Modelling and Linear Control of Quadcopter in One Dimension

Mohammad Manzur Murshid
Dept. of Electrical and Computer Engineering
University of Hartford,
Hartford, CT, USA
murshid@hartford.edu

Dr. Patricia Mellodge*(Advisor)*
Dept. of Electrical and Computer Engineering (ECE)
University of Hartford
Hartford, USA
mellodge@hartford.edu

***Abstract*—This report paper focuses and explains in detail step by step process and implementation of how Quadcopter mathematical model can be used to represent the control of a system. Mathematical model has been developed and performed numerous control system analysis tool to understand the system response and develop a stable system.Matlab Software has been used to validate the model based on theoretical approach and proposed in this paper.**

Keywords— Quadcopter, Transfer function, State –space equation, PD controller, Root-locus, Nyquist plot, system response.

I. INTRODUCTION

Quadcopters has been dominating these days in a variety of application and industries [1]. And as this is a UAV (Unmanned Aerial Vehicle) the use of quadcopters in different location in air is more cost effective than before. Quadcopters has the ability of Vertical Take-off and Landing (VTOL)[2] which makes this little aircraft most attractive and usability in many impossible task like working in rough terrain and in military or govt. security operation. And because the development cost of gyro and different sensors in integrated circuits are getting lower, the production cost of these Quadcopters is getting less [3].

Generally Quadcopters has four motors with propellers which drive the [4] quadcopter in different direction. To lift/move the quad the rotors spins and creates a thrust which makes the quad a dynamically unstable system. In order to achieve the stability the use of sensors like Gyro and Accelerometer comes to play the main role. These two sensors combined and formed a 6-DOF (six degree of freedom) feedback system in 3-Dimension. This is used with the four control parameter throttle/elevator, roll, pitch, yaw to control the quadcopter. These control parameters comes from the translation of four rotor thrust control variables and control angle between them.

In order to control these variables for a stable flight/system there are number of control methods available [5] including PID controller. In this paper we will only focus on one-dimension control of quadcopter and designing the system around it. For simplicity we will be using only throttle control to perform the elevation or altitude.

Our main purpose will be to design the one-dimensional quadcopter mathematical model and perform the control system analysis and using of different tools that we have learned in the control system course, and make a stable system based on the knowledge we gathered.

II. MATHEMATICAL MODELLING OF QUADCOPTER

To design a mathematical model we will follow several steps as the quadcopter is itself a very under actuated and non-linear system [1][2]. As our goal is to control one-dimensional system so we will be using linear version of the dynamics equation and design around these foundation.

A. Design of Quadcopter

As we mentioned before quad-rotor has four rotors that has to support the vehicle weight, so each rotor spins and generate thrust. Figure 1 shows a typical quadcopter design and dynamics according to force and moment.

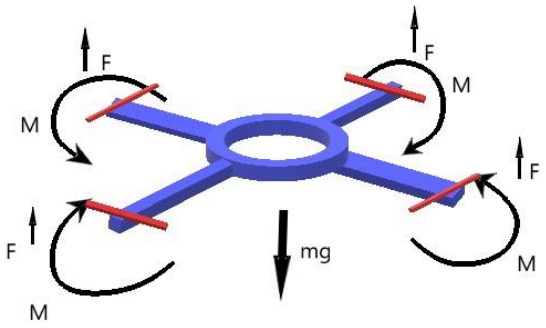


Figure 1: Quadcopter Structure and reference of Force and Moment.

We know the Force, $\boldsymbol{F = k_f \omega^2}$

Where, $\boldsymbol{k} = thrust\ Constant,$

And $\boldsymbol{\omega} = angular\ speed\ or\ velocity(r.p.m).$

If we plot, thrust vs. the angular velocity /r.p.m then we will see that the relationship is a proportional of Thrust and square of angular speed, which is almost quadratic. And every time motor spins there is also drag moment, M generates which the rotor has to overcome. This is also quadratic equation, $\boldsymbol{M = k_M \omega^2}$. The figure below shows the relationship between forces, moment vs. angular speed of a rotor.

