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 explored and used for trajectory generation based on one of the references. Minimum snap trajectory is generated from the given algorithm based on waypoints navigation and a simple proportional plus derivative (PD) is used to track the trajectories. Simulation results are presented which generate smooth trajectories for position and velocity of the quadrotor.

I. INTRODUCTION

Recently, quadrotors have been widely used for research. Because of it’s agility and mobility 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 object 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 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.

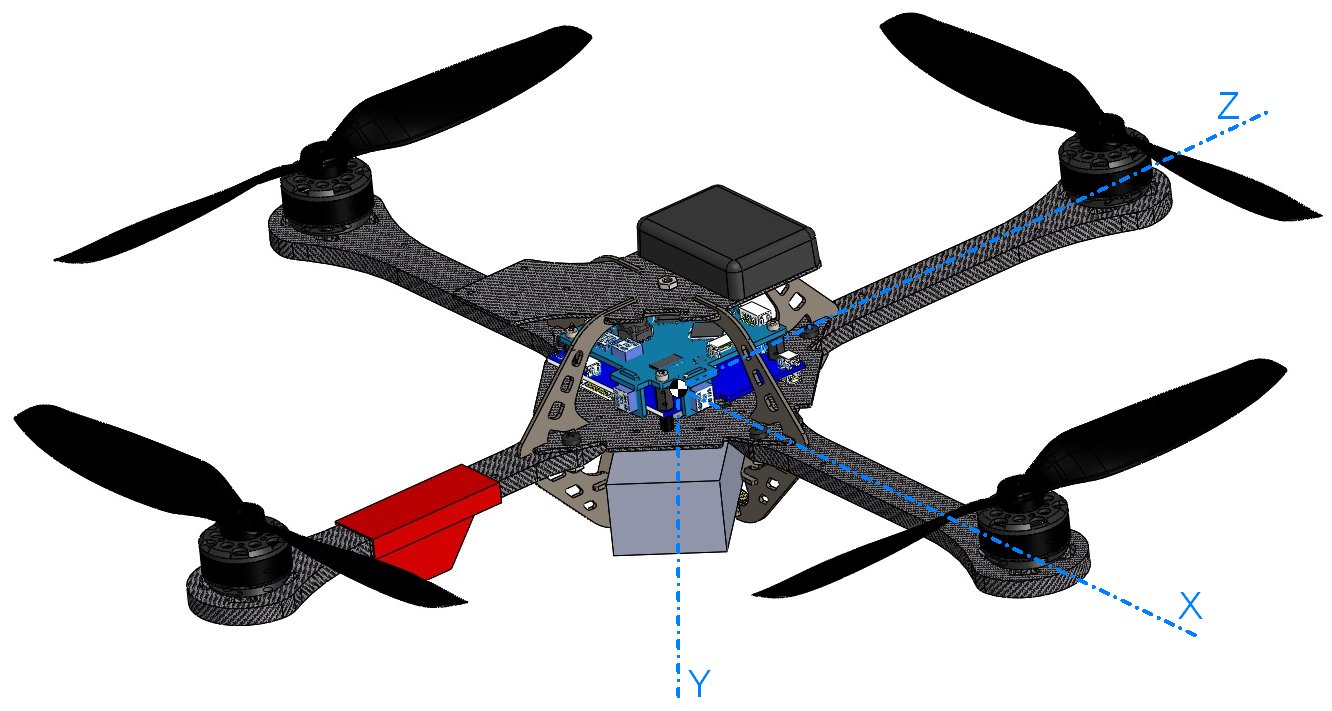
III. METHODOLOGY

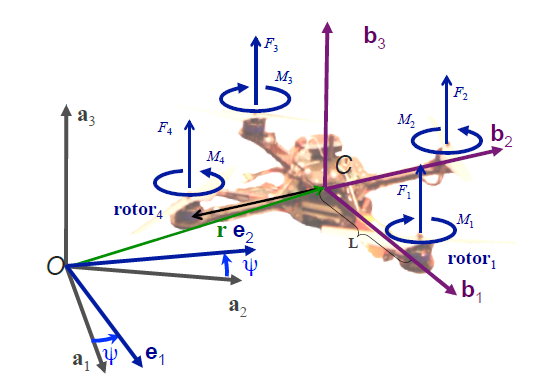
In this paper, I used the dynamic model and control from [1] and implemented in MATLAB. The dynamic model is of ASTEC hummingbird quadrotor, I used a PD controller to track a given trajectory. Trajectories are generated based differential flat which ensures smooth generation of trajectories. I compared minimum snap trajectory and shortest distance trajectory and distinguished their performance.

