Trajectory Generation and Control for Quadrotors

Sandesh Thapa
MAE/ECEN 5473 Digital Control Systems
School of Mechanical and Aerospace Engineering
Oklahoma State University
Stillwater, Oklahoma
Email: sandesh.thapa@okstate.edu

Abstract

In this project, I explored the controller design and trajectory generation for quadrotor. Differential flatness was used for trajectory generation based on the one of the references. Minimum snap trajectory is generated from the given algorithm based on waypoints navigation and a simple proportional plus derivative (PD)controller is used to track the trajectories. Simulations results are presented which generates smooth trajectories for position and velocity of the quadrotor.

I. Introduction

Recently, quadrotors are being widely used for research. Because of it's agility in 3D Euclidean space, it can be used for variety of purpose including photography, inspection, transportation of loads, geo-surveying and so on. For a quadrotor maneuvering in tightly constraint environment, it is necessary to develop optimal trajectories to ensure safe passage through corridors and constraints [1]. So, it is necessary to develop flight plans that leverage the dynamics of system [1].

II. Project Objective

The objective of this project is to review and develop understanding of trajectory generation for quadrotors. I reviewed [1] and related sources [2], [3] to understand and implement differential flatness for trajectory generation Examples code were given at [4], [5]. I implemented Mellinger's minimum snap trajectory and control algorithm for quadrotors based on differential flatness [1], [2]. The goal of the this project was to develop a strong understanding of quadrotor trajectory generation and control algorithm so that it can be applied to mutliagents path planning and control problems in future works.

III. Methodology

In this paper, I used the dynamic model and control from [1] and implemented in MATLAB. I used a PD controller to track a given trajectory. Trajectories are generated based differential flatness which ensures smoothness. I compared minimum snap trajectory and shortest distance trajectory and distinguished their performance.

IV. Dynamic Model

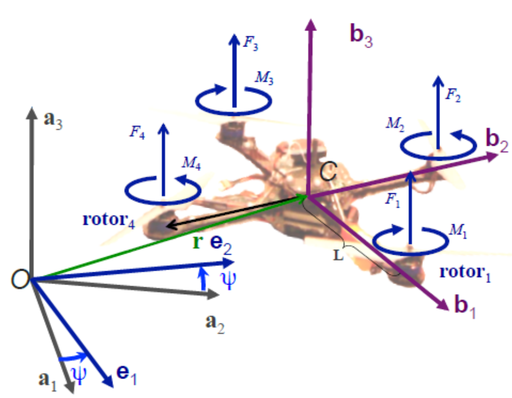


Fig. 1. Quadrotor model with the body-fixed and inertial reference frames.

A. Coordinate Systems

The coordinate systems with a free body diagram for the quadrotor is shown in Fig. 4. $\mathcal{A}$ is the inertial frame with axis a_1, a_2 and a_3. The body frame $\mathcal{B}$, is attached to the center of mass of the quadrotor which is coincident with the inertial frame. $\mathcal{C}$ is the center mass of the rotor and L is the length of the rotors from the center $\mathcal{C}$

B. Motor Model

Each rotor has an angular speed ω_i and produces a vertical force F_i, which is given by

$$F_i = k_F \omega_i^2, \tag{1}$$

$k_F \approx 6.11 \times 10^{-8} \frac{N}{rmp^2}$ [3].

$$M_i = k_M \omega_i^2, \tag{2}$$

$k_M \approx 1.5 \times 10^{-9} \frac{Nm}{rmp^2}$ [3].

C. Rigid Body Dynamics

The orientation of quadrotor is described by triplet of yaw-pitch-roll(ϕ, θ and ψ) $Z-Y-X$ Euler angles. The position of center of mass of quadrotor is given by

$$\mathbf{r} = \begin{bmatrix} x & y & z \end{bmatrix}^T. \tag{3}$$

The state of the quadrotor is given by

$$q = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T. \tag{4}$$

The rotation matrix from $\mathcal{B}$ to $\mathcal{A}$ is given by,

$$R_{BA} = \begin{bmatrix} c\phi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}. \tag{5}$$

The angular velocity of the robot in the body frame is denoted as q, q and r. These values are related to derivatives of the roll, pitch and yaw angles according to

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{6}$$

1) Newton's Equations of Motion: The force in the system are gravity, and the forces from each rotors F_i. The equation governing the acceleration of center of mass of quadrotor is

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R_{BA} \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}, \tag{7}$$

The first input to the system u_1 is defined as

$$u_1 = \sum_{i=1}^{4} F_i$$

.

