

Research Article

Quadcopter Aggressive Maneuvers along Singular Configurations: An Energy-Quaternion Based Approach

Ayman A. El-Badawy^{1,2} and Mohamed A. Bakr²

¹*Mechanical Engineering Department, Al-Azhar University, Cairo, Egypt*

²*Mechatronics Engineering Department, German University in Cairo, El Tagammosa El Khames, New Cairo, Cairo 11835, Egypt*

Correspondence should be addressed to Ayman A. El-Badawy; ayman.elbadawy@guc.edu.eg

Received 5 November 2015; Revised 13 December 2015; Accepted 22 December 2015

Academic Editor: Xiao He

Copyright © 2016 A. A. El-Badawy and M. A. Bakr. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Automatic aggressive maneuvers with quadcopters are regarded as a highly challenging control problem. The aim is to tackle the singularities that exist in a vertical looping maneuver. Modeling singularities are resolved by writing the equations-of-motion of the quadcopter in quaternion form. Physical singularities due to underactuation are resolved by using an energy-based control. Energy-based control is utilized to overcome the uncontrollability of the quadcopter at physical singular configurations, for instance, when commanding the quadcopter to gain altitude while pitched at 90° . Three looping strategies (circular, clothoidal, and newly developed constant thrust) are implemented on a nonlinear model of the quadcopter. The three looping strategies are discussed along with their advantages and limitations.

1. Introduction

A looping maneuver (Figure 1) is executed when the centroid of the quadcopter moves along a circular path while pitching up to complete a 360° rotation.

Singular configurations are encountered while tracking the trajectories of a looping maneuver. Singular configurations are abnormal situations that should be avoided since they indicate either a malfunction of the mechanism, or a bad model [1]. Singular configurations are divided into two types: physical singular configurations and modelling singular configurations. Modelling singular configurations might reflect bad modelling decisions, for example, in a looping maneuver using Euler angles to model the quadcopter dynamics instead of quaternions [2]. Physical singular configurations reflect limitations in the design. For example, when the quadcopter's pitch angle approaches 90° , the controller loses the ability to command an acceleration in the vertical direction as the quadcopter is uncontrollable in this configuration. This is a limitation in the quadcopter's design since it has only 4 actuators in 6DOF (underactuated). At the physical singular configurations the controllability and the Jacobian matrices are rank deficient. Hence the quadcopter is uncontrollable.

Recently the design and implementation of control algorithms for aggressive maneuvers for unmanned aerial vehicles have been of interest to many research institutes and universities [3–13]. In the field of unmanned helicopters, human-piloted maneuvers are executed such that reference trajectories for the maneuver are extracted. These trajectories are then taught to the helicopters controller through reinforcement learning and so the maneuver can be replicated [3]. This approach implemented an entire air show of different maneuvers; however, this work addresses the challenge of designing trajectories in the full without the usage of human pilots.

Designing reference maneuver trajectories is not a simple task and so research has gone into the direction of reducing the complexity of the problem by dividing the flip trajectory into five steps (Acceleration, Start Rotation, Coast, Stop Rotate, and Recovery) [4–6]. Then these five steps were associated with five parameters. These five parameters were optimized to obtain the desired flips. This approach works on the optimization of the final state parameters and not the whole trajectory and thus it managed to perform flips and not a proper looping maneuver.

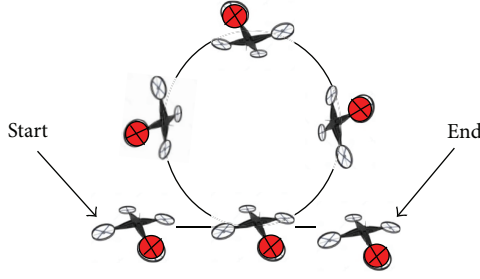


FIGURE 1: Looping maneuver.

In [7] the authors utilized geometric methods to define controllers that can achieve complex aerobatic maneuvers for a quadcopter. The quadcopter dynamics are modeled and expressed globally on the configuration manifold. It is coordinate-free, and therefore it overcomes modeling singularities. The authors managed to implement complex maneuvers such as recovering from being initially upside down. The paper considers the inverted orientation to be undesirable with a 180° attitude error. On the other hand, part of the maneuver in this work is to be inverted while executing an aggressive 360° loop.

The authors in [8] tackle the problem of generating real time trajectories for a quadcopter. These trajectories ensure safe passage through corridors. The authors discarded the small roll and pitch angle approximations as tight aggressive turns were required to maneuver the quadcopter in the corridors. On the other hand in this work the maneuver is more aggressive as it requires the pitch angle to change 360° . Moreover the looping maneuver done in this paper requires dealing with singularities, unlike tight turns where the pitch and roll angle are limited less than 90° .

The authors in [9] extend the flight envelope of the quadcopter by improving the accuracy of the aerodynamic model. The aerodynamic model is improved through the analysis of the blade flapping and thrust. The authors achieved high speeds and performed aggressive maneuvers. A stall turn maneuver is executed where a sudden pitch moment is applied, the quadcopter enters a steep climb, trading kinetic energy for height. At the peak of the maneuver where the velocity approaches zero, a yaw moment is applied to reverse the direction. However, in this paper the maneuver is more complicated where the quadcopter is required to trade kinetic energy with height while pitching up at the same time to execute the aggressive maneuver.

In [10] a complex aerobatic maneuver is approached by decomposing it into a sequence of discrete maneuvers. The author achieved safe switching between the maneuver segments while performing an autonomous backflip. The backflip maneuver is broken into three main stages: impulse, drift, and recovery. The maneuver initializes by rotating the quadcopter. Upon reaching the end of the first maneuver segment the motors are turned off for the drift mode, where the quadcopter rotates and falls under gravity. Finally, the recovery mode brings the quadcopter back to hover condition. The author in the paper depends on the inertia to pitch up and fall in the drift mode. On the other hand, in

this work during the drift mode when the quadcopter has a pitch angle greater than 90° and less than 270° the motors are not turned off and energy is utilized to overcome the singular configurations and avoid falling due to gravity.