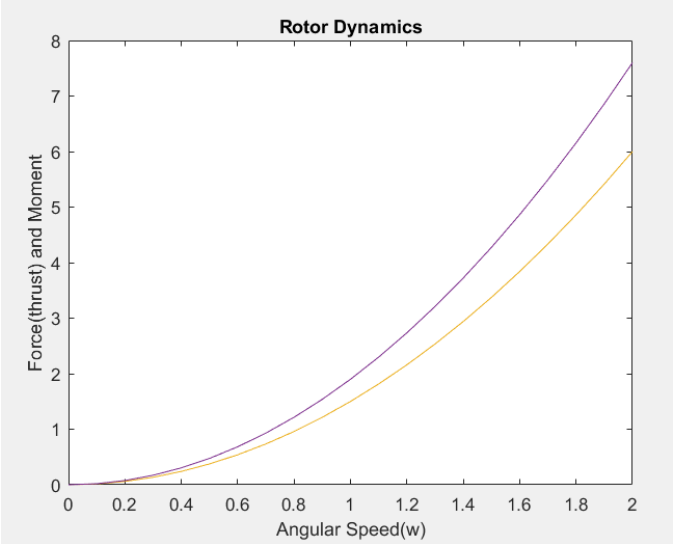


Figure 2: Rotor dynamics (Force/thrust and Moment) vs. Angular speed, ω (r.p.m.))

So if we think about quadcopter, then every rotor has to support roughly 1/4th of its total weight in equilibrium point. From the above curve we can determine the speed that will be required to produce the 1/4th of weight or thrust, which is $operating\ speed, \boldsymbol{\omega_0}$. This operating speed will reduce the drag moment, M that every rotor has to overcome. And that's where the Motor comes in, where we select a size which can produce enough torque, $\boldsymbol{\tau}$. This torque, $\boldsymbol{\tau}$ can overcome the drag moment.

Where, Motor torque relation is:

$$\boldsymbol{\tau = k_m \omega^2.}$$

Now considering the hovering condition, the rotor speed compensate for the weight, and using the weight we can determine the basic operating speed for hovering that in turn shows us what torque, $\boldsymbol{\tau}$ we need to feed to each of the motor. So we can write the motor speed as,

$$\boldsymbol{k_f \omega^2 = \frac{1}{4} mg}$$

So now using this equation we can calculate the sum of the resultant force, which is the sum of four rotor thrust and the gravitational force that is pulling down.

$$\boldsymbol{\sum F = F1 + F2 + F3 + F4 - mg}$$

Where, $\boldsymbol{F1, F2, F3, F4 = each\ motor\ thrust}$

$$\boldsymbol{m = weight\ of\ Quadcopter, Kg}$$

$$\boldsymbol{g = gravitational\ Acceleration, \frac{m}{s^2}}$$

Again as we know the center of mass for the quadcopter we can calculate the resultant moment executed by each rotor which is:

$$\sum M = r1 * F1 + r2 * F2 + r3 * F3 + r4 * F4 + M1 + M2 + M3 + M4\ ,$$

Where total moment, is due to forces applied by rotors and the reaction of the rotor spinning in clockwise or counter-clockwise direction. These reactions are moment and they add to the net moment.

In equilibrium point or hovering position this resultant force and moment is zero

$$\sum F = 0$$

$$\sum M = 0$$

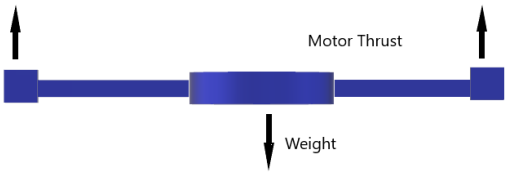


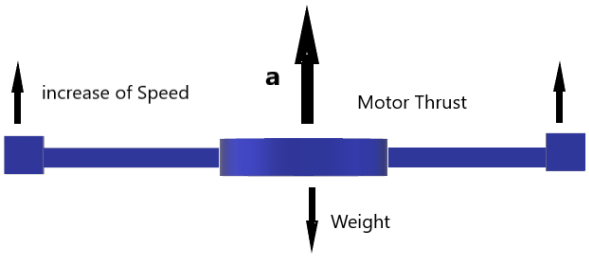
Figure 3: Quad-rotor characteristic in Equilibrium

So now if these force and moment is non-zero then we will get acceleration.