2) Euler's Equations of Motion: In addition to forces, each rotor produces a moment perpendicular to the plan to rotation of the blade, M_i. Rotors 1 and 3 rotate in the $-\mathrm{b}_3$ direction while 2 and 4 rotate in the $+\mathrm{b}_3$ direction. The moment produced on the quadrotor is opposite to the direction of rotation of the blades. So, M_1 and M_3 act in the b_3 direction while M_2 and M_4 act in the $-\mathrm{b}_3$ direction. The angular acceleration is determined by,

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{8}$$

We can rewrite this as:

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{9}$$

where $\gamma = \frac{k_M}{k_F}$. Second input $\mathbf{u}_2$ is defined as:

$$\mathbf{u}_2 = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}.$$

V. CONTROL

The controller used in [1], [3] are derived by linearizing the equation of motion about an operating point, $\mathbf{r} = \mathbf{r}_0$, $\theta = \phi = 0$, $\psi = \psi_0$, $\dot{\mathbf{r}} = 0$, and $\dot{\phi} = \dot{\theta} = \dot{\psi} = 0$. Using small angle approximation for roll and pitch angles ($c\phi = 1$, $c\theta = 1$, $s\phi \approx \phi$, and $s\theta \approx \theta$) gives use the following linearized equations:

$$F_{i,0} = \frac{mg}{4}, \tag{10}$$

At nominal state, the inputs for hover are $u_{1,0} = mg$, $\mathbf{u}_{2,0} = 0$.

Linearizing (7), we get:

$$\begin{aligned}
\ddot{r}_1 &= g(\Delta\theta \cos\psi_0 + \Delta\phi \sin\psi_0) \\
\ddot{r}_2 &= g(\Delta\theta \sin\psi_0 - \Delta\phi \cos\psi_0) \\
\ddot{r}_3 &= \frac{1}{m}u_1 - g
\end{aligned} \tag{11}$$

Linearizing (9), we get:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = I^{-1} \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{12}$$

The rotor cart is assumed to be symmetric so $I_{xx} = I_{yy}$,

$$\begin{aligned}
\dot{p} &= \frac{u_{2,x}}{I_{xx}} = \frac{L}{I_{xx}}(F_2 - F_4) \\
\dot{q} &= \frac{u_{2,y}}{I_{yy}} = \frac{L}{I_{yy}}(F_3 - F_1) \\
\dot{r} &= \frac{u_{2,z}}{I_{zz}} = \frac{\gamma}{I_{zz}}(F_1 - F_2 + F_3 - F_4)
\end{aligned} \tag{13}$$

A. Position and Attitude Controller

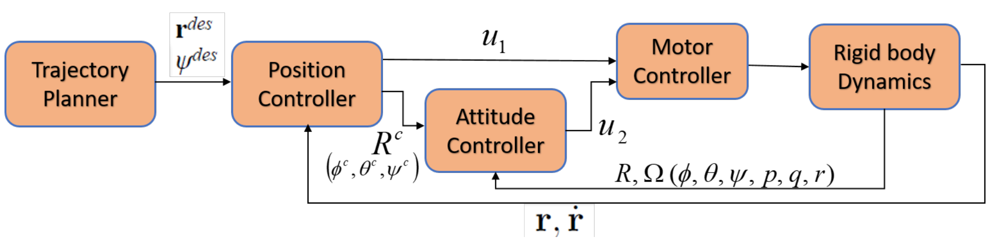


Fig. 2. The position and attitude control loops.

The control problem is to determine the four inputs, $(u_1, \mathbf{u}_2)$ required to hover or to follow a desired trajectory, $\mathbf{z}_des$. As shown in Fig. 2, error in the robot's position is used to derive the controller from (11) The equations in (11) also allows us to derive a desired orientation. The attitude controller for this desired orientation is derived from (12). We require the attitude control loop to run in a higher rate than the position control loop.

1) Attitude Control: Mellinger [1] develop a following PD controller for attitude control:

$$\mathbf{u}_2 = \begin{bmatrix} k_{p,\phi}(\phi_{des} - \phi) + k_{d,\phi}(p_{des} - p) \\ k_{p,\theta}(\theta_{des} - \theta) + k_{d,\theta}(q_{des} - q) \\ k_{p,\psi}(\psi_{des} - \psi) + k_{d,\psi}(r_{des} - r) \end{bmatrix}. \tag{14}$$

here, $k_{p,i}$ and $k_{d,i}$ are the control gains that needs to be tuned.

