5 DoF Drone Controller

Kaimeng Du
Dept of Engineering
Univerity of British Colunmbia
Kelowna, Canda
shadowdu@student.ubc.ca

***Abstract*—In this project, a drone system was built, linearized; a controller was developed and tested in simulation.**

***Keywords*—drone, controller, nonlinear, observer**

I. Introduction

I bought a drone development board last year, pydrone, which is equipped with an ESP32-S3 microcontroller, a barometer, a compass, an accelerometer and a gyroscope, Figure 1. I was planning to develop a controller firmware, yet found it was too complicated for me. This is one of the reasons why I selected this course.

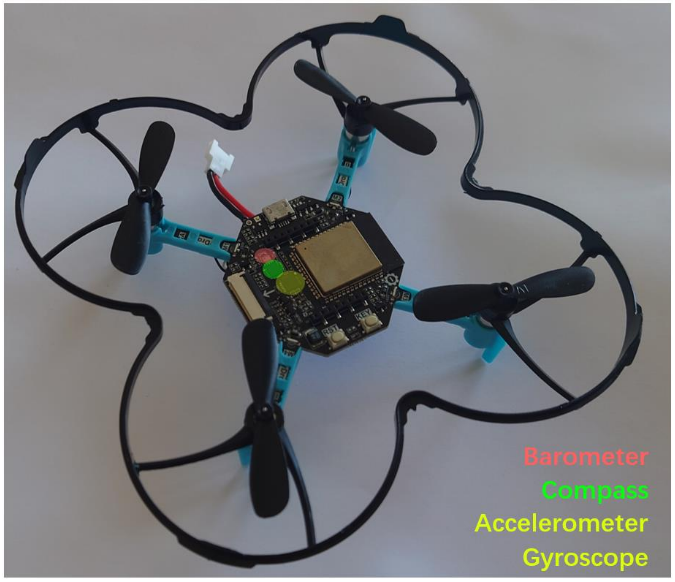


Fig. 1 Pydrone

II. Project Goals

The controller property will be examined by altering the parameters. The performance of the controller will also be tested under noise. The stability of the drone in wind will be simulated and tested. Finally, the possibility of constructing a feedback controller only using the measurable variables, e.g., readings from barometer/compass/accelerometer/gyroscope will be tested.

To design a feedback controller of quadrotor drone. To simplify the model, the rotor and propeller properties will not be considered. Pulling force generated from propellers will be considered input. Moreover, wind effect will not be taken into consideration. Finally, only headless mode will be implemented in this model, where no yaw will be conducted by this drone.

III. Model Definition

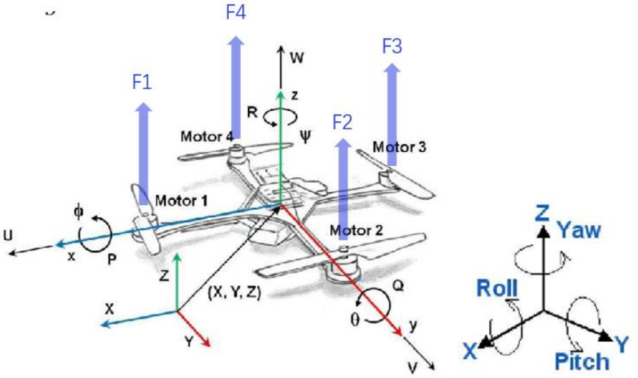


Fig. 2 The definition of axis and variables. Image from Douglas et. al. 2016.

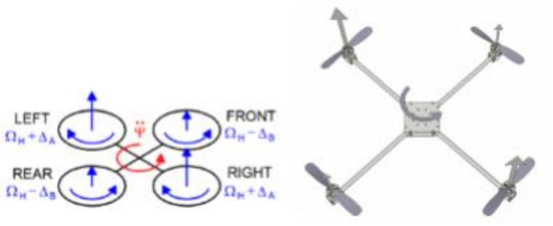
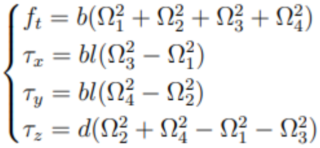


Fig. 3 Explanation how yaw generated in drone. Figure from Sabatino, 2015

Sabatino described the dynamics of the actuators as Equation 1 [1], where f_t is the total force generated; Ω_i is the velocity of the propeller i; b is the ratio of pulling force generated based on angular velocity of propellers; d is the drag coefficient of propellers.



$$\begin{cases} f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_x = bl(\Omega_3^2 - \Omega_1^2) \\ \tau_y = bl(\Omega_4^2 - \Omega_2^2) \\ \tau_z = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{cases}$$

Equ. 1 Actuator dynamics from 4 motors. Sabatino, 2015

To simplify this model, I select to use Headless mode of drone, where the direction of the drone is fixed. That is, no rotation along Z axis, which can be express as:

$$\psi = 0, \forall t$$
$$\tau_z = 0$$
$$\psi = 0, \dot{\psi} = 0, \ddot{\psi} = 0$$

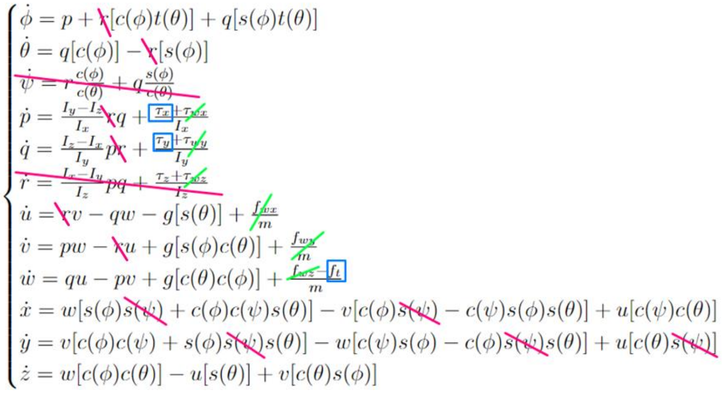
So the force in Equation (1) can be simplified as:

$$f_t = F1 + F2 + F3 + F4$$
$$\tau_x = l(F2 - F4)$$
$$\tau_y = l(F3 - F1)$$
$$F2 + F4 = F1 + F3$$

Equ. 2 Simplified Actuator dynamics from 4 motors given no Z axis rotation.