Research in [11–13] was successful in executing aggressive maneuvers as flying through narrow, vertical gaps such that the pitch angle of the quadcopter reaches 90° . Also the authors managed to perch on inverted surfaces with maximum inclination angle of 120° . This is done by designing trajectories and controllers defined by a sequence of segments. Each controller of each segment is then refined through iteration to account for errors in the dynamic model and noise in the sensors and actuators.

The goal of this work is to execute an aggressive 360° looping maneuver as shown in Figure 1. Performing such maneuver singularities that exist along the maneuver path has to be resolved. Modeling singularities were tackled by writing the equations-of-motion of the quadcopter in quaternion form. Energy-based control is then utilized to overcome the uncontrollability of the quadcopter at physical singular configurations. Three looping maneuvers (circular, clothoidal, and newly developed constant thrust) were implemented on a nonlinear model of the quadcopter. The three looping maneuvers were discussed along with their advantages and limitations.

2. Quadcopter Modelling and Control

2.1. Modelling. The inertial frame $I : \{O^I, X^I, Y^I, Z^I\}$ is defined by the ground, with gravity pointing in the positive Z^I direction. The body frame $B : \{O^B, X^B, Y^B, Z^B\}$ is defined by the orientation of the quadcopter, with the rotor axes pointing in the negative Z^B direction and the arms pointing in the x and y directions as shown in Figure 2.

The quad-rotor helicopter dynamic equations in quaternion form are found to be [14]

$$\begin{aligned} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \mathbf{q} \otimes \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m} \end{bmatrix} U_1 \otimes \mathbf{q}^*, \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} \frac{(qr(I_z - I_y) + U_2 + I_p q (\Omega_{\text{front}} + \Omega_{\text{rear}} + \Omega_{\text{right}} + \Omega_{\text{left}}))}{I_x} \\ \frac{(pr(I_x - I_z) + U_3 - I_p p (\Omega_{\text{front}} + \Omega_{\text{rear}} + \Omega_{\text{right}} + \Omega_{\text{left}}))}{I_y} \\ \frac{(pq(I_y - I_x) + U_4)}{I_z} \end{bmatrix}, \\ \dot{\mathbf{q}} &= \frac{1}{2} \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} \otimes \mathbf{q}. \end{aligned} \quad (1)$$

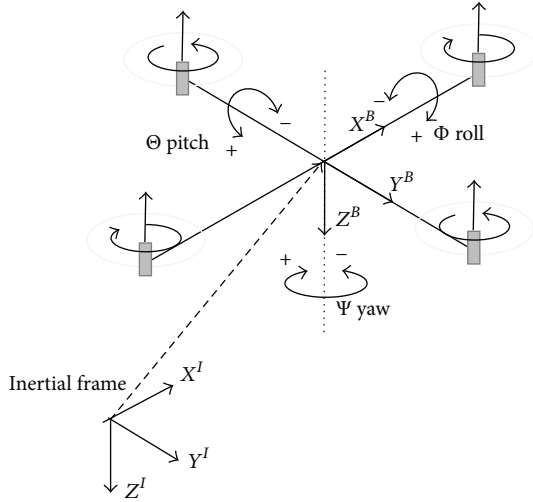


FIGURE 2: Coordinate systems.

The x , y , and z represent the position of the centroid of the quadcopter relative to the inertial frame. The quadcopter is affected by the gravity shown by the gravitational acceleration g . The mass of the quadcopter is represented by m . I_x , I_y , and I_z are the moments of inertia about the X^B , Y^B , and Z^B axes, respectively. I_p is the propeller's moment of inertia about the Z^B axis. The quaternion \mathbf{q} is a hyper complex number of rank 4, where $\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^T$. The quaternion units from q_1 to q_3 are called the vector part of the quaternion, while q_0 is the scalar part. The multiplication of two quaternions is done by the Kronecker product, denoted as \otimes [14]. The quadcopter's p , q , and r represent the body-fixed angular velocity about the X^B , Y^B , and Z^B axes, respectively. The rotor propellers' speed are represented by Ω_{front} , Ω_{rear} , Ω_{right} , and Ω_{left} . Finally, U_1 , U_2 , U_3 , and U_4 are the four control inputs of the quadcopter.

2.2. Controller Design. In order to track the desired trajectories a simple PD control law is designed for motions. For example, the roll angle control law is found to be

$$q_{1\text{err}} = [0 \ 1 \ 0 \ 0] (\mathbf{q}_c \otimes \mathbf{q}_a), \quad (2)$$

$$U_2 = K_p q_{1\text{err}} + K_d \dot{q}_{1\text{err}}, \quad (3)$$

where (2) is the quaternion difference between the command quaternion \mathbf{q}_c and the actual quaternion \mathbf{q}_a . This difference gives the error quaternion [14]. The error quaternion is then multiplied by a vector to obtain the error quaternion " $q_{1\text{err}}$ " that is related to the roll angle. Similarly, the pitch and yaw controllers are found to be

$$U_3 = K_p q_{2\text{err}} + K_d \dot{q}_{2\text{err}}, \quad (4)$$

$$U_4 = K_p q_{3\text{err}} - q_{3a} + K_d \dot{q}_{3\text{err}}.$$

The control signal U_1 will be derived later while designing the trajectories.

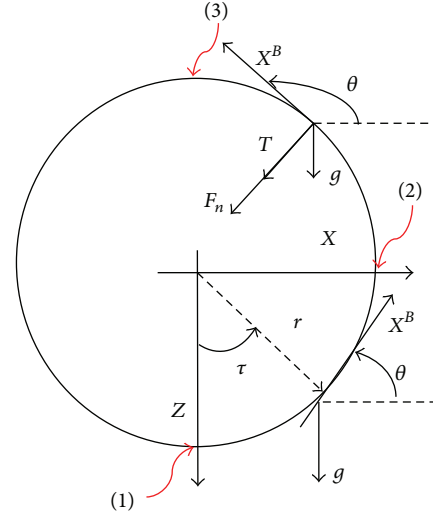


FIGURE 3: Perfect circular looping. In order to have sufficient energy for the quadcopter to traverse the interval between points (2) and (3) as shown in Figure 3, the quadcopter must build up enough energy at point (1). During the interval between points (2) and (3) the thrust vector only acts as a direction changer and does not contribute in increasing the altitude.