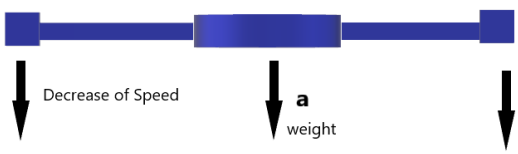
To keep calculation simple and our aim of this paper is to present only one direction we will use vertical acceleration. Where in the hovering point the acceleration is zero

$$\sum \boldsymbol{k_f \omega^2} + \boldsymbol{mg} = 0$$

So, every motor thrust is the same and they all add up to support the total weight of the quad, but when we increase the motor thrust or speed then the quad will accelerate up and if we decrease the speed it will go down.



(a)



(b)

Figure 4: (a) and (b) Quad-rotor characteristics in acceleration

B. Plant Model

We already know in equilibrium point

Force and moment is zero

$$\sum F = 0$$

$$\sum M = 0$$

So as, acceleration $\sum \boldsymbol{k_f \omega^2} + \boldsymbol{mg} = 0$

But, when we have acceleration,

We have, $\sum \boldsymbol{k_f \omega^2} + \boldsymbol{mg} = m\boldsymbol{a_{\hat{z}}}$

Let us use $\boldsymbol{x}$ *as a height or vertical displacement*

Then we can mention $\boldsymbol{a}$ as second order derivative for displacement,

$$\boldsymbol{a} = \frac{\boldsymbol{d^2x}}{\boldsymbol{dt^2}} = \ddot{\boldsymbol{x}}$$

Now we write

$$\sum \boldsymbol{k_f \omega^2} + \boldsymbol{mg} = m.\ddot{\boldsymbol{x}}$$

If we take the left side which is sum of all the forces divided by mass then we can write,

$$\boldsymbol{u} = \frac{1}{4}\left(\sum \boldsymbol{k_f \omega^2} + \boldsymbol{mg}\right)$$

Where $u = control\ input\ or\ plant\ input\ model$

Now we have input $\boldsymbol{u}$ *and variable* $\boldsymbol{x}$

So we can write, $\boldsymbol{u} = \ddot{\boldsymbol{x}}$ **(1)**

C. Controller Design of a linear system

Until now we have,

State and input which is $\boldsymbol{x}, \boldsymbol{u} \in \mathfrak{R}$

And our plant model from (1) is

$$\boldsymbol{u} = \ddot{\boldsymbol{x}}$$

Our goal is to find the control input function, $u(t)$ for the plant model so that function of $x(t)$ follows the trajectory of $x_{des}(t)$,

Where $\boldsymbol{x(t)} = \boldsymbol{position}$ and

$$\boldsymbol{x_{des}(t)} = \boldsymbol{desired\ position}$$

So the State-Space model we can develop is:

$$\dot{x_{des}} = \begin{bmatrix} 0 & 0 \\ kf/m & 0 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} \omega^2 \\ 0 \end{bmatrix} U$$

$$y = [0 \;\; x(t)] \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} U$$

The below figure show the linear model of the quadcopter control system without PD controller.

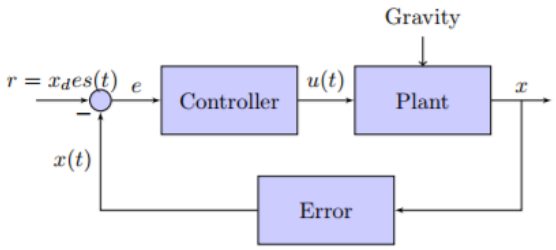


Figure 4: Linear Model Schematic

In order to calculate the desired position we also have to calculate the error from actual position.