Fig. 1. ASTEC Humming quadrotor used in paper for simulation purpose [1] [6] IV. DYNAMIC MODEL

*A. Coordinate Systems*

The coordinate systems with a free body diagram for the quadrotor is shown in Fig. 2. A is the inertial frame with axis a1, a2 and a3. The body frame B, is attached to the center of mass of the quadrotor which is coincident with the inertial frame. C is the center mass of the rotor and L is the length of the rotors from the center C





*B. Motor Model*

k ≈6.11×10−8 N [3]. F rmp2

k ≈1.5×10−9 Nm [3]. M rmp2

*C. Rigid Body Dynamics*

M i = k M ω i2 (2)

Fig. 2. Quadrotor model with the body-fixed and inertial reference frames.

Each rotor has an angular speed ωi and produces a vertical force Fi, which is given by F i = k F ω i2 (1)

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

The state of the quadrotor is given by

The rotation matrix from B to A is given by,

cφcθ − sφsψsθ RBA = cθsψ + cψsφsθ

−cφsθ 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

p cθ 0 −cφsθφ ̇ q=01 sφθ ̇ (6)

r= x y z T. (3) q= x y z φ θ ψ T. (4)

−cφsψ cψsθ + cθsφsψ cφcψ sψsθ − cψcθsφ . (5)

sφ cφcθ

ψ ̇ 00

r sθ 0 cφcθ *1) Newton’s Equations of Motion:* The force in the system are gravity, and the forces from each rotors Fi. The equation

governing the acceleration of center of mass of quadrotor is m ̈r =  0  + RBA  0  (7)

The first input to the system u1 is defined as

mg F1 +F2 +F3 +F4

4 u1 = Fi

i=1

*2) Euler’s Equations of Motion:* In addition to forces, each rotor produces a moment perpendicular to the plan to rotation of the blade, Mi. Rotors 1 and 3 rotate in the −b3 direction while 2 and 4 rotate in the +b3 direction. The moment produced oon the quadrotor is opposite to the direction of rotation of the blades. So, M1 and M3 act in the b3 direction while M2 and M4 act in the −b3 directio. The angular acceleration is determined by,

We can rewrite this as:

 p ̇   L ( F 2 − F 4 )   p   p  Iq ̇= L(F3 −F1) −q×Iq (8)

r ̇ M1−M2+M3−M4 r r

p ̇  0 L 0 −L p p Iq ̇=−L 0 L 0 −q×Iq (9)

r ̇ γ−γγ−γ r r where γ = kM . Second input u2 is defined as:

kF



At nominal state, the inputs for hover are u1,0 = mg, u2,0 = 0. Linearizing (7), we get:

0 L 0 −LF1 u2= −L 0 L 0 F2

   F 3  γ −γ γ −γ F4

V. CONTROL

The controller used in [1], [3] are derived by linearizing the equation of motion about an operating point, r = r0, θ = φ = 0, ψ = ψ0, r ̇ = 0, and φ ̇ = θ ̇ = ψ ̇ = 0. Using small angle approximation for roll and pitch angles (cφ = 1, cθ = 1, sφ ≈ φ, and sθ ≈ θ) gives use the following linearized equations:

Fi,0 = mg, (10) 4



r ̈1 =g(∆θcosψ0 +∆φsinψ0) r ̈2 =g(∆θsinψ0 −∆φcosψ0)

r ̈ 3 = 1 u 1 − g m

p ̇ 0L0−LF1      F 3 

(11)



Linearizing (9), we get:

The rotor cart is assumed to be symmetric so Ixx = Iyy, p ̇=u2,x = L(F2−F4)