2.3. Looping Trajectory Generation: An Energy-Based Approach. In order to approach the physical singularity problem looping trajectories are designed. These trajectories consist of a looping path and the thrust necessary to perform such loop. By calculating the controllability matrix at pitch angles 90° and 270° the system is uncontrollable at these points (physical singularity) and so the thrust calculated must be sufficient such that the quadcopter can cross these physical singularities using stored energy. In this section three types of looping paths are considered, a circular path, a clothoidal path, and a noncircular constant thrust path. The assumptions used for deriving the looping paths are as follows.

- (1) The thrust vector is always directed normal to the path.
- (2) The velocity is being commanded based on the energy conservation principle assuming a conservative system.
- (3) The zero potential energy reference is the lowest point in the looping.

2.3.1. Perfect Circular Looping. Shown in Figure 3 is a side view of a perfect circular looping path.

By resolving the gravity and normalised thrust vector in the normal direction to the path, the normal force normalized by mass is found to be

$$F_n = T - g \cos(\tau). \quad (5)$$

By substituting (5) in Newton's second law in the normal direction, the normalised thrust is found to be

$$T = \frac{(V)^2}{r} + g \cos(\tau). \quad (6)$$

The term V^2/r is the centripetal acceleration and the $g \cos(\tau)$ is the gravity component in the normal direction. Since the quadcopter's X^B coordinate axis is always tangential to the path, therefore the path parameter τ can be replaced by the pitch angle θ . The magnitude of the velocity in (6) can be written using the conservation of energy principle:

$$V = \sqrt{\frac{2(L - U)}{m}}, \quad (7)$$

where L is the total energy and U is the potential energy of the quadcopter. Substituting (7) in (6) results in an equation relating the normalised thrust with the energy in the system:

$$T = \frac{2(L - U)}{mr} + g \cos(\theta). \quad (8)$$

From assumption (3) the potential energy is calculated using trigonometry to be

$$U = mgr(1 - \cos(\theta)). \quad (9)$$

Substitute (9) in (8) and simplify

$$T = \frac{2L}{mr} + 3g \cos(\theta) - 2g. \quad (10)$$

The minimum total energy required is found to be

$$L > \frac{mr(-3g \cos(\theta) + 2g)}{2}. \quad (11)$$

Note that the equality has been replaced by greater than since the rotational dynamics and thrust dynamics are coupled; therefore, at the peak of the looping, the thrust cannot be equal to zero.

The quadcopter must have enough energy to do the whole looping especially at the peak of the loop at $\tau = \theta = \pi$. Thus, the minimum total energy required to do a perfect circular looping must satisfy

$$L > \frac{5mrg}{2}. \quad (12)$$

The minimum total energy $5mrg/2$ from (12) is substituted in (10). The maximum value of the normalised thrust at the lowest point of the looping where the pitch angle is equal to zero. At pitch angle equal to zero the thrust normalised by mass is equal to $6g$. This thrust magnitude (normalised by mass) might be more demanding while designing the quadcopter. Therefore another approach is considered in order to perform the maneuver. The clothoid looping is mainly used in roller coasters as it is characterised by lower load factor than a perfect circular looping [15].

2.3.2. Clothoid Looping. The clothoid looping is constructed from two clothoid curves as shown in Figure 4. λ is the path parameter, and E_t and E_n are the tangent and normal unit vectors. The first spiral curve extends from the start ($\lambda = 0$) to the peak ($\lambda = \sqrt{2}$) and the second spiral curve is a mirror

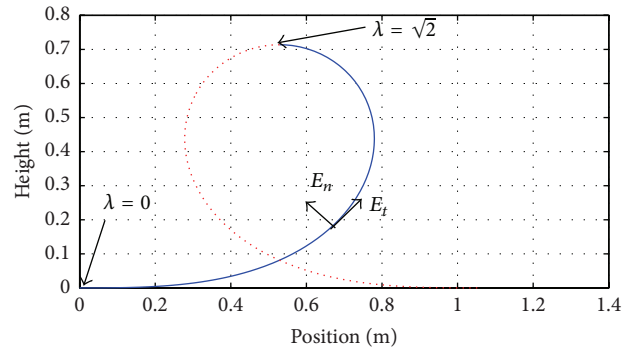


FIGURE 4: Clothoid looping.

of the first curve about the vertical axis passing through the peak.

The thrust analysis of the clothoid looping is done on one half of the loop (one clothoid curve). A clothoid curve is a member of the Euler spiral curves. It is a curved path whose curvature is linearly related to the arc length [15]. The parametric representation of the clothoid curve is found to be

$$\begin{aligned} x(\lambda) &= B \cdot C(\lambda), \\ z(\lambda) &= -B \cdot S(\lambda), \end{aligned} \quad (13)$$

where B is a constant which will determine the maximum height of the clothoid looping as it will be shown later. The functions $C(\lambda)$ and $S(\lambda)$ are the Fresnel integrals. They are expressed as definite integrals

$$C(\lambda) = \int_0^\lambda \cos\left(\frac{\pi}{2}t^2\right) dt, \quad (14)$$

$$S(\lambda) = \int_0^\lambda \sin\left(\frac{\pi}{2}t^2\right) dt, \quad (15)$$

where t is a dummy variable.