As a general approach we can define,

Error function, $\boldsymbol{e(t) = \ddot{x}_{des}(t) - \ddot{x}(t)}$

But we want this error function to become zero and specifically we want this function to merge exponentially to zero. Or we can say we want to find $u(t)$ so that in equilibrium point error, $e(t) = 0$

Or, $\ddot{e} + K_v\dot{e}(t) + K_p e(t) = 0 \qquad K_v, K_p > 0$

Where we have two unknown K_p and K_v as a proportional and derivative gain respectively. And we can write,

$$\boldsymbol{u(t) = \ddot{x}_{des}(t) + K_v\dot{e}(t) + K_p e(t)} \qquad \textbf{(2)}$$

So we have now introduced the **PD controller** in equation (2) which will reduce the error function of $\boldsymbol{u(t)}$ and shown in below figure.

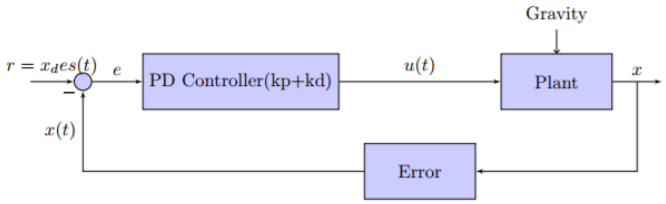
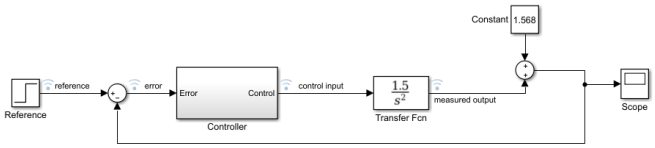


Figure 5: Linear control with PD control

III. Control System Analysis

A. *Plant Design based on real data*



a.

Figure 6: Block diagram of drone system

Considering above equations and the block diagram we can develop the transfer function as below:

$$\boldsymbol{k_f\omega^2 + mg = m.\ddot{x}} \qquad \textbf{(3)}$$

$$\textbf{So } \boldsymbol{U(s)k_f = ms^2}\textbf{X(s)} \qquad \textbf{(4)}$$

Where we will consider the input from the controller which will be $\boldsymbol{u = \omega^2}$ as we wanted to control the thrust for desired height $x(s)$. We will also consider the gravitational force which is pulling down as a disturbance in the system.

So $from\ equation\ (4) \quad G(s) = \frac{X(s)}{U(s)} = \boldsymbol{k_f/ms^2}$

Where, $\boldsymbol{k_f = 1.5}$ $\boldsymbol{(Thrust\ Constant)}$

$\boldsymbol{m = 1.6\ kg}$

Considering the 1/4th weight $\boldsymbol{m = \frac{1.6}{4} = 0.4\ kg}$

We get the final transfer function as,

$$G(s) = \frac{X(s)}{U(s)} = \frac{k_f}{ms^2} = \frac{1.5}{0.4s^2}$$

```
g =

    1.5
  -------
  0.4 s^2

Continuous-time transfer function.
```

And the State-Space equation becomes:

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} U$$

$$y = \begin{bmatrix} 0 & 1.875 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} U$$

Using the function in MATLAB we achieve the following step response

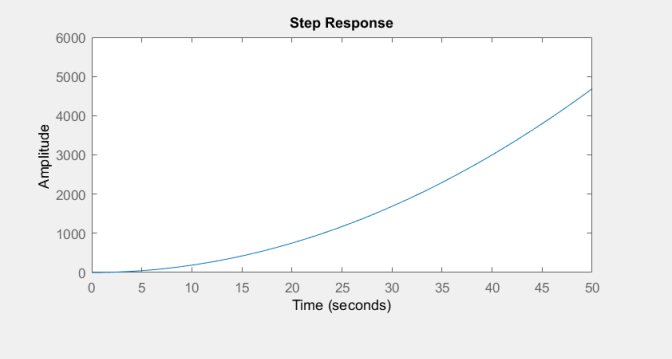


Figure 7: step response of open loop Transfer function

We found the system as a type-2 system and after using the closed loop function with PD controller and adding the disturbance term in the system we find this result.