*A. Position and Attitude Controller*

q ̇ =I−1 −L 0 L 0 F2 (12) r ̇ γ −γ γ −γ F4

Ixx Ixx q ̇=u2,y = L(F3−F1) (13)

Iyy Iyy r ̇ = u2,z = γ (F1 − F2 + F3 − F4)

Izz Izz

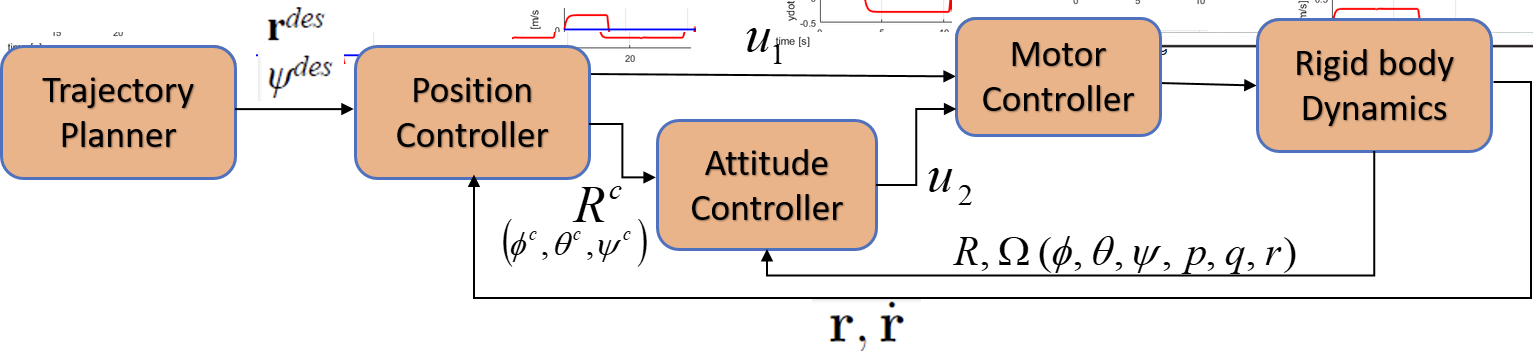


Fig. 3. The position and attitude control loops.

The control problem is to determine the four inputs, (u1,u2) required to hover or to follow a desired trajectory, zdes. As shown in Fig. 3, 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:

kp,φ(φdes − φ) + kd,φ(pdes − p) u2 =  kp,θ(θdes − θ) + kd,θ(qdes − q)  (14)

kp,ψ (ψdes − ψ) + kd,ψ (rdes − r) here, kp,iandkd,i are the control gains that needs to be tuned.

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

rdes (t) zdes= ψT(t) ,

following PD control is used to track the yaw angles and 3 position of the rotor. r ̈i,c = r ̈i,des + kd,i(r ̇i,des − r ̇i) + kp,i(ri,des − ri) (15)

which gives us we find the desired roll and pitch angles as

u1 =m(g+r ̈3,c) (16)

φdes = 1(r ̈1,c sinψdes −r ̈2,c cosψdes) g

θdes = 1(r ̈1,c cosψdes +r ̈2,c sinψdes) g

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

(17)

(18)

*Definition 1:* A nonlinear system

x ̇ =f(x,u), x∈Rn, u∈Rm y = h(x), y ∈ Rm

is differentially flat if we can find outputs z ∈ Rm of the form z ζ(x,u,u ̇,...,u(l)) (19)

such that

x = x(z, z ̇, ..., z(l) := x(z ̄)

u = u(z, z ̇, ..., z(l)) := u(z ̄) 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*

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

x ̇ =cosθcosφv1 y ̇ = sin θ cos φv2

θ ̇ = 1 sin φv1 l

(21)

(20)

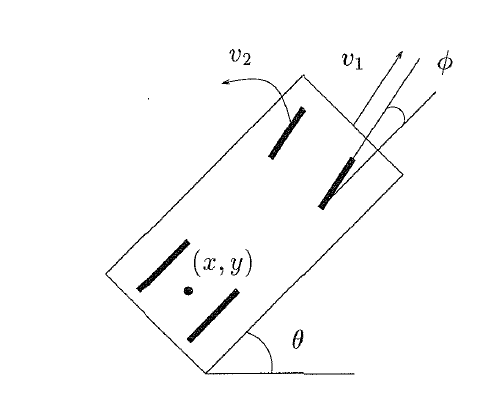


φ = v2 If output is (x, y), φ can be expressed in terms of flat outputs as

1

3 φarctan(l(x ̇2 +y ̇2)2 ,x ̇x ̈−y ̇y ̈) (22)



