE (48) and Eq. (51) is the one for feed-forward Eq. (49). In the first round, Eq. (48) and (49) are solved backward from $t_\mathrm{f} \rightarrow t_0$ and after storing feedforward vector $\mathbf{s}\big(\mathbf{x}(t)\big)$ and optimal gain $\mathbf{K}\big(\mathbf{x}(t)\big)$, the forward solution begins. Further details on stability and solution to tracking case of SDDRE are referred to (M. H. Korayem & Nekoo, 2015b).

To implement the closed-loop tracking, the translational dynamic is controlled separately although the interaction with the rotational dynamic is preserved; assuming three virtual inputs for translation control. Then, the three virtual controls are set into one total thrust, such as (40).

The rotational dynamic has already fully actuated with three inputs; so, overall control is a vector $\mathbf{u}(t) = [T_{\mathrm{B}}(t) \quad \boldsymbol{\tau}_{\mathrm{B}}^{T}(t)]^{T}$.

The summary of the tracking solution for the flip is as follows. First, design the open-loop optimal trajectory based on the TPBVP in Section 4, then track the trajectory with the SDDRE nonlinear optimal controller, Section 5. So for the second solution for acrobatic flip maneuver, tracking an optimal smooth trajectory has been employed.

6. Simulation

6-1. Acrobatic flip with optimal trajectory tracking

The two-point boundary value problem is intended to perform a flip maneuver in the desired period. The time of simulation is set 3 seconds, and weighting matrices for TPBVP are $\mathbf{Q} = \mathrm{diag}[\mathbf{1}_{1\times 9}, \mathbf{10}_{1\times 3}]$ and $\mathbf{R} = \mathrm{diag}[\mathbf{1}_{1\times 3}, \mathbf{0}.\mathbf{1}_{1\times 3}]$. The model details are presented in Tables 1 and 2. The initial position is set on the equilibrium point and the desired position is $(-3,2,1)$ m. The initial orientation of the quadrotor is zero for three angles and the final orientation is $(\pi, 0,0)$ rad representing flip in ϕ direction.

The SDDRE tracking parameters were also chosen as follows. The weighting matrices were selected as $\mathbf{R}_{\mathrm{o}} = 0.1 \times \mathbf{I}_{3\times 3}$, $\mathbf{R}_{\mathrm{t}} = 0.1 \times \mathbf{I}_{3\times 3}$, $\mathbf{Q}_{\mathrm{o}} = 1000 \times \mathbf{I}_{3\times 3}$, $\mathbf{F}_{\mathrm{o}} = \mathbf{I}_{3\times 3}$, $\mathbf{Q}_{\mathrm{t}} = 100000 \times \mathbf{I}_{3\times 3}$ and $\mathbf{F}_{\mathrm{t}} = \mathbf{I}_{3\times 3}$. The output vector also defined by $\mathbf{C}_{\mathrm{o}} = \mathbf{C}_{\mathrm{t}} = [\mathbf{I}_{3\times 3}, \mathbf{0}_{3\times 3}]$.

The translation and orientation states of the system are presented in Fig. 3 and Fig. 4. The UAV’s linear and angular velocities are illustrated in Fig. 5 and Fig. 6. The optimal trajectory and configuration of the variable-pitch quadrotor are presented in Fig. 7. The thrust of the system and input moments of the quadrotor are illustrated in Fig. 8 and Fig. 9. The error of SDDRE tracking is shown in Fig. 10. The angles of the blades are illustrated in Fig. 11.

Table 1. Parameters of the quadrotor (Luukkonen, 2011).

para.	values	units	definition
l	0.225	m	dist. between motor and CoM of UAV
R	0.075	m	radi. of the propeller
k	2.98×10^{-5}	$\mathrm{Ns^2/rad^2}$	lift constant - thrust factor (fixed-pitch case)
I_{xx}	4.856×10^{-3}	$\mathrm{kgm^2}$	mom. of iner. of UAV around X axis
I_{yy}	4.856×10^{-3}	$\mathrm{kgm^2}$	mom. of iner. of UAV around Y axis
I_{zz}	8.801×10^{-3}	$\mathrm{kgm^2}$	mom. of iner. of UAV around Z axis
m	0.468	kg	mass of the UAV
D	$\begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$	kg/s	drag matrix
g	9.81	$\mathrm{m/s^2}$	gravity constant

Table 2. Parameters of the variable-pitch rotor dynamics.