$$t = \frac{0.6272\ s\text{^}2 + 3.852\ s + 3.852}{0.4\ s\text{^}2 + 1.5\ s + 1.5}$$

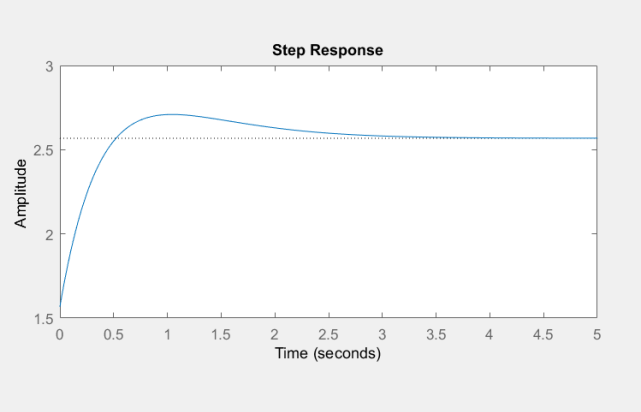


Figure 8: Closed loop transfer function step response

Where in the primary step we used Kp and Kd value as 1.

Using the MATLAB "stepinfo" command we find following results

RiseTime Tr: 0.3828
SettlingTime, Ts: 2.7598
Overshoot, $\%OS$: 14.1207
Peak, p: 1.1412
PeakTime, Tp: 1.0561

We performed the root locus and Nyquist analysis which shows the system is in stable condition which is shown in the below figure:

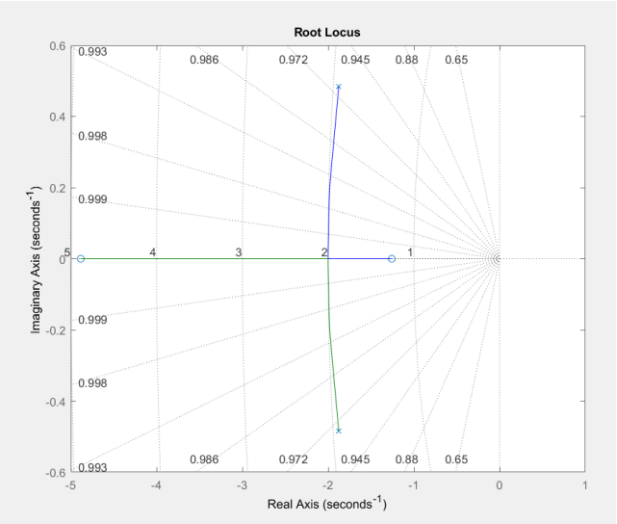


Figure 9: Root-locus plot

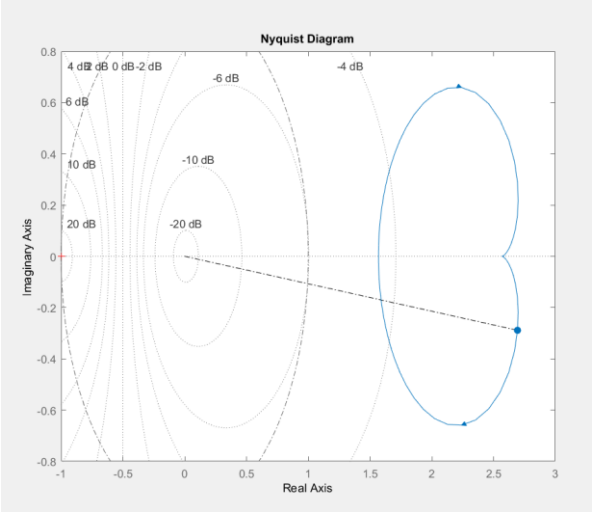


Figure 10: Nyquist Plot

B. *PD Controller Design*

For this project we initially proposed that we will be using the PD Controller. In this section we will try to get the parameters for the PD controller output.