So the system dynamics can be simplified as follows, in Equation 3:

1. where **pink** line crossed variables are simplified because of $\psi = 0, r \doteq \psi = 0, \ddot{\psi} = 0$.
2. where **green** line crossed variables are simplified because wind factors are not considered.
3. **blue** box indicates the input to the system.



$$\begin{cases} \dot{\phi} = p + \cancel{r}[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\ \dot{\theta} = q[c(\phi)] - \cancel{r}[s(\phi)] \\ \cancel{\dot{\psi} = r\frac{c(\phi)}{c(\theta)} + q\frac{s(\phi)}{c(\theta)}} \\ \dot{p} = \frac{I_y - I_z}{I_x}\cancel{r}q + \frac{\boxed{\tau_x} + \cancel{\tau_{wx}}}{I_x} \\ \dot{q} = \frac{I_z - I_x}{I_y}p\cancel{r} + \frac{\boxed{\tau_y} + \cancel{\tau_{wy}}}{I_y} \\ \cancel{\dot{r} = \frac{I_x - I_y}{I_z}pq + \frac{\tau_z + \cancel{\tau_{wz}}}{I_z}} \\ \dot{u} = \cancel{r}v - qw - g[s(\theta)] + \cancel{\frac{f_{wx}}{m}} \\ \dot{v} = pw - \cancel{r}u + g[s(\phi)c(\theta)] + \cancel{\frac{f_{wy}}{m}} \\ \dot{w} = qu - pv + g[c(\theta)c(\phi)] + \frac{\cancel{f_{wz}} - \boxed{f_t}}{m} \\ \dot{x} = w[s(\phi)\cancel{s(\psi)} + c(\phi)c(\psi)s(\theta)] - v[c(\phi)\cancel{s(\psi)} - c(\psi)s(\phi)s(\theta)] + u[c(\psi)c(\theta)] \\ \dot{y} = v[c(\phi)c(\psi) + s(\phi)\cancel{s(\psi)}s(\theta)] - w[c(\psi)s(\phi) - c(\phi)\cancel{s(\psi)}s(\theta)] + u[c(\theta)\cancel{s(\psi)}] \\ \dot{z} = w[c(\phi)c(\theta)] - u[s(\theta)] + v[c(\theta)s(\phi)] \end{cases}$$

Equ. 3 The quadrotor system dynamics equation defined based on state variable that $\boldsymbol{X} = [\psi \ \theta \ \varphi \ p \ q \ r \ u \ v \ w \ x \ y \ z]$. Sabatino,2015. c for cos(); s for sin(); t for tan().

To summarize, the simplified the overall dynamics of the headless drone system with no wind effect can be concluded as follows:

$$\begin{cases} \dot{\varphi} = p + r[c(\varphi)t(\theta)] + q[s(\varphi)t(\theta)] \cdots\cdots\cdots\cdots\cdots & (4.1) \\ \dot{\theta} = q[c(\varphi)] - r[s(\varphi)] \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & (4.2) \\ \dot{p} = \frac{\tau_x}{I_x} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & (4.3) \\ \dot{q} = \frac{\tau_y}{I_y} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & (4.4) \\ \ddot{x} = -\dot{\theta} \cdot \dot{z} - g \cdot s(\theta) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & (4.5) \\ \ddot{y} = \dot{\varphi} \cdot \dot{z} + g \cdot c(\theta)s(\varphi) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & (4.6) \\ \ddot{z} = \dot{\theta} \cdot \dot{x} - \dot{\varphi} \cdot \dot{y} + g \cdot c(\theta)c(\varphi) - \frac{f_t}{m} \cdots\cdots\cdots\cdots & (4.7) \end{cases}$$

Equ. 4 The headless quadrotor dynamics summarized with no wind effect

Define the states variables as $\boldsymbol{X} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ \varphi \ \theta \ p \ q]$, the input is defined as $\boldsymbol{u} = [f_t \ \tau_x \ \tau_y]$. Then the nonlinear state space system can be described as Equation 5:

$$\dot{\boldsymbol{X}} = f(\boldsymbol{X}) + g_1(\boldsymbol{X})f_t + g_2(\boldsymbol{X})\tau_x + g_3(\boldsymbol{X})\tau_y \qquad (5.1)$$

$$f(\boldsymbol{X}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ -\dot{\theta} \cdot \dot{z} - g \cdot s(\theta) \\ \dot{\varphi} \cdot \dot{z} + g \cdot c(\theta)s(\varphi) \\ \dot{\theta} \cdot \dot{x} - \dot{\varphi} \cdot \dot{y} + g \cdot c(\theta)c(\varphi) \\ p + r[c(\varphi)t(\theta)] + q[s(\varphi)t(\theta)] \\ q[c(\varphi)] - r[s(\varphi)] \\ 0 \\ 0 \end{bmatrix} \qquad (5.2)$$

$$g_1(\boldsymbol{X}) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad -\tfrac{1}{m} \quad 0 \quad 0 \quad 0 \quad 0] \qquad (5.3)$$

$$g_2(\boldsymbol{X}) = \left[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \tfrac{1}{I_x} \quad 0\right] \qquad (5.4)$$

$$g_3(\boldsymbol{X}) = \left[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \tfrac{1}{I_y}\right] \qquad (5.5)$$