para.	values	units	definition
ρ	1.225	$\mathrm{kg/m^3}$	density of air
ω_{min}	196.2	rad/s	minimum constant rotor angular velocity
ω_{ss}	392.5	rad/s	steady-state constant rotor angular velocity
N_{b}	2	-	number of the blades in each rotor
c	0.03	m	chord length of the rotor
C_{l_α}	5.73	1/rad	airfoil lift curve slope (Bhargavapuri et al., 2019)
$\alpha_{\mathrm{max,min}}$	$\pm\pi/6$	rad	maximum and minimum bound of blade angles
v_0	0.5	m/s	air flow velocity

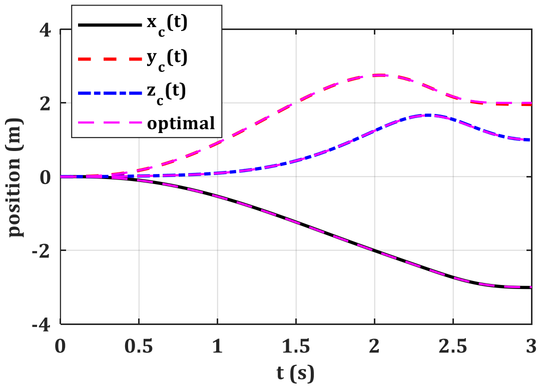

Fig. 3. Translation states of the UAV in the Cartesian coordinate.

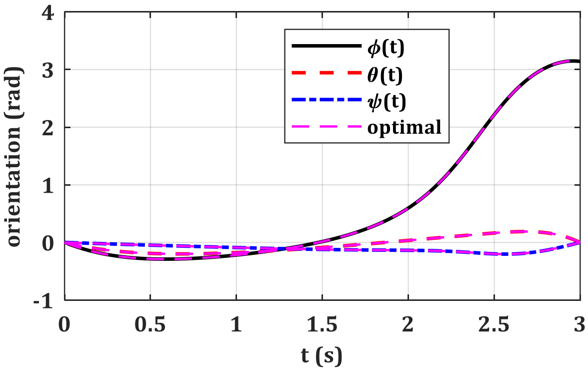

Fig. 4. The UAV's orientation states.

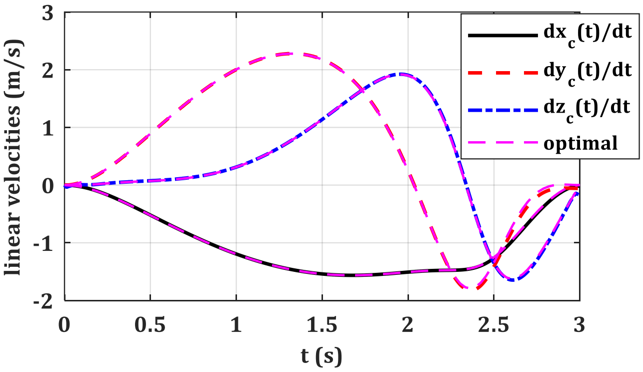

Fig. 5. Linear state velocities.

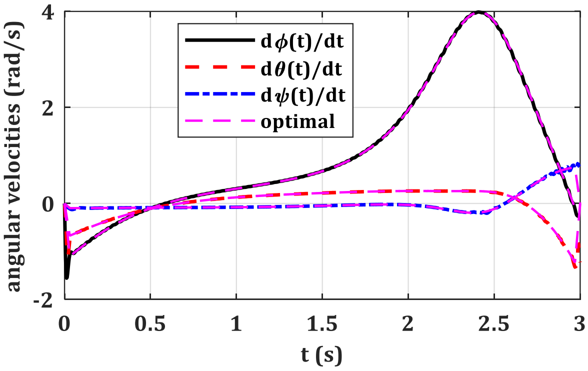

Fig. 6. Angular velocities of the variable-pitch quadrotor.

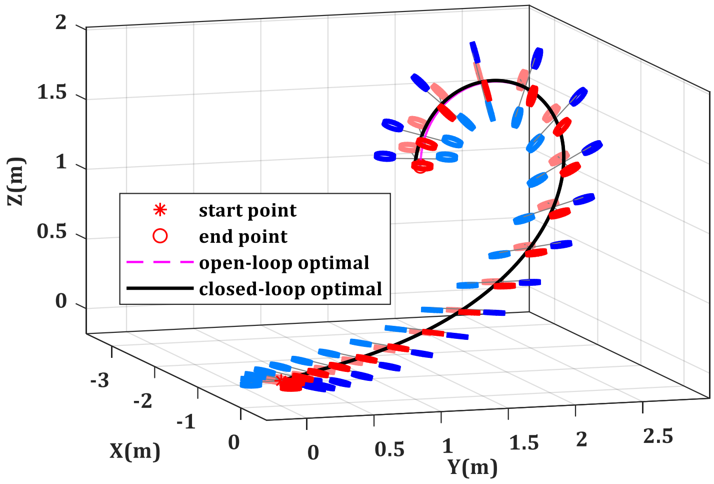

Fig. 7. The optimal trajectory tracking and configuration of the quadrotor during the flip.