We know for PD controller

$$\boldsymbol{D(s) = \hat{k}(s + Zc)}$$

Where, $\boldsymbol{D(s) = controller}$

$\boldsymbol{and\ \ Zc}$ *is the Zero near the origin in the root locus plot*

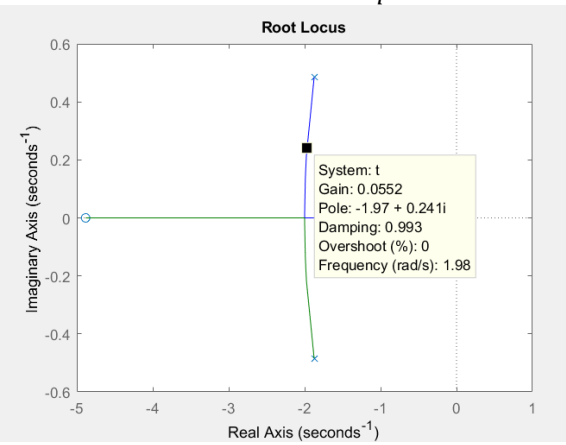


Figure 11: Root-Locus plot

From the root-locus plot we can observe that at 0 % os the uncompensated gain k = 0.0552 and the pole location is

$$P = -1.97 + 0.24j$$

This system has already is in stable condition.

IV. Results and Discussion

For this system the result that we have achieved is shows that out primary system response for the transfer fucntion having parabolic response and using the PD controller we achieved the response we are looking for.As we also noticed that our system is a non linear system and we ignored the non-linear term using disturbance, our system is acting as a linear system.

V. Conclusion

In this paper we tried to model a Quadcopter using mathematical equations and simulation result where our goal was to demonstrate the control system analysis to achieve the necessary control system tool implementation in the model and getting the result we are expecting. In future work we will achieve more dimensions by using the non-linear terms and linearizing the system.

VI. References

[1] Kyaw Myat Thua*, A.I. Gavrilov- Designing and modeling of quadcopter control system using L1 adaptive control- XIIth International Symposium «Intelligent Systems», INTELS'16, 5-7 October 2016, Moscow,Russia

[2] Mohamed A.Abdelkhalek, Mohamed S.El-Demerdash, Ahmed A.El-Tahan, Tarek N.Dief," Attitude Stability of Quadcopter Using Classic Control with Angular Acceleration". http://www.ijcsits.org/papers/vol5no42015/4vol5no4.pdf

[3] Low Cost and Flexible UAV Deployment of Sensors, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5298727/

[4] Zoran Benić1, *, Petar Piljek2 and Denis Kotarski3, "MATHEMATICAL MODELLING OF UNMANNED AERIAL VEHICLES WITH FOUR ROTORS". 3DOI: 10.7906/indecs.14.1.9.

[5] Teppo Luukkonen, "Modelling and control of quadcopter" School of Science, Mat-2.4108, Independent research project in applied mathematics, Espoo, August 22, 2011, Aalto University, School of Science

VII. Appendix

```
%Thrust or Force VS Angular Speed curve
w = 0:0.1: 2;
f = 1.5*w.^2;
title('Rotor Physics')
xlabel('Angular Speed(w)')
ylabel('Force(thrust) and Moment')
plot(w, f)
hold on

%Moment or drag VS Angular Speed curve
w = 0:0.1: 2;
m = 1.9*w.^2;
title('Rotor Dynamics')
xlabel('Angular Speed(w)')
ylabel('Force(thrust) and Moment')
plot(w, m)

[2] to design the Flow chart for Control system
https://www.overleaf.com/12861018yhhwtdjtpzjn#/49129850/


Code:
num = [0, 0, 1.5]
den = [0.4, 0, 0]
g = tf(num, den)
subplot(2, 2, 1)
step(g)
kp = 1
ki = 0
kd = 1

c = pid(kp, ki, kd)

error = 1

t = feedback(g*c, error)+1.568
subplot(2, 2, 2)
step(t)
```