Equ. 5 The nonlinear system representation of the drone system

IV. Linearization

An equilibrium point is found by set $\dot{\boldsymbol{X}} = \boldsymbol{0}$:

From Eq 4.6, $s(\varphi) = 0 \rightarrow \ \varphi = 0$

From Eq 4.5, $c(\varphi)s(\theta) = 0 \rightarrow \ \theta = 0$

From Eq 4.7, $\ddot{z} = g - \frac{f_t}{m} = 0 \rightarrow g = \frac{f_t}{m} \rightarrow \ f_t = mg$

From Eq 4.3, $\dot{p} = \frac{\tau_x}{I_x} = 0 \rightarrow \ \tau_x = 0$

From Eq 4.4, $\dot{q} = \frac{\tau_y}{I_y} = 0 \rightarrow \ \tau_y = 0$

Therefore, the equilibrium point is $\boldsymbol{X} = [\tilde{x} \ \tilde{y} \ \tilde{z} \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$, where $\tilde{x}, \tilde{y}, \tilde{z} \in R$, and $\boldsymbol{u} = [mg \ \ 0 \ \ 0]$

The linearization can be calculated as follows:

$$\boldsymbol{A} = \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\dot{\theta} & 0 & -g & 0 & -\dot{z} \\ 0 & 0 & 0 & 0 & 0 & \dot{\varphi} & g & 0 & \dot{z} & 0 \\ 0 & 0 & 0 & 0 & \dot{\theta} & -\dot{\varphi} & 0 & 0 & -\dot{y} & \dot{x} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

At the equilibrium point, with the assumption that when θ is small, $\sin(\theta) \to \theta, \cos(\theta) \to 1$:

$$\boldsymbol{A}|_{\boldsymbol{X=0}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{B} = \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 \\ 0 & 0 & \frac{1}{I_y} \end{bmatrix}$$

V. Output Definition

The output that can be measured form the system is from accelerometer, gyroscope and barometer. The angular velocity $\dot{\varphi}\ \ \dot{\theta}$ can be measured by gyroscope. The z displacement z can be measured by barometer. The velocity in x and y direction $\dot{x}\ \ \dot{y}$ can be measured by integration of accelerometer signals, though noisy.

Therefore:

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and $D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

VI. System Development

A. Input and Output

The system has 3 inputs: $\boldsymbol{u} = [f_t \quad \tau_x \quad \tau_y]$.
The system has 10 states: $\boldsymbol{X} = [x\ \ y\ \ z\ \ \dot{x}\ \ \dot{y}\ \ \dot{z}\ \ \varphi\ \ \theta\ \ p\ \ q]$
The system has 5 outputs: $\boldsymbol{y} = [\dot{\varphi}\ \ \dot{\theta}\ \ z\ \ \dot{x}\ \ \dot{y}\ ]$
(handout #2 3.1)

B. Jordan form and Stability

The Jordan form of the system is: (handout #2 3.2)

$$J = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The eigenvalues are all zero. The Jordan form consists of 2 blocks with zero eigenvalues of size 4 (blue and green), and one block with zero eigenvalues of size 2 (red).

Therefore, the system is not stable. (handout #2 3.3).
The nonlinear system is not stable at equilibrium point, any disturbance can make the drone fall. (handout #2 3.4)
The system is not BIBO stable because of zero poles. (handout #2 3.5)

I forgot to bring the lipo battery with me to Canada and it is extremely difficult to buy proper one. I was not able to do a test on a real system. (handout #2 3.6)

The open loop simulation shows the linear drift of the linear system and an exponential drift in the nonlinear system. (handout #2 3.7)

C. Open loop Homogeneous Response

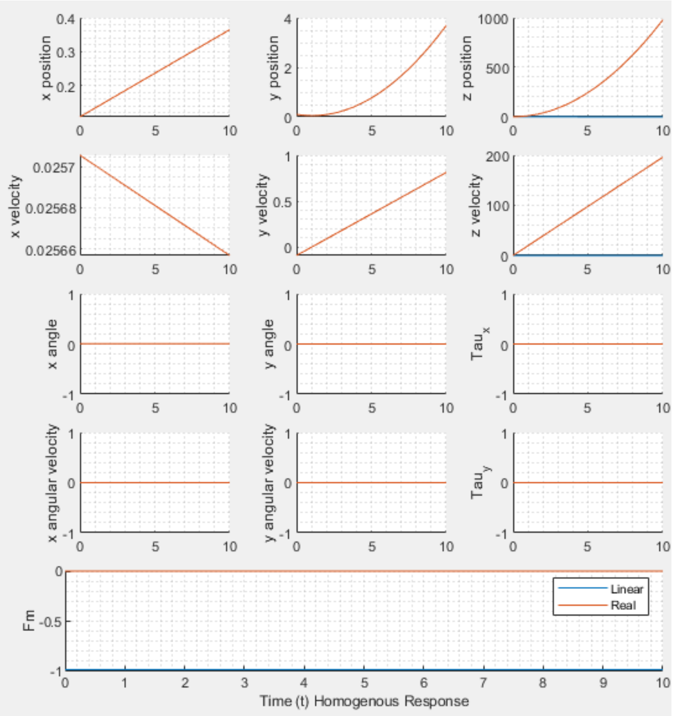


Fig. 4 Open loop simulation of the drone system. Exponential deviation from equilibrium point can be observed on the nonlinear system (orange line).