2) Position Control: If a desired trajectory is given as:

$$\mathbf{z}_des = \begin{bmatrix} \mathbf{r}_{des}(t) \\ \psi_T(t) \end{bmatrix},$$

following PD control is used to track the yaw angles and 3 position of the rotor.

$$\ddot{r}_{i,c} = \ddot{r}_{i,des} + k_{d,i}(\dot{r}_{i,des} - \dot{r}_i) + k_{p,i}(r_{i,des} - r_i). \tag{15}$$

which gives us

$$u_1 = m(g + \ddot{r}_{3,c}). \tag{16}$$

we find the desired roll and pitch angles as

$$\begin{aligned} \phi_{des} &= \frac{1}{g}(\ddot{r}_{1,c} \sin \psi_{des} - \ddot{r}_{2,c} \cos \psi_{des}), \\ \theta_{des} &= \frac{1}{g}(\ddot{r}_{1,c} \cos \psi_{des} + \ddot{r}_{2,c} \sin \psi_{des}.) \end{aligned} \tag{17}$$

The desired roll and pitch velocities are taken to be zero.

VI. Differential Flatness

Definition 1: A nonlinear system

$$\begin{aligned} &\dot{x} = f(x, u), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m, \\ &y = h(x), \qquad y \in \mathbb{R}^m, \end{aligned} \tag{18}$$

is differentially flat if we can find outputs $z \in \mathbb{R}^m$ of the form

$$z \; \zeta(x, u, \dot{u}, ..., u^{(l)}), \tag{19}$$

such that

$$\begin{aligned} x &= x(z, \dot{z}, ..., z^{(l)} := x(\bar{z}), \\ u &= u(z, \dot{z}, ..., z^{(l)}) := u(\bar{z}). \end{aligned} \tag{20}$$

A non linear system is differentially flat if we can find a set of outputs such that we can express all states and inputs in terms of those outputs and their derivatives [7].

A. Example

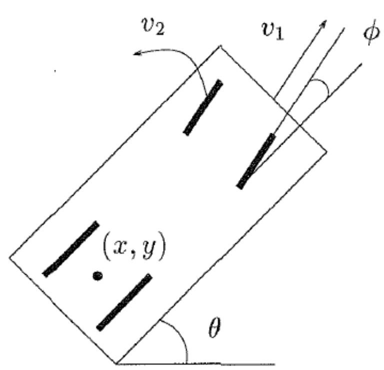


Fig. 3. Kinematic Car

A kinematic car is differentially flat system as whown in [8]
Equations of motion

$$\begin{aligned} \dot{x} &= \cos\theta \cos\phi v_1 \\ \dot{y} &= \sin\theta \cos\phi v_2 \\ \dot{\theta} &= \frac{1}{l} \sin\phi v_1 \\ \phi &= v_2 \end{aligned} \tag{21}$$

where, v_1 and v_2 are the inputs of the system. If output is chosen to be (x, y), ϕ can be expressed in terms of flat outputs as

$$\phi \, \arctan(\frac{1}{l}(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}, \dot{x}\ddot{x} - \dot{y}\ddot{y}), \tag{22}$$

which shows that kinematic car is differentially flat system.

B. Quadrotor a Differentially Flat System

From (7), we get

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R_{BA} \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} \tag{23}$$

Using linearized equation of motions near hover assuming small angle assumptions we get,

$$\begin{aligned} m\ddot{x} &= (\theta c\psi + \phi s\psi)u_1 \\ m\ddot{y} &= (\theta s\psi - \phi c\psi)u_1 \\ m\ddot{z} &= -mg + u_1 \end{aligned} \tag{24}$$

It shows the second derivative of position is proportional to input u_1.