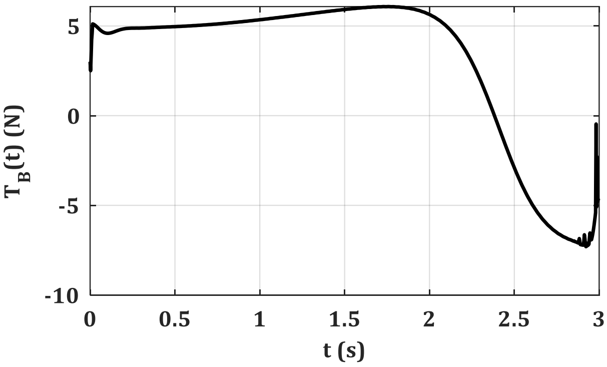

Fig. 8. Total thrust, SDDRE.

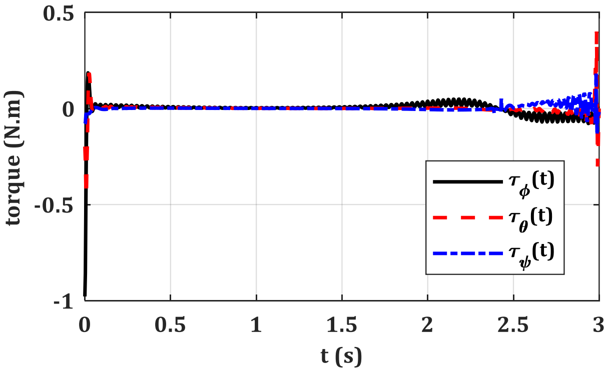

Fig. 9. Input torque, SDDRE.

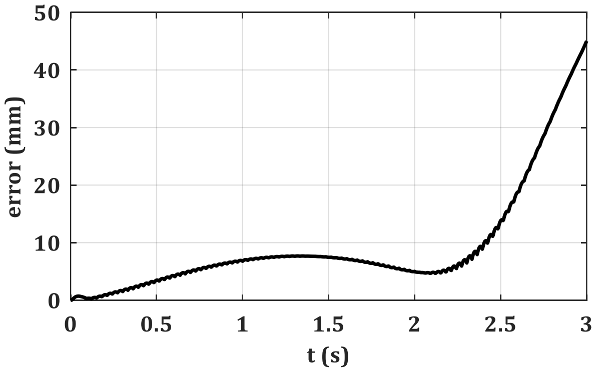

Fig. 10. The error of the SDDRE closed-loop tracking.

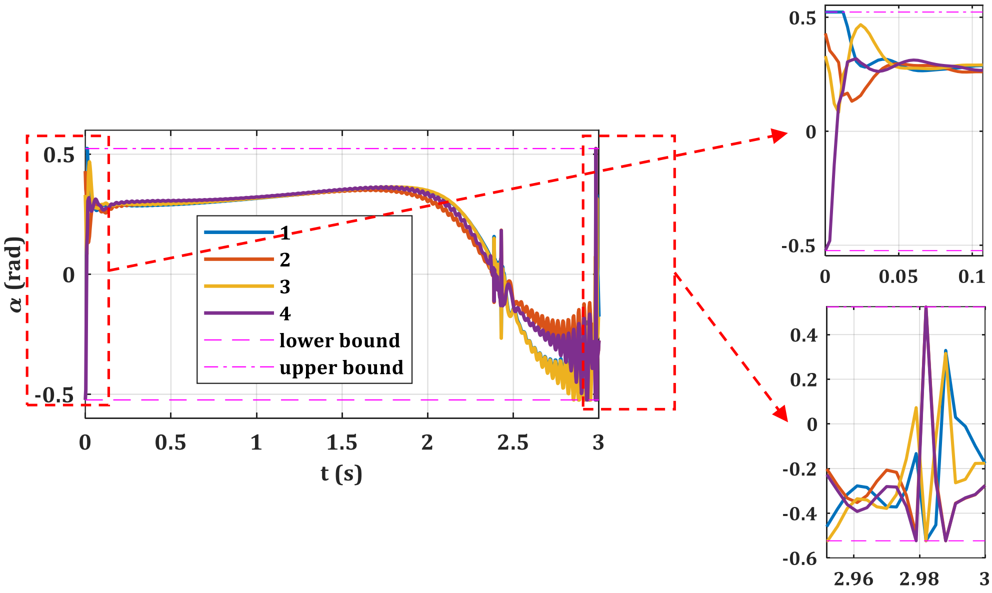

Fig. 11. Blade angles of the system in tracking simulation.