D. Open loop Step Response

Step in inputs is not interesting since this system is extremely unstable. (handout #2 3.8)

1) Step in Fm:

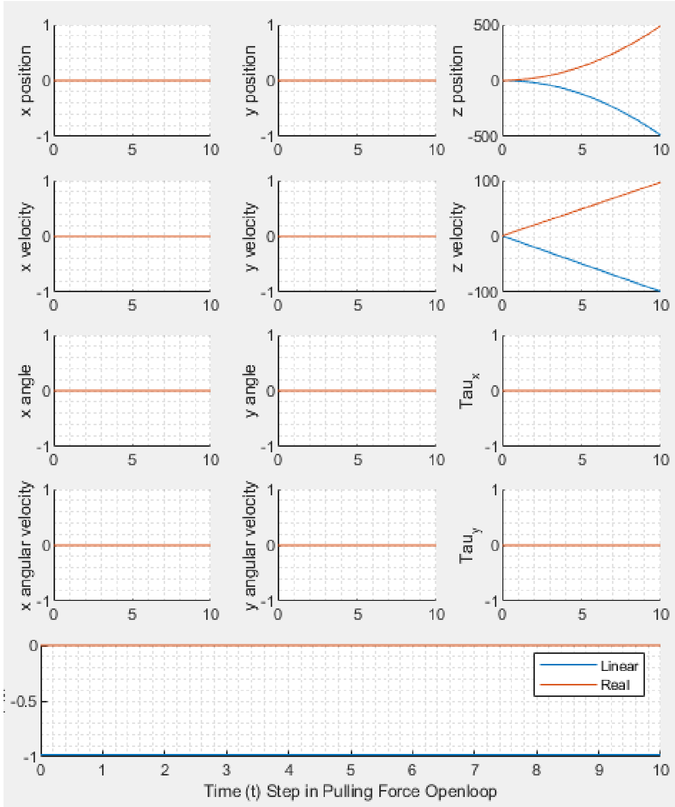


Fig. 5 The step response in total pulling force. The linear and nonlinear system go to opposite directions.

2) Step in τ_x

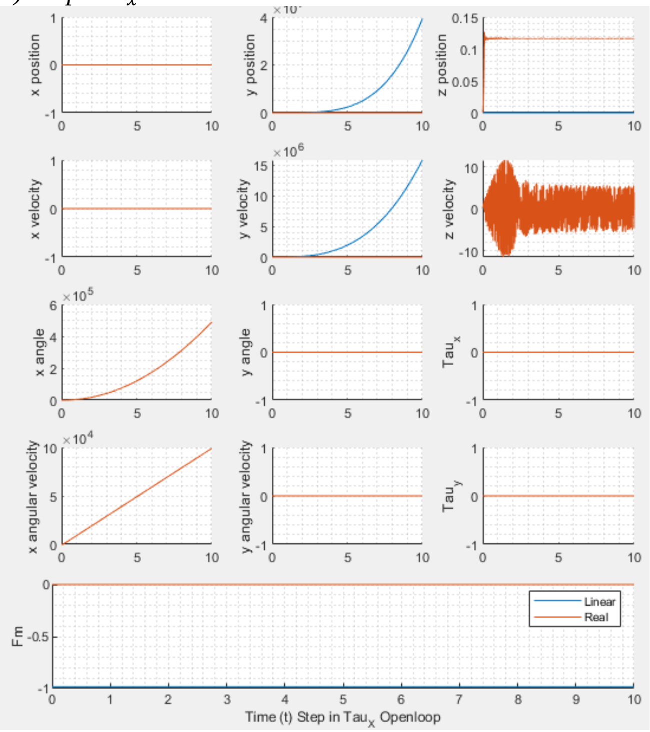


Fig. 6 A step torque input was applied on x direction. A constant increase of angular velocity can be observed in x direction.

3) Step in τ_y

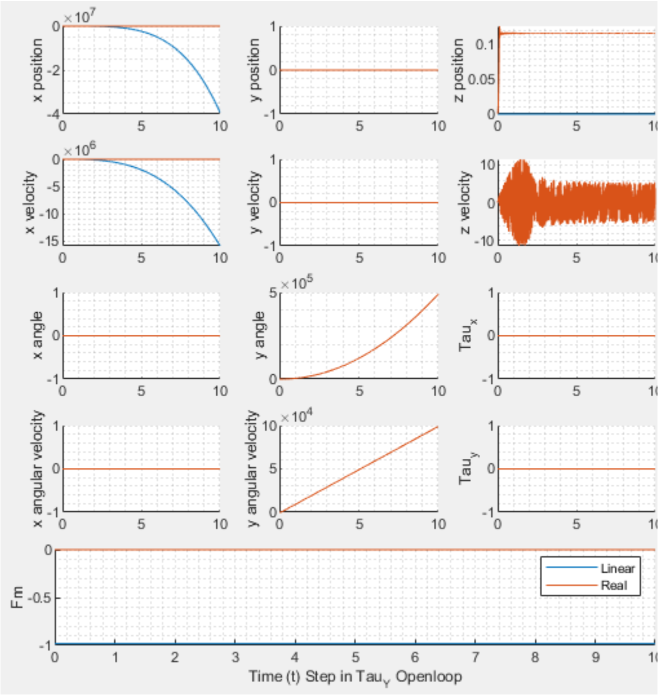


Fig. 7 A step torque input was applied on y direction. A constant increase of angular velocity can be observed in y direction.

Because the response of the nonlinear system is so large, the response of the linear system is neglectable. (handout #2 3.9)

E. *Design Specification*

As a drone, the controller should be able to make drone stay in a relative stable position in the air. It should not drift around. At least, the controller should not make the drone fall on the ground. (handout #2 3.10)

The tentative specification can be defined as follows:

- The drone should be able to take off and land.
- The drone should be able to stay in certain height with limited drift, within 0.1m/s.
- The drone should be able to move up to 3m/s.