By differentiating (13) the speed along the clothoid curve is equal to " B " and so the path length is found to be

$$s = B\lambda. \quad (16)$$

The position vector $\bar{R}^T(s)$ of the first curve of the looping can be written as

$$\bar{R}^T(s) = \left(B \cdot C\left(\frac{s}{B}\right) - B \cdot S\left(\frac{s}{B}\right) \right)^T. \quad (17)$$

The tangent unit vector is calculated by differentiating with respect to s [15] and thus it is determined to be

$$\bar{E}_t = \left(\cos\left(\frac{\pi s^2}{2B^2}\right) - \sin\left(\frac{\pi s^2}{2B^2}\right) \right)^T. \quad (18)$$

The normal unit vector is calculated from the tangent unit vector and is found to be [15]

$$\bar{E}_n = \left(-\sin\left(\frac{\pi s^2}{2B^2}\right) - \cos\left(\frac{\pi s^2}{2B^2}\right) \right)^T, \quad (19)$$

while the radius of curvature is

$$r = \frac{B^2}{\pi s}. \quad (20)$$

From the energy conservation principle, the magnitude of the velocity of one half of the clothoid looping can be written as

$$V^2 = V_0^2 - 2gz, \quad (21)$$

$$V = \sqrt{V_0^2 - 2gB \cdot S\left(\frac{s}{B}\right)}, \quad (22)$$

where V_0 is the initial velocity in the X^I direction. Same as the perfect circular path, if the normalised thrust and gravity vectors are resolved in the normal direction and then substituted in Newton's second law, the normalised thrust is expressed as

$$T = \frac{(V)^2}{r} + g \cos(\tau). \quad (23)$$

The path parameter τ is interpreted as the path angle which is also equal to the quadcopter's pitch angle since the normalised thrust vector is always perpendicular to the path. Now all the equations are available to calculate the normalised thrust requirements to do a clothoid looping. By substituting (22), (20) and the path parameter from (19) in (23) the normalised thrust is found to be

$$T = \frac{\pi s V_0^2}{B^2} - \frac{2\pi s}{B} g S\left(\frac{s}{B}\right) + g \cos\left(\frac{\pi s^2}{2B^2}\right), \quad (24)$$

where the command input of the quadcopter can be written as:

$$U_1 = mT. \quad (25)$$

Since the quadcopter is limited to produce only positive thrust, therefore the thrust normalised by mass vector must be always greater than zero. The most critical point in the loop is at the peak at $\lambda = s/B = \sqrt{2}$, where the quadcopter must have enough energy to cross the peak of the looping. Therefore by substituting the $s/B = \sqrt{2}$, $S(\sqrt{2}) \approx 0.71$ (calculated numerically) and the thrust being greater than zero, the condition for the initial velocity is found to be

$$V_0 > \sqrt{1.65gB}. \quad (26)$$

Physically B is used to determine the maximum height of the clothoid looping. Since $S(\sqrt{2}) \approx 0.71$ and $z(t) = -B \cdot S(t)$ therefore the maximum height of the looping is approximately equal to $0.71B$. So the choice of B determines both the maximum height and the minimum initial velocity needed to perform the looping. The normalised thrust of (24) is plotted with the path length as shown in Figure 5.

Figure 5 shows two curves for the normalised thrust requirements to do one half of a clothoid looping. Both curves represent the same looping with the same maximum height ($B = 5$). The difference between them is the initial velocity.

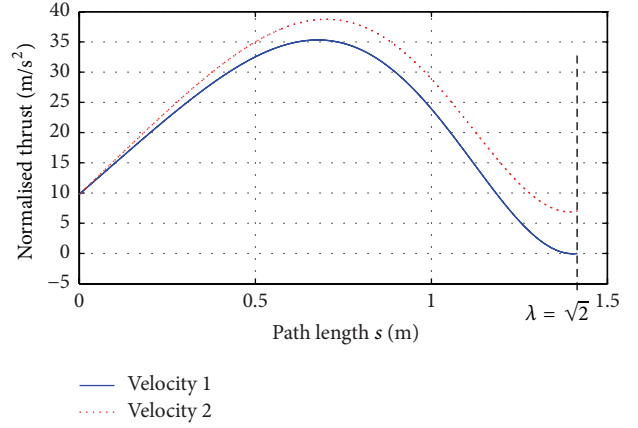


FIGURE 5: Normalised thrust requirements of one half of a clothoid loop.

Velocity 1 is the minimum initial velocity required to not to fall and velocity 2 is greater than velocity 1. Since B is the same, if the initial velocity is chosen higher than the minimum velocity, the centripetal acceleration along the path will increase. Therefore more normalised thrust is required in order to follow the looping path as shown in Figure 5. For the minimum velocity curve there is a linear increase of the normalised thrust from $1g$ to a maximum of $3.6g$. Based on design consideration this maximum thrust might be demanding but less than the circular path.

A second approach is then considered as shown next where the normalised thrust is specified and then the path is calculated.

2.3.3. Constant Thrust Looping: A New Trajectory Generation Approach. In this section the thrust is specified and then the path of the looping is determined. The thrust is assumed to be constant along the looping path.

The normalised thrust equation for an arbitrary path is shown in (23). Then by substituting the magnitude of the velocity with the conservation of energy equation, the normalised thrust is found to be

$$T = \frac{V_0^2 - 2gh}{r} + g \cos(\theta), \quad (27)$$

where h is the height which is equivalent to $-Z^I$ axis. Equation (27) contains three unknowns along the path, the height h , the radius of curvature r , and the pitch angle θ . The path is divided into arcs where a new pitch angle is calculated from the initial values of h and r along with the V_0 and T which are known. The new pitch angle is used to determine the final values of the arc which are used as the initial values for the next arc as it will be shown.