6-2. Acrobatic flip with geometric control: Regulation

This section shows the acrobatic flip maneuver by a regulation approach based on geometric control, presented in Section 3. The first difference between the regulation and tracking approach is the time of the flip. In the tracking problem, the time of the flip and motion between the initial and final conditions was set 3 seconds. For this regulation, almost 6 seconds is necessary to complete the point-to-point motion. So, the time of simulation is defined as 6 seconds. Regulation embeds the flip inside the task, between $t_1 = 1$s and $t_2 = 2$s, therefore the time for the flip is 1s. The initial and final conditions are similar to Section 6-1. The weighting matrices were selected as $\mathbf{R}_\text{o} = \mathbf{I}_{3\times3}$, $\mathbf{R}_\text{t} = \mathbf{I}_{3\times3}$, $\mathbf{Q}_\text{o} = \text{diag}[\mathbf{1}_{1\times3}, \mathbf{0}_{1\times3}]$, and $\mathbf{Q}_\text{t} = \text{diag}[\mathbf{5}_{1\times3}, \mathbf{2.5}_{1\times3}]$. The initial condition for the rotation matrix is $\mathbf{R}(0) = \mathbf{I}_{3\times3}$. The position variables of the UAV are presented in Fig. 12. The results are also compared with the proposed approach for a sudden flip in Ref. (Nekoo, Acosta, Gomez-Tamm, et al., 2019), known as GPGV (global position global velocity). The orientation of the quadcopter is presented in Fig. 13, by rotational generalized coordinates in the global frame. The generated trajectories and configuration of the UAVs are presented in Fig. 14 controlled by geometric control and GPGV. Total thrusts of the UAVs are illustrated in Fig. 15. The input moments of geometric control are demonstrated in Fig. 16. The blade angles are shown in Fig. 17. Checking the determinant of the rotation matrix during numerical integration is important, presented in Fig. 18 to emphasize correct integration. The error function is also presented in Fig. 19.

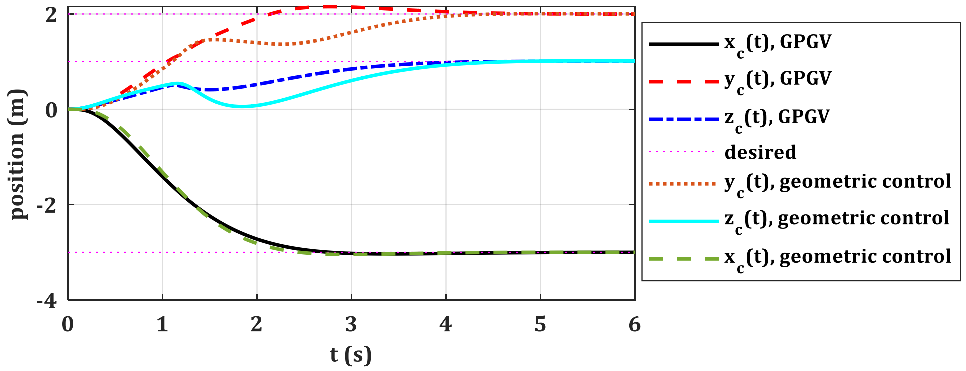

Fig. 12. Cartesian coordinates of the quadcopter in regulation problem; in this graph, geometric control is compared with GPGV (global position global velocity) in Ref. (Nekoo, Acosta, Gomez-Tamm, et al., 2019).