F. *Controllability*

The system is controllable. Controllability matrix ranks 10, the same as number of states. (handout #3 2.1)

The input of the system is simplified to the total pulling force and torque generated by the propellers. So, one motor fail will impact all three inputs. If we only consider the impact on torque, the system is already uncontrollable (Controllability matrix ranks 6<10). (handout #3 2.3)

The Kalman decomposition result is:

$$A_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -9.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.4142 & 0 & 0 \end{bmatrix}$$

$$B_k = \begin{bmatrix} 10.0499 & 0.0001 & 0 \\ 0 & 0 & 0^- \\ 0 & 0 & 0 \\ 0^- & 0.0499 & 0^- \\ 0 & -10000 & 0 \\ 0^- & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C_k = \begin{bmatrix} 0^+ & 0^- & 0 & -1 & 0 & 0 & 0 & 0^+ & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0^+ & 1 & 0 \\ 0 & 0 & 0.995 & 0 & 0 & 0 & 0 & 0^+ & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Dimension of $CO = 5, C\bar{O} = 1, \bar{C}O = 3, \overline{CO} = 1$.

The 4 dimensions where the uncontrollable have zero eigenvalues. So, the system is not stabilizable if one motor is broken. (handout #3 2.3)

G. *Controller Design*

1) *LQR Controller*

The controller is design with LQR, where the Q=I and R=0.0001. The response if quick and match the specification defined above (handout #3 2.4). The drone can fly and hold its position in air. The top speed reached around 6m/s.

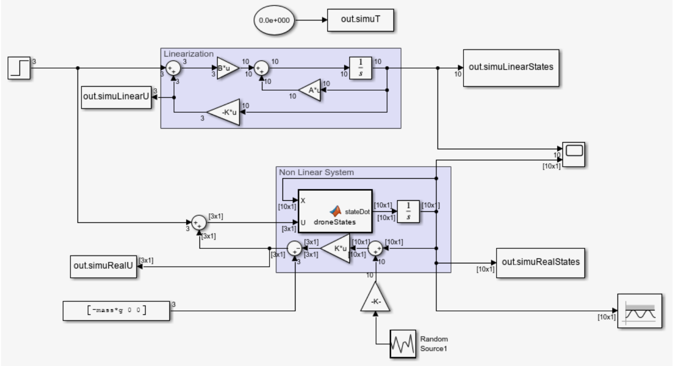


Fig. 8 The nonlinear system and linearization. Both systems use a state feedback controller.

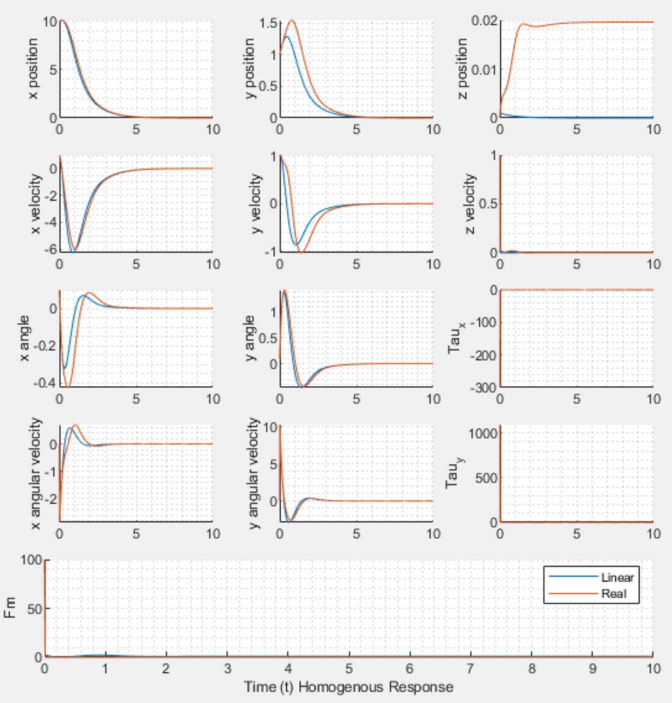


Fig. 9 The LQR controller quickly balance the drone back to equilibrium point and the only drift left is on the z axis because of the gravity.

2) *Eigenvalue Placement*

I was not able to design a proper controller by eigenvalue placement. The best result is as follows:

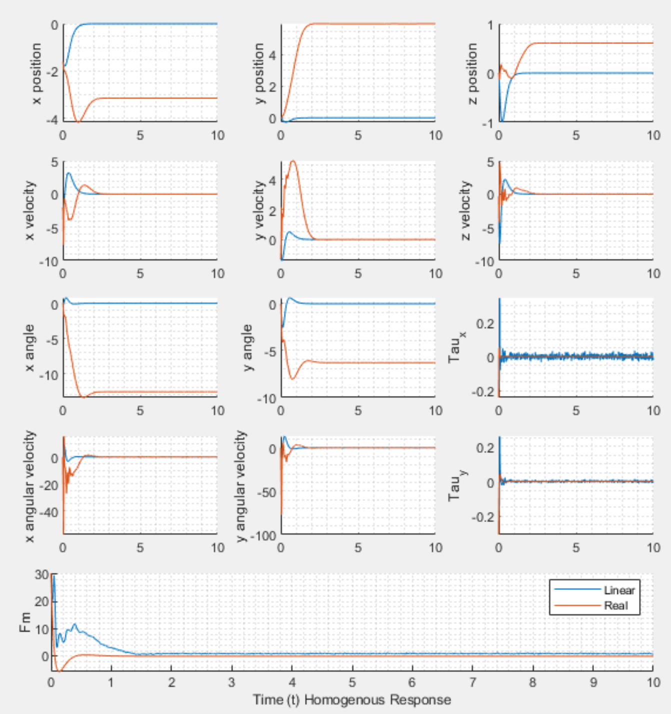


Fig. 10 The eigenvalue placement controller can balance the drone back, but not to the equilibrium point.