From the angular rate equation (6), we get:

$$\begin{aligned} p &= \dot{\phi}c\theta - \dot{\psi}c\phi s\theta \\ q &= \dot{\theta} + \dot{\psi}s\phi \\ r &= \dot{\phi}s\theta + \dot{\psi}c\phi c\theta \end{aligned} \tag{25}$$

Substituting the approximation:

$\sin(\theta) \approx \theta, \sin(\phi) \approx \phi, \cos(\phi) = \cos(\theta) \approx 1$, we get:

$$\begin{aligned} p &= \dot{\phi} - \dot{\psi}\theta \\ q &= \dot{\theta} + \dot{\psi}\phi \\ r &= \dot{\phi}\theta + \dot{\psi} \end{aligned} \tag{26}$$

Further substituting:

$\dot{\psi}\theta \approx \dot{\psi}\phi \approx \dot{\phi}\theta = 0$, we get:

$$\begin{aligned} p &= \dot{\phi} \\ q &= \dot{\theta} \\ r &= \dot{\psi} \end{aligned} \tag{27}$$

Now only considering the principal axis for inertia of the quadrotor and using (9), we get:

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{q} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} u_{2,x} \\ u_{2,y} \\ u_{2,z} \end{bmatrix} - \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{28}$$

From (28), we get:

$$\begin{aligned} I_{xx}\dot{p} &= u_{2,x} - I_{yy}qr + I_{zz}qr \\ I_{yy}\dot{q} &= u_{2,y} + I_{xx}pr - I_{zz}pr \\ I_{xx}\dot{r} &= u_{2,z} - I_{xx}pq + I_{zz}pq \end{aligned} \tag{29}$$

Using approximation, $\dot{\psi}\dot{\theta} \approx \dot{\psi}\dot{\phi} \approx \dot{\phi}\dot{\theta} = pq = pr = qr = 0$,

$$\begin{aligned} I_{xx}\dot{p} &= u_{2,x} \\ I_{yy}\dot{q} &= u_{2,y} \\ I_{xx}\dot{r} &= u_{2,z} \end{aligned} \tag{30}$$

From (27) and (31), we get :

$$\begin{aligned} \ddot{\phi} &= \frac{u_{2,x}}{I_{xx}} \\ \ddot{\theta} &= \frac{u_{2,y}}{I_{yy}} \\ \ddot{\psi} &= \frac{u_{2,z}}{I_{xx}} \end{aligned} \tag{31}$$

Differentiating (24),

$$\dddot{x} = \Big(\theta c\psi + \phi s\psi\Big)\dot{u_1} + \Big(\dot{\theta}c\psi - \psi\dot{\psi} + \dot{\phi}s\psi + \phi c\psi\dot{\psi}\Big)u_1 \tag{32}$$

Differentiating again,

$$\ddddot{x} = \Big(\theta c\psi + \phi s\psi)\Big)\ddot{u_1} + 2\Big(\dot{\theta} c\psi - \psi\dot{\psi} + \dot{\phi} s\psi + \phi c\psi\dot{\psi}\Big)\dot{u_1} + \Big(\ddot{\theta} c\psi - \dot{\theta} s\psi\dot{\psi} - \theta s\psi\ddot{\psi} - \theta s\psi\dot{\psi}^2 + \ddot{\phi} s\psi + \dot{\phi} c\psi\dot{\psi} + \dot{\phi} c\psi\ddot{\psi} - \dot{\phi} c\psi\dot{\psi}^2\Big)u_1 \tag{33}$$

After substituting $\ddddot{x}$ into (30), we can prove that the fourth derivative of position is proportional to u_2. So,it is proved that quadrotor is a differentially flat system [1] [4].

VII. 3D Trajectory Generation

For the purpose of this simulation, I used the optimal trajectory defined in [1] For optimal trajectory following function is used based on Calculus of Variations,

$$x*(t) = arg \min_{x(t)} \int_0^T L(\dot{x}, xt)dt \tag{34}$$

Solving functional L using Euler-Lagrange equation gives use the optimal function.

$$ddt\Big(\frac{\partial L}{\partial \dot{x}}\Big) - \frac{\partial L}{\partial x} = 0 \tag{35}$$

For smooth trajectories generation, Mellinger [1], use following functional

$$x*(t) = arg \min_{x(t)} \int_0^T ((x^n)^2)dt \tag{36}$$

when n = 1, we get shortest distance with minimum velocity, n = 2 gives minimum acceleration, n = 3 gives minimum acceleration and n = 4 produces minimum snap trajectory. For rest of the paper, I will be using minimum snap trajectory for simulation purpose.

A. Minimum Snap Trajectory

Minimum snap trajectory is 7th order polynomial of time [4], [1]. If we are given a set of $n+1$ waypoints $w_0, ...w_n$, the minimum snap trajectory is a piecewise polynomial composed of n 7th order polynomials. Each polynomial piece p_i travels between a pair of waypoints w_{i-1} and w_i and takes a known amount of time T_i to complete $i = 1, ..., n$.