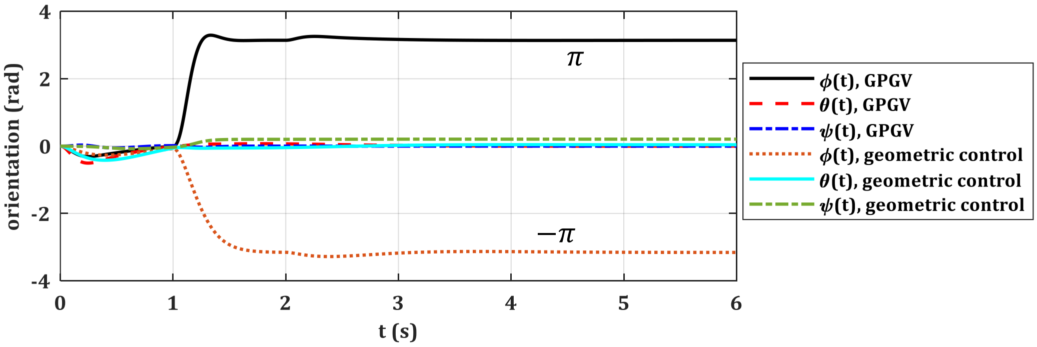

Fig. 13. The orientation of the system for both geometric approach, and GPGV (Nekoo, Acosta, Gomez-Tamm, et al., 2019).

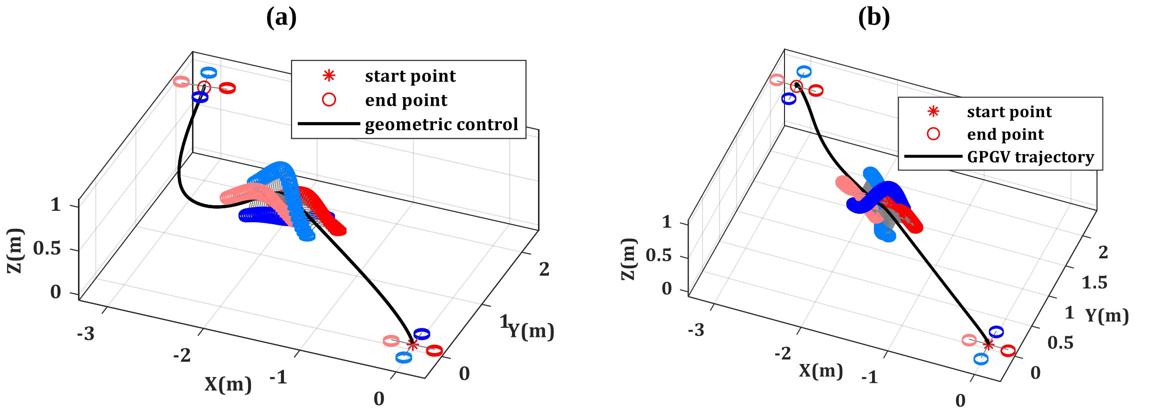

Fig. 14. Flip maneuver and configuration of the UAV by (a) geometric control, and (b) GPGV (Nekoo, Acosta, Gomez-Tamm, et al., 2019).

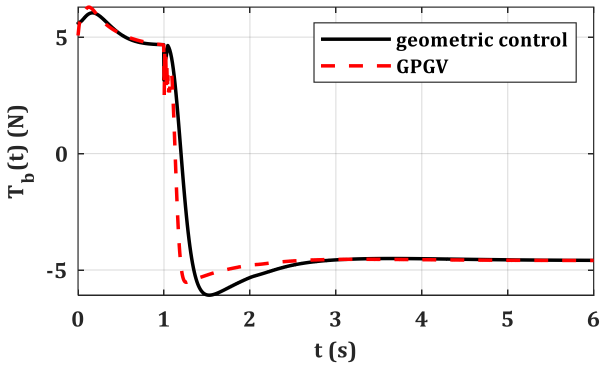

Fig. 15. Total thrust of the UAVs in flip, controlled by regulation approach.

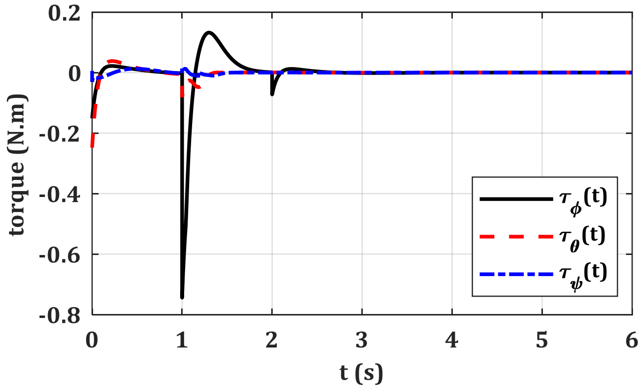

Fig. 16. Input moments of the geometric control approach.