The rotation angle in x and y direction, instead of converging back to zero, it stabilized at multiple of 2pi. And a very sharp oscillation can be observed in multiple states. (handout #3 2.5)

H. *Close loop response with nonzero initial condition*

With the LQR controller, the initial condition was quickly compensated, where the initial condition is:

- 0.06m for x direction offset
- -0.01m for y direction offset
- No z offset
- 0.008m/s in x direction
- 0.017m/s in y direction
- 0.04m/s in z direction
- -0.006 rad in x rotation direction
- -0.008 rad in y rotation direction

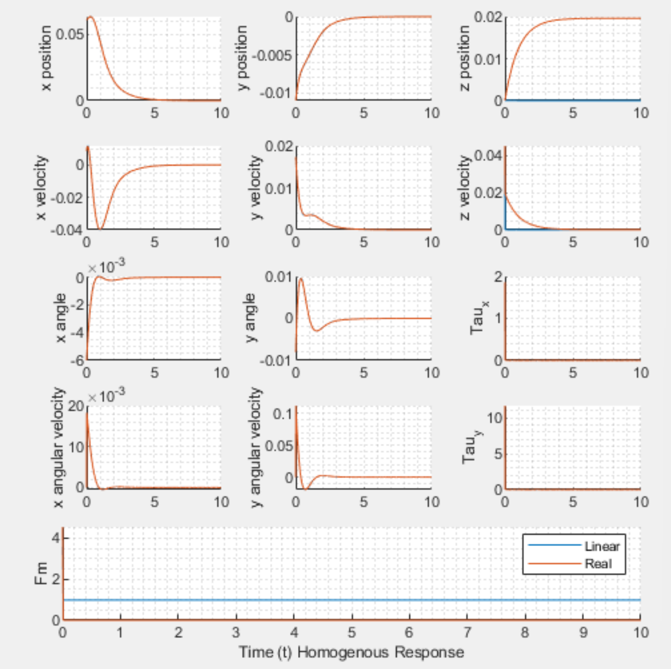


Fig. 11 Close loop response with nonzero initial condition. The linear system is a good approximation of the nonlinear system except the offset in z axis.

Even if the initial condition is amplified for 5 times, it is also quickly compensated. It is a strong controller. (handout #3 2.6)

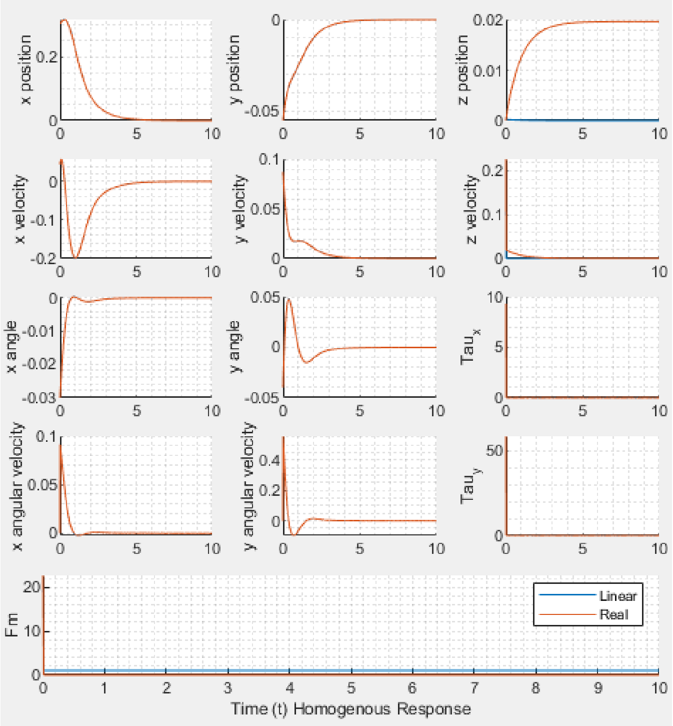


Fig. 12 Close loop response with larger nonzero initial condition. The drone was able to stablize and hold positon within 5 seconds.

I. *Noise in States of the Close loop System*

Noise is added to the states. The noise in the translational directions is expected to be small, since the position of drone can be measured by indoor camera systems. And the rotational noise is also small, since multiple IR reflectors can be used for precise angle measurement.

- The linear noise in position is assumed to be 1cm
- The linear noise in velocity is assumed to be 1cm/s
- The rotational noise in angles is assumed to be 1 degree
- The rotational noise in angular velocity is assumed to be 1 degree/s

1) Noise in States of LQR Controller

With noise in the states, the LQR controller performs as the same. Probably the noise is small. (handout #3 2.7)