Figure 6 shows an arc of the looping path, where s is the path length and r is the radius of curvature. θ is the path parameter which is also equal to the pitch angle. Finally V is the velocity. The path length s is found to be

$$s = r\theta. \quad (28)$$

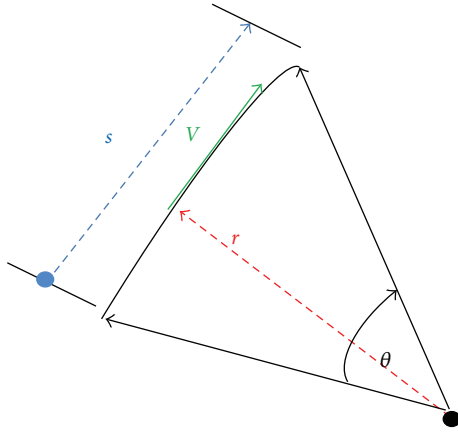


FIGURE 6: Constant thrust arc.

Then by taking the limit as the size of the arc approaches zero, the velocity is found to be

$$V = \frac{s}{t_{\text{step}}}, \quad (29)$$

where t_{step} is a small time step. Referring to the constant thrust algorithm shown in Algorithm 1 by substituting (28) in (29) the pitch angle calculation is shown in line (17). Thus, every time step the pitch angle is updated. Then using the energy conservation principle the new velocity is calculated as shown in line (19). Then position of the quadcopter is updated using the equations in line (22) and line (23). The final radius of curvature is calculated from the new the velocity as shown in line (24). In Algorithm 1 the lines from (2) to (15) show the initialisations used to start the constant thrust looping algorithm.

The numerical technique presented here is Euler integration. The error in the solution is approximately proportional to the time step [16]. Therefore by decreasing the time step the error is minimized.

Figure 7 shows three looping paths at constant normalised thrust $3.5g$ but different initial velocities. Note that all the paths are plotted for the same period of time ($t = 12$ s).

As shown in Figure 7 at lower initial velocities, the number of loops per unit time increases since the normalised thrust is able to steer the quadcopter at a fast pitch rate due to the presence of less amount of energy in the system. Thus the rate of change of the pitch angle and the rate of change of the radius of curvature will decrease as the initial velocity increases. Increasing the initial velocity will increase the looping height and radius.

Alternatively shown next the normalised thrust is changing from one path to the other while the initial velocity is kept constant. Figure 8 shows three looping paths all at the same initial velocity ($V = 10 \text{ ms}^{-1}$) but different normalised thrust.

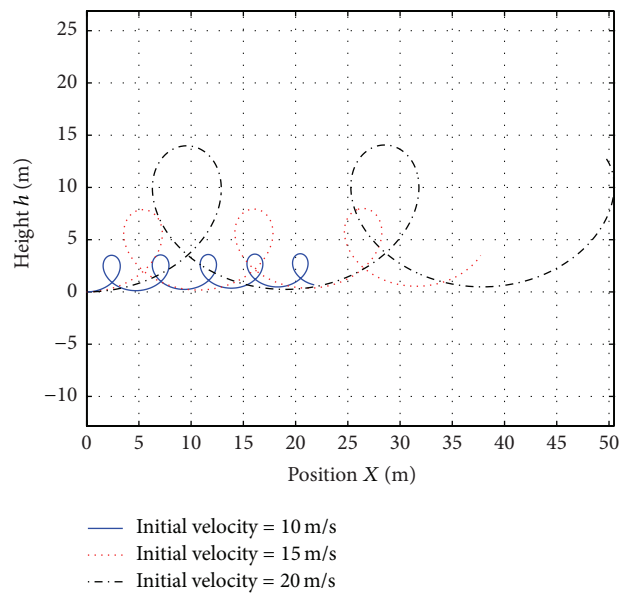
As shown in Figure 8 in the $2.5g$ and $3.5g$ looping there is enough normalised thrust to produce more centripetal acceleration capable of changing the direction of the velocity at a faster pitch rate. Thus, having a faster rate of change of the pitch angle will increase the rate of change of the local radius of curvature of the looping. As a result the quadcopter

```

(1) Initialisation;
(2)  $g = 9.81$ ;
(3)  $T = 14$ ;
(4)  $t_{\text{step}} = 0.005$ ;
(5)  $X = \text{zeros}(1, 2000)$ ;
(6)  $Z = \text{zeros}(1, 2000)$ ;
(7)  $\theta = 0$ ;
(8)  $\text{Energy} = \frac{V_{\text{initial}}^2}{2}$ ;
(9)  $V_x = V_{\text{initial}}$ ;
(10)  $V_z = 0$ ;
(11)  $V = \sqrt{V_x^2 + V_z^2}$ ;
(12)  $g_{\text{component}} = g \cos(\theta)$ ;
(13)  $r = \frac{V^2}{T - g_{\text{component}}}$ ;  $\triangleright$  radius of curvature
(14)  $n = 1$ ;
(15)  $t_{\text{end}} = 12$ ;
(16) while  $t < t_{\text{end}}$  do
(17)  $\theta_{n+1} = \frac{V * t_{\text{step}}}{r} + \theta_n$ ;
(18)  $g_{\text{component}} = g \cos(\theta)$ ;
(19)  $V = \sqrt{2 * (\text{Energy} - g * Z_n)}$ ;
(20)  $V_z = V \sin(\theta)$ ;
(21)  $V_x = V \cos(\theta)$ ;
(22)  $Z_{n+1} = Z_n + t_{\text{step}} * V_z$ ;
(23)  $X_{n+1} = X_n + t_{\text{step}} * V_x$ ;
(24)  $r = \frac{V^2}{T - g_{\text{component}}}$ ;
(25)  $t_{n+1} = t_n + t_{\text{step}}$ ;
(26)  $n = n + 1$ ;
(27) end

```

ALGORITHM 1: Constant thrust.

FIGURE 7: Varying the initial velocity at $3.5g$ normalised thrust for the same period of time.