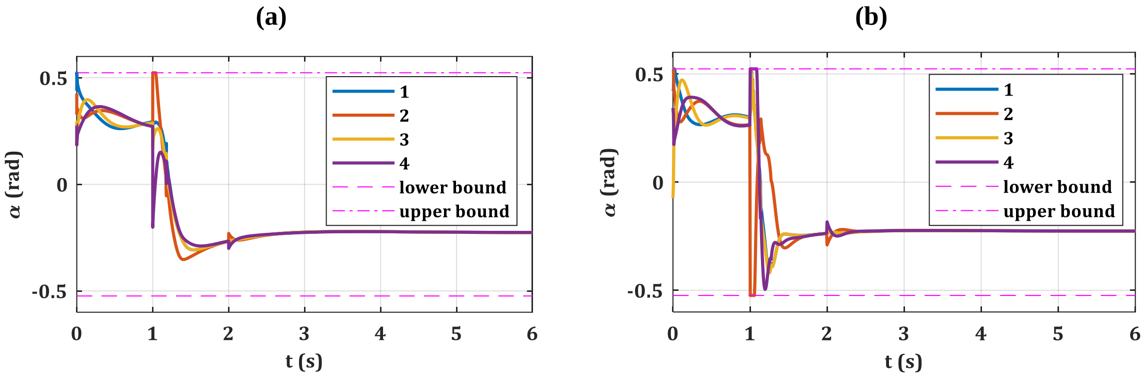

Fig. 17. The angle of the blades for (a) geometric control, and (b) GPGV (Nekoo, Acosta, Gomez-Tamm, et al., 2019).

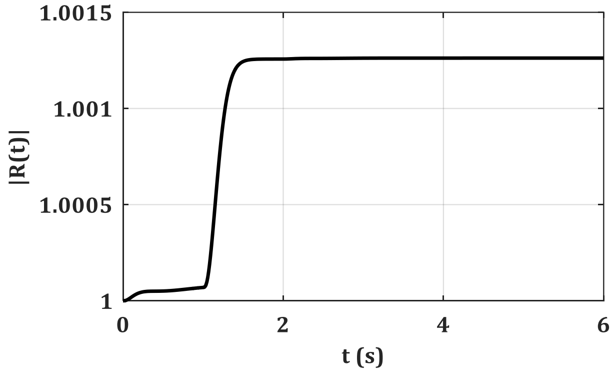

Fig. 18. The determinant of the rotation matrix to check the correctness of integration.

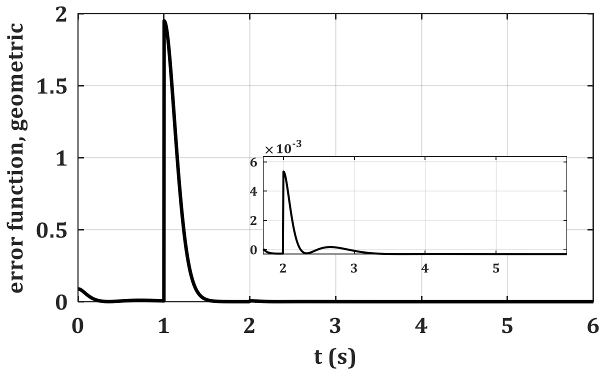

Fig. 19. Error function in the geometric control approach.

6-3. Discussions and comparisons

One of the advantages of the tracking scenario in Section 6-1 is that the initial and final velocity of the UAV is zero and causes stable tracking of the optimal trajectory for the flip. The time-span of the flip is 3 seconds and the flip maneuver is very smoothly done in the entire time-span, Fig. 7. The angle of the blade started from the positive side and went to the negative side at 2.3s to finish the flip, Fig. 11. The error of the motion tracking was found 45mm at the end of the motion and increased gradually over time. The error of the regulation approach by geometric control was found 14.4mm and the GPGV was obtained 4.4mm. The focus of the work in this research is the flip action, not the error, though the reason for the larger error for SDDRE tracking is the nature of path following. The error was zero at the beginning since the initial condition was set perfectly and going towards the final time, the error was increased to follow the trajectory. Comparing geometric control and GPGV, the smoother flip of the geometric control dedicated more time for the flip and less time for the rest of the regulation, therefore obtained more error.