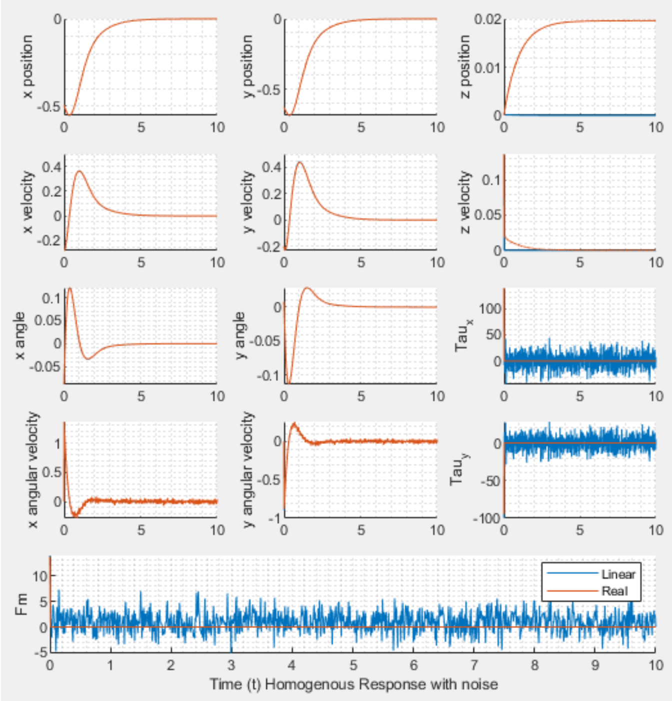


Fig. 13 The homogeneous response of the state feedback nonlinear system with LQR controller

1) Noise in States of Eigenvalue Placement Controller

The eigenvalue placement controller can also handle the noise. It seems that the eigenvalue placement controller is more sensitive to initial conditions than noise. This probably because of the eigenvalue I assigned are smaller than those generated by LQR, so the drone takes more time to stable given nonzero initial conditions.

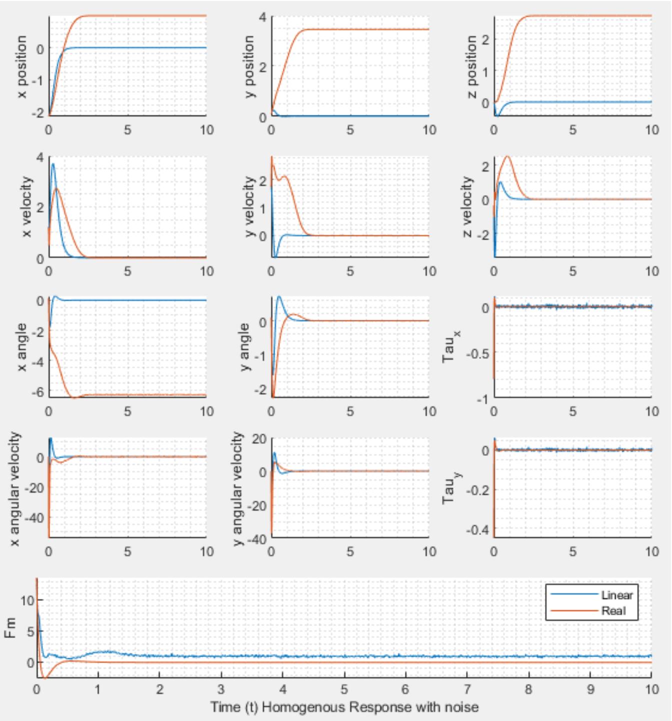


Fig. 14 The homogeneous response of the state feedback nonlinear system with Eigenvalue placement controller

J. *Observability*

The drone system is not observable, the rank of observability matrix is 8<10. The x and y position are not observable with barometers, accelerometers and gyroscope. (handout #4 2.1).

The Kalman decomposition is:

$$A_k = \begin{bmatrix} -0.005 & 0^- & 0 & 1 & 0 & 0^- & 0 & 0 & 0^- & 0 \\ 0^- & 0^- & 0 & 0^- & 0 & -9.810 & 0 & 0 & 0^+ & 0 \\ 0 & 0 & 0^+ & 0 & 9.81 & 0 & 0 & 0 & 0 & 0 \\ 0^- & 0^- & 0 & 0.005 & 0 & 0^- & 0 & 0 & 0^- & 0 \\ 0 & 0 & 0^- & 0 & 0^- & 0 & 1 & 0 & 0 & 0 \\ 0^+ & 0^+ & 0 & 0^- & 0 & 0^+ & 0 & 1 & 0^+ & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0^+ & 0 & 0 & 0^- & 0 & 0^+ & 0 & 0^- & 0^+ & 0 \\ 0^- & -1 & 0 & 0^- & 0 & 0^+ & 0 & 0^+ & 0 & 0 \\ 0 & 0 & 1 & 0 & 0^- & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_k = \begin{bmatrix} 0^- & 0 & 0^- \\ 0^- & 0 & 0^- \\ 0 & 0 & 0 \\ 10 & 0 & 0^+ \\ 0^- & 0 & 0 \\ 0 & -10000 & 0 \\ 0 & 0 & -10000 \\ 0^+ & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$C_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0^- & 0 & -1 & 0 & 0 \\ -0.995 & 0 & 0 & -0.005 & 0 & 0^+ & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0^+ & 0^+ & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The system is not detectable, because the unobservable part has zero eigenvalues.

To make the drone x and y direction position detectable, a GPS sensor can be added, e.g. A9G GPRS/GPS module. (handout #4 2.2).

An observer was design based on the linearization system, Fig 14. The linear observer is used to estimate the states in the nonlinear system. The setup of "a linear observer on a linear system" is not implemented, since the system exactly match, and the controller response is so quick. The simulation would be meaningless since the would match perfectly.

The observer gain, L, was obtained by LQR of the dual system with Q:R=10000.

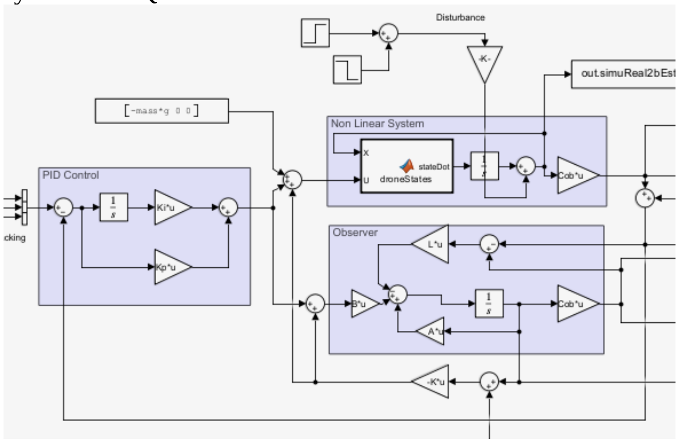


Fig. 15 Linear observer added to the nonlinear system to estimate the states and a state feedback contoller.