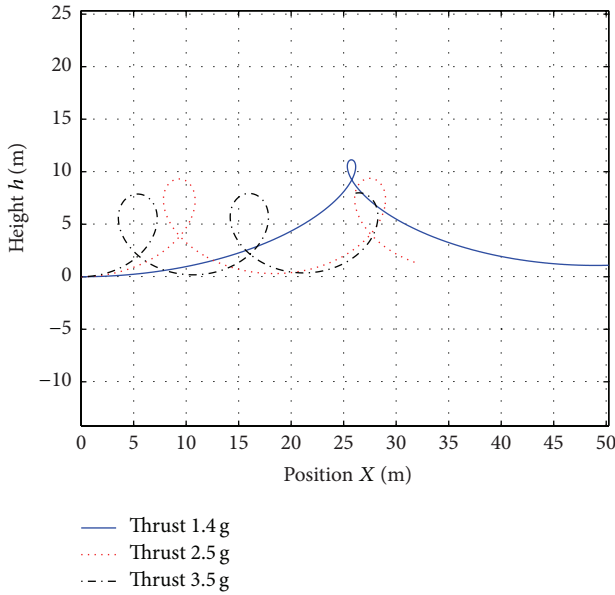


FIGURE 8: Varying the normalised thrust at equal initial velocities for the period of time ($t = 12$ s).

will have enough kinetic energy at the peak of the looping in order to move sideways and form a looping shape. The $1.4g$ path is not really considered as a looping it is rather a flip. All three paths start with the same initial kinetic energy, the difference in shape is specified according to how much of the initial kinetic energy is transferred to potential energy and how much remains as kinetic energy at the peak of the looping.

3. Simulation

In this section the tracking performance of the 6DOF quadcopter model is evaluated with the looping commands. The block diagram shown in Figure 9 illustrates the simulation process.

As shown in Figure 9, the initial velocity is input along with the B for clothoidal commands or T for the constant thrust commands. The trajectories generated are then transformed into quaternions and the control input U_1 . The error quaternion q_{err} is calculated from the quaternion difference between the command q_c and actual quaternion q_a . The error quaternion is then input to a PD controller and then the control signals are fed to the dynamics.

Since the perfect circular path is very demanding, therefore the perfect circle will not be implemented. The model parameters that are used in all simulations are shown in Table 1.

3.1. Clothoid Looping Commands. The following parameters were plugged in the clothoidal looping trajectory generation section and so the command thrust required and the command pitch angle are determined. The B is set to be equal to 5. The initial velocity is set to be the minimum initial velocity (refer to clothoid section) equal to 12.5 ms^{-1} .

TABLE 1: Looping parameters.

I_x	$7.5 * 10^{-3} \text{ m}^4$
I_y	$7.5 * 10^{-3} \text{ m}^4$
I_z	$1.3 * 10^{-2} \text{ m}^4$
I_p	$6 * 10^{-5} \text{ m}^4$
m	0.65 Kg

The time to perform the maneuver is 3 s. The proportional gain K_p and the derivative gain K_D are tuned to be equal 5 and 10, respectively.

The results are shown in Figures 10, 11, 12, and 13. Figures 10 and 11 show the command and state quaternion components q_0, q_2 as well as the command and state pitch angles.

As shown in Figure 10 the PD controller allows the state quaternion to track the command smoothly without any overshoot. Moreover due to the absence of any disturbances the tracking error is minimum. Note that the usage of quaternions eliminates the angular singularities. The quaternion components q_1 and q_3 are equal to zero since they are related to the roll and yaw angles.

Figure 11 is just a transformation of Figure 10 to illustrate the change in the pitch angle of the quadcopter along the loop. As shown in Figure 11 the PD control in the pitch angle controller allows the pitch to track the command smoothly without any overshoot.

Figure 12 shows the command normalised thrust over mass calculated from the clothoidal looping section.

As shown in Figure 12 the thrust curve is symmetrical. The first half of the curve represents the first spiral curve of the clothoid loop and the second represents the second spiral curve. Note that the normalised thrust reaches a maximum value of $4.5g$. An important point to be noticed is that the normalised thrust cannot reach zero at the peak of the looping since the pitching motion and the thrust are coupled. Therefore, if the thrust reaches zero at the peak, the quadrotor will lose its ability to pitch up. Finally Figure 13 shows a visualisation of the quadcopter doing a clothoidal looping.

As shown in Figure 13 the black dashed loop is the reference path and the blue solid loop is the actual loop. The quadcopter is first commanded to hover to 10 m. The blue loop is not identical to the black loop. This is due to the presence of the quadcopter dynamics. The pitch angle dynamics and the position dynamics are not identical, since the pitch dynamics are much faster than the position dynamics, the quadcopter did not track the desired trajectory. The addition of the damping term in the pitch angle controller slows down the pitch angle dynamics. Figure 14 shows the effect of the removal of the D-controller from the pitch angle controller.

As shown in Figure 14 the removal of the damping gain caused the pitch angle to change faster than the position dynamics, thus, leading the quadcopter to exit the maneuver at a higher altitude.

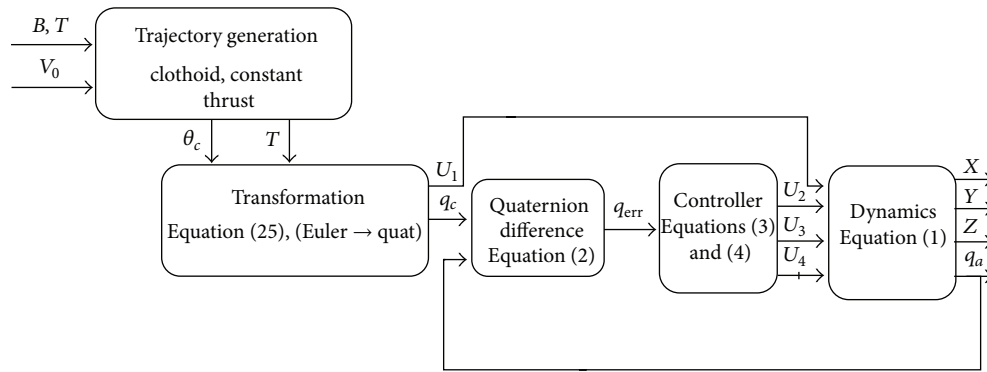


FIGURE 9: System block diagram.