The commanded flip value was set $\pi(\mathrm{rad})$, though the geometric control flipped $-\pi(\mathrm{rad})$, Fig. 13. It should be noted that in the conventional GPGV approach, the desired orientation could be commanded directly; however, in the geometric approach, the desired rotation matrix is commanded. The desired rotation matrix could possess several possible solutions for orientation angles. Observing Fig. 13, flip started at 1s and finished at 2s. Then the transient response appeared to regulate the system to new desired angles based on Eq. (30). Geometric control is smoother and GPGV is faster. This was the motivation to use a geometric approach to avoid input saturation and sudden motions, presented in Fig. 17. Geometric error function perfectly reduced to zero; however, the desired orientation was set arbitrary at the end. *The cause is to command directly the desired rotation matrix instead of desired orientation angles*.

7. Conclusions

This work presented two new methods for flip maneuver, a representative of the most difficult acrobatic motion which possesses singularities. Tracking and regulation were proposed to cover the subject. The tracking method consists of point-to-point motion control of a general quadrotor using open-loop optimal control and then following the trajectory by the SDDRE controller. A fully coupled nonlinear six-DoF model of the quadrotor was considered to implement optimal trajectory generation including flip maneuver. A quadratic cost function was presented to provide a trade-off between energy consumption and state trajectories of the system. Three conditions of optimality generated the control law and co-state dynamics in addition to the dynamics of the system. Solving the three sets of equations simultaneously resulted in optimal motion design. Agile and aggressive maneuver such as flip is not possible with conventional fixed pitch propeller, so, the variable pitch concept was regarded in the dynamics of the system to make the operation possible. The regulation approach based on geometric control was also selected since it is immune to singularities. A combination of the SDRE and the geometric control has been done for the first time and applied for variable-pitch rotor quadcopter control. The geometric control generated a smoother trajectory for flip and obtained less variation and saturation for blade angles.

Acknowledgments

This work is supported by the HYFLIERS project (HYbrid FLying-rolling with-snakE-aRm robot for contact inspection) funded by the European Commission H2020 Programme under grant agreement ID: 779411 (https://cordis.europa.eu/project/rcn/213049).

References

Abdelrahman, M., & Park, S.-Y. (2013). Spacecraft attitude control via a combined state-dependent Riccati equation and adaptive neuro-fuzzy approach. *Aerospace Science and Technology, 26*(1), 16-28

Batmani, Y., Davoodi, M., & Meskin, N. (2016). On design of suboptimal tracking controller for a class of nonlinear systems *American Control Conference* (pp. 1094-1098). Boston, MA, USA: IEEE.

Beeler, S. C., Tran, H. T., & Banks, H. T. (2003). State estimation and tracking control of nonlinear dynamical systems *Control and Estimation of Distributed Parameter Systems* (pp. 1-24): Springer.

Bhargavapuri, M., Sahoo, S. R., & Kothari, M. (2019). Robust nonlinear control of a variable-pitch quadrotor with the flip maneuver. *Control Engineering Practice, 87*, 26-42

Boscariol, P., & Richiedei, D. (2018). Robust point-to-point trajectory planning for nonlinear underactuated systems: Theory and experimental assessment. *Robotics and Computer-Integrated Manufacturing, 50*, 256-265

Cimen, T. (2007). Approximate nonlinear optimal SDRE tracking control *Automatic Control in Aerospace* (Vol. 17, pp. 147-152).

Cutler, M., & How, J. (2012). Actuator constrained trajectory generation and control for variable-pitch quadrotors *AIAA Guidance, Navigation, and Control Conference* (pp. 4777).

Fernando, T., Chandiramani, J., Lee, T., & Gutierrez, H. (2011). Robust adaptive geometric tracking controls on SO (3) with an application to the attitude dynamics of a quadrotor UAV *50th IEEE Conference on Decision and Control and European Control Conference* (pp. 7380-7385). Orlando, FL, USA: IEEE.

Gamagedara, K., Bisheban, M., Kaufman, E., & Lee, T. (2019). Geometric controls of a quadrotor UAV with decoupled yaw control *American Control Conference* (pp. 3285-3290). Philadelphia, PA, USA: IEEE.

Goodarzi, F., Lee, D., & Lee, T. (2013). Geometric nonlinear PID control of a quadrotor UAV on SE (3) *2013 European control conference* (pp. 3845-3850). Zürich, Switzerland: IEEE.

Goodarzi, F. A., Lee, D., & Lee, T. (2015). Geometric control of a quadrotor UAV transporting a payload connected via flexible cable. *International Journal of Control, Automation and Systems, 13*(6), 1486-1498

Hassan, A. M., & Taha, H. E. (2017). Geometric control formulation and nonlinear controllability of airplane flight dynamics. *Nonlinear Dynamics, 88*(4), 2651-2669

Hernández-Díez, J.-E., Méndez-Barrios, C.-F., Mondié, S., Niculescu, S.-I., & González-Galván, E. J. (2018). Proportional-delayed controllers design for LTI-systems: a geometric approach. *International Journal of Control, 91*(4), 907-925

https://www.oulu.fi/hyfliers/.