Let $S_0 = 0$ and for $i = 1, ..., n, S_i = \sum_{k=1}^{i} T_k$. S_i is the time it takes to reach waypoint w_i from waypoint w_0. Then the polynomial p_i has the form:

$$p_i(t) = \alpha_{i0} + \alpha_{i1}\frac{t - S_{-i1}}{T_i} + \alpha_{i2}\Big(\frac{t - S_{-i1}}{T_i}\Big)^2 + ... + \alpha_{i7}\Big(\frac{t - S_{-i1}}{T_i}\Big)^7 \tag{37}$$

To obtain the complete equation for piecewise trajectory, we need to solve for all the coefficient $\approx_{ij}$. There are $8n$ such coefficients. These coefficient must satisfy a series of constraint. First the polynomial must go thought all the way points

$$p_i(S_{i-1}) = w_{i-1} \quad \text{and} \quad p_i(S_i) = w_i \quad \text{for all i = 1,...,n (2n constraints)} \tag{38}$$

Second, velocity, acceleration, jerk are zero at the end points

$$p_1^{(k)}(S_0) = p_n^{(k)} = 0 \quad \text{and} \quad \text{for al k = 1,.., 3 (6 constraints)} \tag{39}$$

Third, Velocity, acceleration, 3rd to 6th derivatives are continuous

$$p_1^{(k)}(S_i) = p_{i+1}^{(k)} \quad \text{and} \quad \text{for al k = 1,.., 6 (6n-6 constraints)} \tag{40}$$

The coefficient can be solved by converting the equations in matrix form

$$A\alpha = b \tag{41}$$

The coefficients can be obtained by solving

$$\alpha = A^{-1}b \tag{42}$$

VIII. Simulation Results

In this section, results obtained from minimum snap trajectory generation is included. Examples code were given in [4] [5], I implemented trajectory generation and control algorithm in their example code. The model used is ASTEC hummingbird quadrotor. The mass of the rotor is 0.18 kg, the length (L) of the rotor from center of mass to motors is 0.086 m, principal moment of inertia are $I_{xx} = 0.00025kgm^2$, $I_{yy} = 0.000232kgm^2$, $I_{zz} = 0.0003738kgm^2$