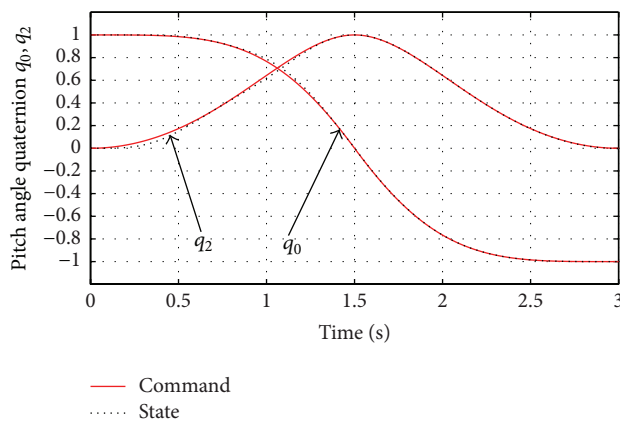


FIGURE 10: Command and state quaternion for a clothoid loop.

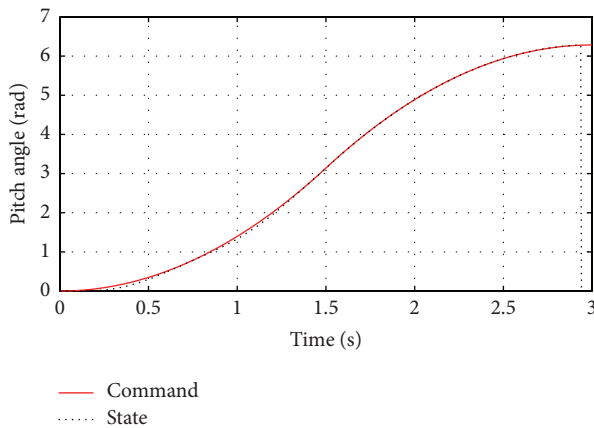


FIGURE 11: Command and state pitch angle for a clothoid loop.

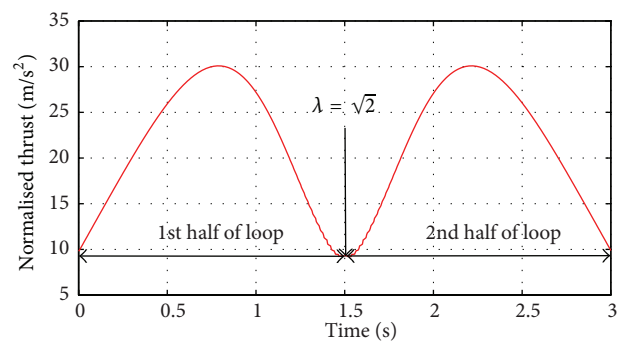


FIGURE 12: Command normalised thrust for a clothoid loop.

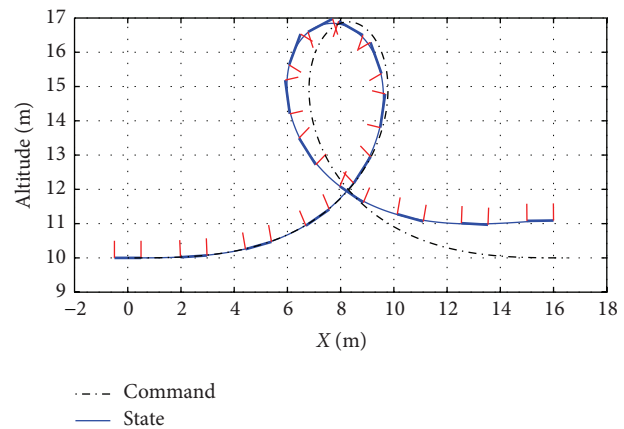


FIGURE 13: Quadcopter visualisation with PD controller.

3.2. Constant Thrust Looping. The following parameters were plugged in the constant looping thrust equations and so the command pitch angle is determined. The constant normalised thrust chosen for this simulation is $T = 20 \text{ m/s}^2$. The initial velocity is set to be 10 m/s . The time to perform the maneuver is 2.5 s . The proportional gain K_p and the derivative gain K_D are tuned to be equal to 5 and 10, respectively.

The results are shown in Figures 15, 16, and 17. Figures 15 and 16 show the command and state quaternion components q_0 , q_2 as well as the command and state pitch angles.

As shown in Figure 15 the state quaternion tracks the command smoothly. The quaternion components q_1 and q_3 are equal to zero since they are related to the roll and yaw angles.

Due to the absence of external disturbance and the presence of a smooth PD controller, the state pitch angle

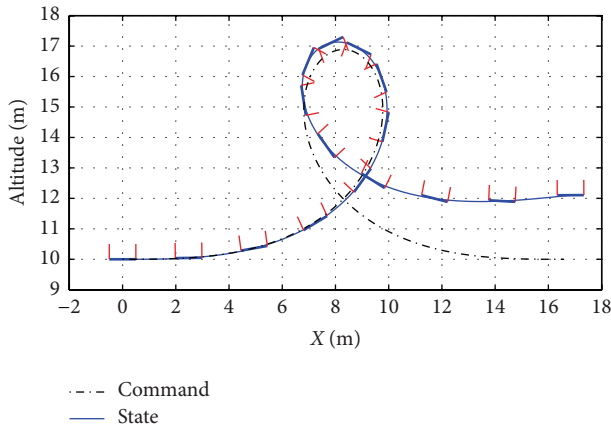


FIGURE 14: Quadcopter visualisation with P controller.