Invernizzi, D., & Lovera, M. (2017). Geometric tracking control of a quadcopter tiltrotor UAV. *IFAC-PapersOnLine, 50*(1), 11565-11570

Jung, J.-W., Park, S.-Y., Kim, S.-W., Eun, Y., & Chang, Y.-K. (2013). Hardware-in-the-loop simulations of spacecraft attitude synchronization using the state-dependent Riccati equation technique. *Advances in Space Research, 51*(3), 434-449

Kirk, D. E. (2012). *Optimal control theory: An introduction*: Courier Corporation.

Korayem, A. H., Nekoo, S. R., & Korayem, M. H. (2019). Sliding mode control design based on the state-dependent Riccati equation: Theoretical and experimental implementation. *International Journal of Control, 92*(9), 2136-2149

Korayem, M. H., & Nekoo, S. R. (2015a). State-dependent differential Riccati equation to track control of time-varying systems with state and control nonlinearities. *ISA Transactions, 57*, 117-135

Korayem, M. H., & Nekoo, S. R. (2015b). Suboptimal tracking control of nonlinear systems via state-dependent differential Riccati equation for robotic manipulators *Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on* (pp. 25-30). Tehran, Iran: IEEE.

Korayem, M. H., & Nikoobin, A. (2008). Maximum payload for flexible joint manipulators in point-to-point task using optimal control approach. *The International Journal of Advanced Manufacturing Technology, 38*(9-10), 1045-1060

Lee, T. (2012). Exponential stability of an attitude tracking control system on SO (3) for large-angle rotational maneuvers. *Systems & Control Letters, 61*(1), 231-237

Loianno, G., Brunner, C., McGrath, G., & Kumar, V. (2016). Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU. *IEEE Robotics and Automation Letters, 2*(2), 404-411

Luukkonen, T. (2011). Modelling and control of quadcopter. *Independent research project in applied mathematics, Espoo, 22*

Nekoo, S. R. (2020). A PDE breach to the SDRE. *Asian Journal of Control, 22*(2), 667-676

Nekoo, S. R., Acosta, J. Á., Gomez-Tamm, A. E., & Ollero, A. (2019). Optimized thrust allocation of variable-pitch propellers quadrotor control: A comparative study on flip maneuver *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)* (pp. 86-95). Cranfield, United Kingdom, United Kingdom: IEEE.

Nekoo, S. R., Acosta, J. Á., & Ollero, A. (2019). Fully coupled six-DoF nonlinear suboptimal control of a quadrotor: Application to variable-pitch rotor design *Iberian Robotics conference* (pp. 72-83): Springer.

Pretorius, A., & Boje, E. (2014). Design and modelling of a quadrotor helicopter with variable pitch rotors for aggressive manoeuvres. *IFAC Proceedings Volumes, 47*(3), 12208-12213

Raj, N., Banavar, R. N., & Kothari, M. (2017). Attitude tracking control for aerobatic helicopters: A geometric approach *IEEE 56th Annual Conference on Decision and Control* (pp. 1951-1956). Melbourne, Australia: IEEE.

Seguchi, H., & Ohtsuka, T. (2003). Nonlinear receding horizon control of an underactuated hovercraft. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal, 13*(3-4), 381-398

Shastry, A. K., Bhargavapuri, M. T., Kothari, M., & Sahoo, S. R. (2018). Quaternion based adaptive control for package delivery using variable-pitch quadrotors *Indian Control Conference* (pp. 340-345): IEEE.

Shi, X.-N., Zhang, Y.-A., & Zhou, D. (2015). A geometric approach for quadrotor trajectory tracking control. *International Journal of Control, 88*(11), 2217-2227

Wang, M., Dong, X., Ren, X., & Chen, Q. (2020). SDRE based optimal finite-time tracking control of a multi-motor driving system. *International Journal of Control*, 1-13

Zhu, Y., Chen, X., & Li, C. (2017). Some Discussions about the Error Functions on SO (3) and SE (3) for the Guidance of a UAV Using the Screw Algebra Theory. *Advances in Mathematical Physics, 2017*(Article ID 1016530)

Zuo, Z. (2010). Trajectory tracking control design with command-filtered compensation for a quadrotor. *IET control theory & applications, 4*(11), 2343-2355