*B. Quadrotor a Differentially Flat System*

From (7), we get

Fig. 4. Kinematic Car

0 0 m ̈r =  0  + R B A  0 

mg u1 Using linearized equation of motions near hover assuming small angle assumptions we get,

mx ̈ = (θcψ + φsψ)u1 my ̈ = (θsψ − φcψ)u1 mz ̈ = −mg + u1

It shows the second derivative of position is proportional to input u1.

(23)

(24)

(25)

(26)

(27)

0 p 0 q (28)

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

p = φ ̇cθ − ψ ̇cφsθ q = θ ̇ + ψ ̇sφ r = φ ̇sθ + ψ ̇cφcθ

Substituting the approximation: sin(θ) ≈ θ, sin(φ) ≈ φ, cos(φ) = cos(θ) ≈ 1, we get:

Further substituting: ψ ̇θ ≈ ψ ̇φ ≈ φ ̇θ = 0, we get:

p = φ ̇ − ψ ̇ θ q=θ ̇+ψ ̇φ r = φ ̇θ + ψ ̇

p = φ ̇ q = θ ̇ r = ψ ̇

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

From (28), we get:

00Izzr ̇u2,z q−p000Izzr

Ixxp ̇ = u2,x − Iyyqr + Izzqr Iyyq ̇ = u2,y + Ixxpr − Izzpr (29) Ixxr ̇ = u2,z − Ixxpq + Izzpq

Ixx 0 0 q ̇ u2,x  0 r −qIxx 0  0 Iyy 0 q ̇=u2,y−−r 0 p  0 Iyy

Using approximation, ψ ̇θ ̇ ≈ ψ ̇φ ̇ ≈ φ ̇θ ̇ = pq = pr = qr = 0, Ixxp ̇ = u2,x

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

Differentiating (24), Differentiating again,

φ ̈ = u2,x Ixx

θ ̈= u2,y Iyy

ψ ̈ = u2,z Ixx

(31)

Iyyq ̇ = u2,y (30) Ixxr ̇ = u2,z

... ̇ ̇ ̇ ̇  x = θcψ+φsψ) u ̇1 + θcψ−ψψ+φsψ+φcψψ u1 (32)

.... ̇ ̇ ̇ ̇ ̈ ̇ ̇ ̈ ̇2 ̈ ̇ ̇ ̇ ̈ ̇ ̇2 x = θcψ+φsψ) u ̈ +2 θcψ−ψψ+φsψ+φcψψ u ̇ + θcψ−θsψψ−θsψψ−θsψψ +φsψ+φcψψ+φcψψ−φcψψ u

111

(33) After substituting x into (30), we can prove that the fourth derivative of position is proportional to u1. 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,

T x∗(t) = argmin L(x ̇,xt)dt (34)

x(t) 0 solving functional L using Euler-Lagrange equation gives use the optimal function.

ddt ∂L − ∂L = 0 (35) ∂ x ̇ ∂ x

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

T x ∗ (t) = arg min ((xn)2)dt (36)

x(t) 0

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. If we are given a set of n + 1 waypoints w0 , ...wn , the minimum snap trajectory is a piecewise polynomial composed of n 7th order polynomials. Each polynomial piece pi travels between a pair of waypoints wi−1 and wi and takes a known amount of time Ti to complete i = 1, ..., n.