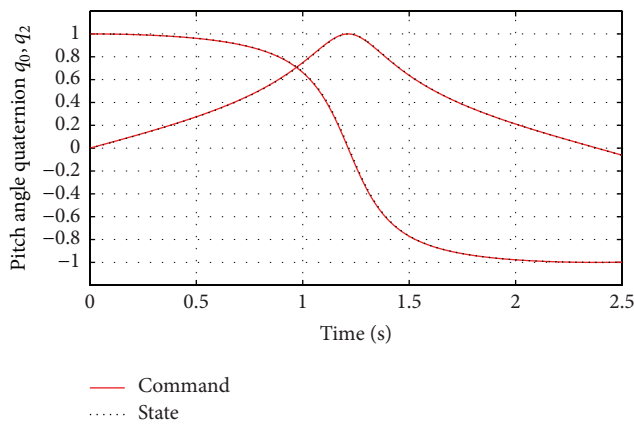


FIGURE 15: Command and state quaternion for a constant thrust loop.

tracks the command smoothly with minimum error as shown in Figure 16.

Figure 17 shows the visualisation of the quadcopter while performing the constant thrust looping.

As shown in Figure 17 the black dashed loop is the reference path and the blue solid loop is the actual loop. The quadcopter is first commanded to hover to 10 m. The blue loop nearly identical to the black loop. The position and pitch dynamics do not affect much the shape of the looping since in this case normalised thrust (acceleration) is always at steady state.

4. Conclusion

The trajectories designed using energy allowed the quadcopter to cross the physical singular configurations smoothly. The three looping paths and trajectories designed were found to be very promising in extending the flight envelope of the quadcopter in order to perform aggressive looping maneuver. The perfect circular looping is found to be the most demanding as it required an entry thrust equal to $6g$. A clothoid looping was found to be less demanding and as it requires normalised thrust equal to $3.5g$. Finally a new

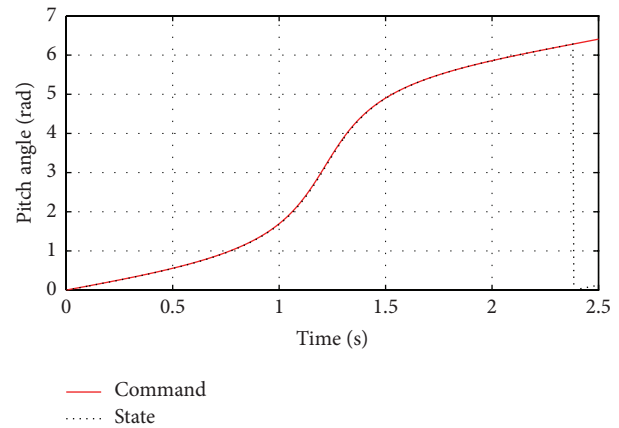


FIGURE 16: Command and state pitch angle for a constant thrust loop.

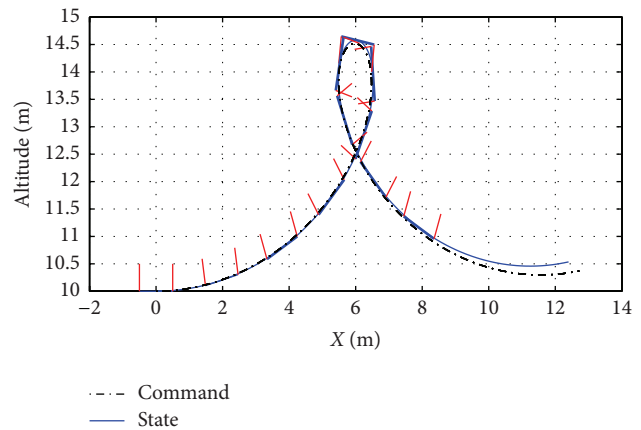


FIGURE 17: Visualisation of constant thrust loop.

innovative approach is proposed where the thrust is specified and then the path is determined. This method used a constant thrust approach which might be the most applicable since the looping maneuver depends on the design specifications of the quadcopter. A quadcopter can perform a constant thrust looping at normalised thrust starting from $1.4g$.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] E. J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems*, vol. 1, Allyn & Bacon, Boston, Mass, USA, 1989.
- [2] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, p. 1516, 2006.
- [3] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

- [4] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadcopter multi-flips," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 1642–1648, IEEE, Anchorage, Alaska, USA, May 2010.
- [5] S. Lupashin and R. D'Andrea, "Adaptive openloop aerobatic maneuvers for quadcopters," in *Proceedings of the 18th IFAC World Congress*, pp. 2600–2606, Milan, Italy, August–September 2011.
- [6] S. Lupashin and R. D'Andrea, "Adaptive fast open-loop maneuvers for quadcopters," *Autonomous Robots*, vol. 33, no. 1-2, pp. 89–102, 2012.
- [7] T. Lee, M. Leok, and N. H. McClamroch, "Control of complex maneuvers for a quadrotor UAV using geometric methods on SE (3)," <http://arxiv.org/abs/1003.2005>.
- [8] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 2520–2525, Shanghai, China, May 2011.
- [9] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 3277–3282, IEEE, Kobe, Japan, May 2009.
- [10] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: autonomous quadrotor aerobatics in theory and practice," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 1649–1654, IEEE, Anchorage, Alaska, USA, May 2010.
- [11] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.
- [12] D. Mellinger, M. Shomin, and V. Kumar, "Control of quadrotors for robust perching and landing," in *Proceedings of the International Powered Lift Conference*, pp. 205–225, 2010.
- [13] M. Piedmonte and E. Feron, "Aggressive maneuvering of autonomous aerial vehicles: a human-centered approach," in *Proceedings of the International Symposium of Robotics Research*, vol. 9, pp. 413–420, London, UK, 2000.
- [14] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in *Proceedings of the 12th European Control Conference (ECC '13)*, pp. 3864–3869, Zurich, Switzerland, July 2013.
- [15] R. Müller, "Roller coasters without differential equations—a Newtonian approach to constrained motion," *European Journal of Physics*, vol. 31, no. 4, pp. 835–848, 2010.
- [16] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, vol. 1, John Wiley & Sons, New York, NY, USA, 2003.