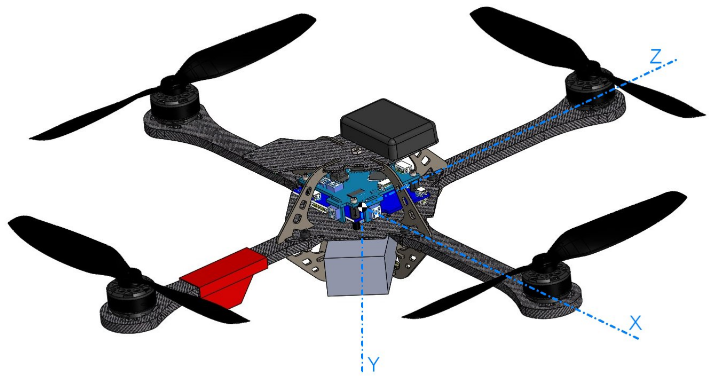
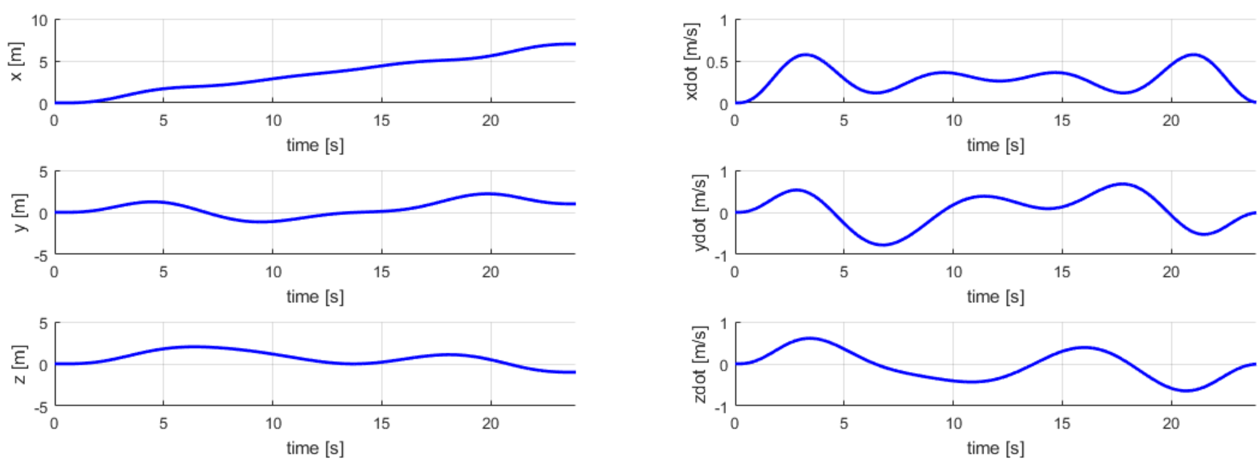
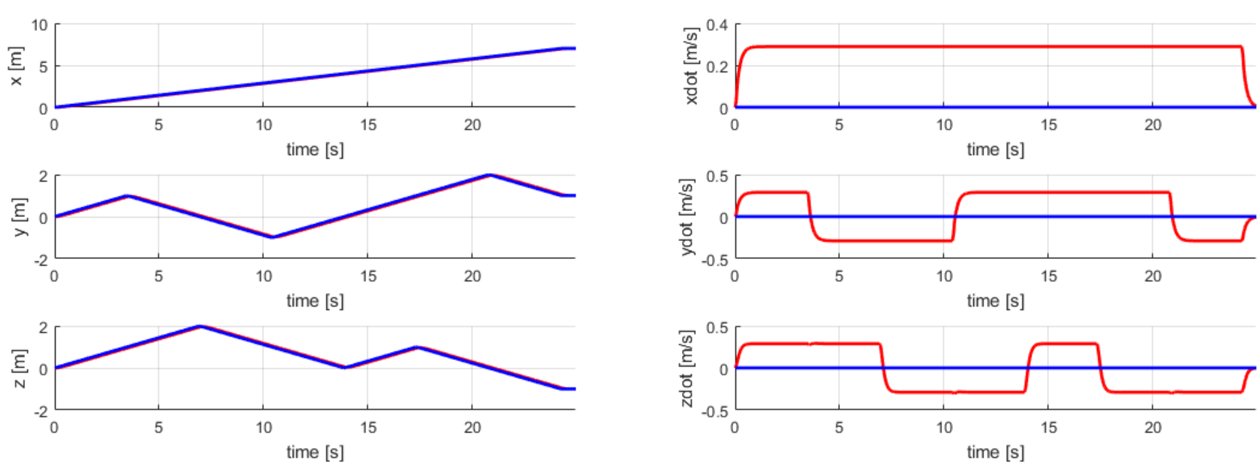


Fig. 4. ASTEC Humming quadrotor used in paper for simulation purpose [1] [6]



(a) (b)

Fig. 5. Position and velocity trajectories for quadrotor based on minimum snap. All the trajectories are smooth and based on differential flatness.



(a) (b)

Fig. 6. Position and velocity trajectories for quadrotor without minimum snap. The points are connected as linearly with each other based on shortest distance (minimum velocity).

As we can observe from the 5, that the trajectories are smooth and there is no discontinuity in the system. The smooth trajectories allows better performance and planning of quadrotor. However, in 6, it can be observed that the trajectories are not continuous, it passes through specified waypoints but because of sudden jump in the velocity and position graphs it's not smooth. Also, because of inertia present in the system, it may not be feasible to make such sharp turn.

IX. Conclusion

From this project I learned interesting application of trajectory generation for quadrotors. Minimum snap trajectories can be used for aggressive maneuvers and perching and still system will have no any perturbations. This is one of the important concept, I learned from this project. The trajectories are always smooth and optimal. One can selected waypoints and introduce more dynamic behavior to the quadrotor. I plan to use differential flatness based trajectory generation for my future work and test in the experimental platform with one of our rotors. I also plan to extend it to multi-agents motion planning and control.

References

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

[2] 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.

[3] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[4] V. K. U. of Pennsylvania. (2017) Aerial robotics. [Online]. Available: https://www.coursera.org/learn/robotics-flight/

[5] (2017) Aerial robotics. [Online]. Available: https://github.com/WenjinTao/Robotics--Coursera/tree/master/Robotics-Aerial_Robotics/

[6] ASTEC. (2017). [Online]. Available: http://wiki.asctec.de/display/AR/CAD+Models/master/

[7] M. J. Van Nieuwstadt and R. M. Murray, "Real time trajectory generation for differentially flat systems," 1997.

[8] M. J. Van Nieuwstadt, "Trajectory generation for nonlinear control systems," Ph.D. dissertation, California Institute of Technology, 1997.