Let S0 = 0 and for i = 1,...,n,Si = ik=1 Tk. Si is the time it takes to reach waypoint wi from waypoint w0. Then the polynomial pi has the form:

t − S−i1 t − S−i1 2 t − S−i1 7 pi(t)=αi0 +αi1 T +αi2 T +...+αi7 T (37)

iii To obtain the complete equation for piecewise trajectory, we need to solve for all the coefficient ≈ij. There are 8n such

coefficients. These coefficient must satisfy a series of constraint. First the polynomial must go thought all the way points pi(Si−1) = wi−1 and pi(Si) = wi for all i = 1,...,n (2n constraints) (38)

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

p(k)(S ) = p(k) = 0 and for al k = 1,.., 3 (6 constraints) (39) 10n

Third, Velocity, acceleration, 3rd to 6th derivatives are continuous p(k)(S ) = p(k) and for al k = 1,.., 6 (6n-6 constraints) (40)

1 i i+1 The coefficient can be solved by converting the equations in matrix form

The coefficients can be obtained by solving

Aα = b (41) α = A−1b (42)

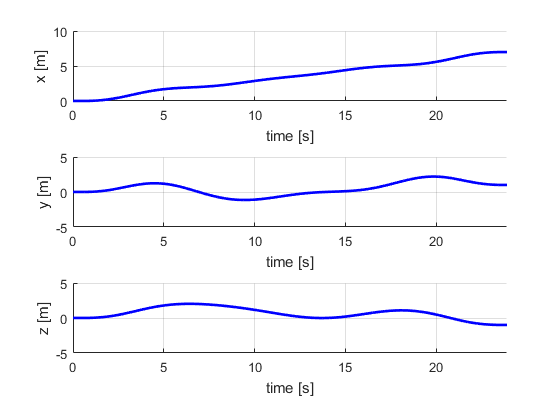
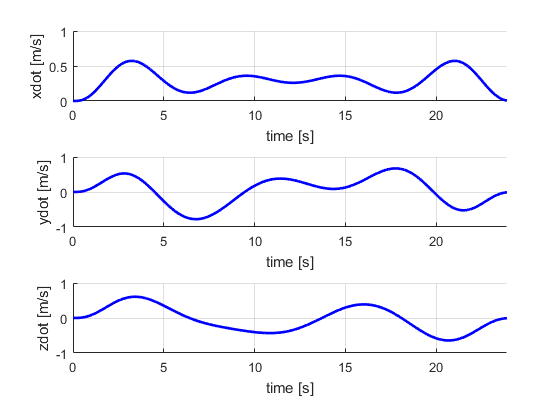
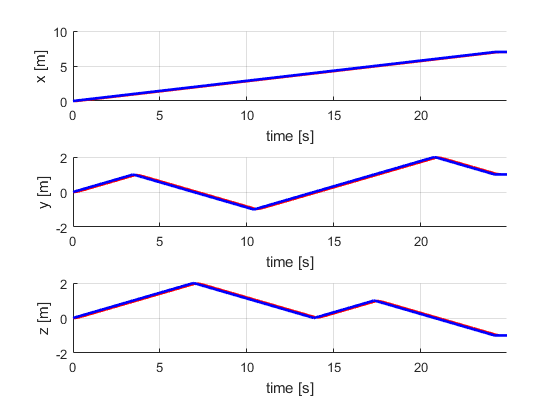
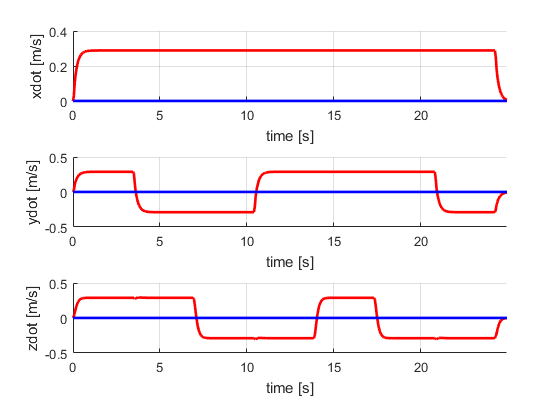
VIII. SIMULATION RESULTS

In this section, results obtained from minimum snap trajectory generation are 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 Ixx = 0.00025kgm2, Iyy = 0.000232kgm2, Izz = 0.0003738kgm2

(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 6, that the trajectories are smooth and their no discontinuity in the system. The smooth trajectory 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 applications 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 concepts 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.MellingerandV.Kumar,“Minimumsnaptrajectorygenerationandcontrolforquadrotors,”in*RoboticsandAutomation(ICRA),2011IEEEInternational 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.