The linear observer got a good estimation on x or y position, exact match with the nonlinear system. The angles estimation is reasonable. A constant deviation was observed in z direction before adding a PID output controller. After that the z direction deviation as diminished. (handout #4 2.4&2.5)

I was not able to obtain the inverse of steady state gain from τ_x *and* τ_y to x and y position. So, I used state feedback in controlling x and y position; I used the output feedback in controlling the z position.

I checked drone controller implementation samples online, and found people use a separate position control block to translate x and y distance to rotational motions for tracking [2]. Therefore, for simplicity, in this project, only state feedback is used.

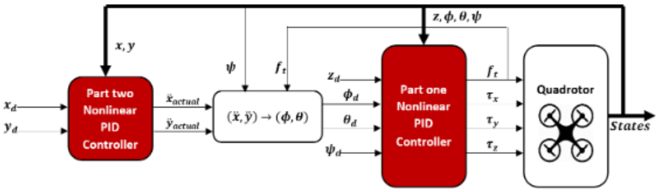


Fig. 16 A 2-stage controller suggested by Najm et. al., in which the first controller converts translational motion targets to drone rotations.

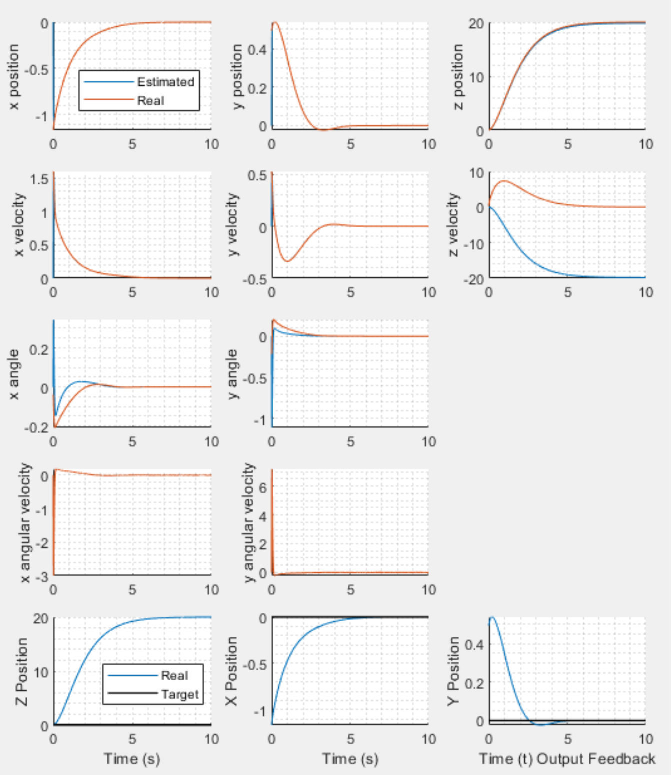


Fig. 17 Simulation result of homogreneous response of a drone system with nonzero initial states with linear observer and a state feedback contoller, as well as a PID output feedback controller.

K. Tracking

A tracking route was added for the drone. The drone is expected to raise slowly, and draw circles in x and y plain, 1m radius, 10s per round:

- Z=tanh(t)
- X= Z*sind(t*36)
- Y= Z*cosd(t*36)

The simulation result is as follows: (handout #4 2.6)

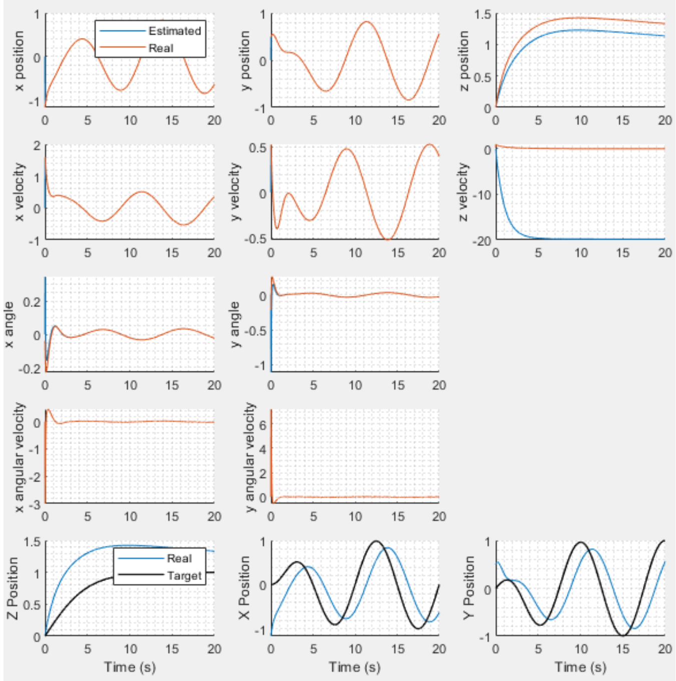


Fig. 18 Tracking result using LQR controller on linear observer.

The z axis has an overshoot, and x y direction has a phase lag when compared with the tracking target.

L. Disturbance during tracking

1) Disturbance in horizontal translational direction

Assume a disturbance as wind making the drone to drift in x direction, from T=5 to T=6:

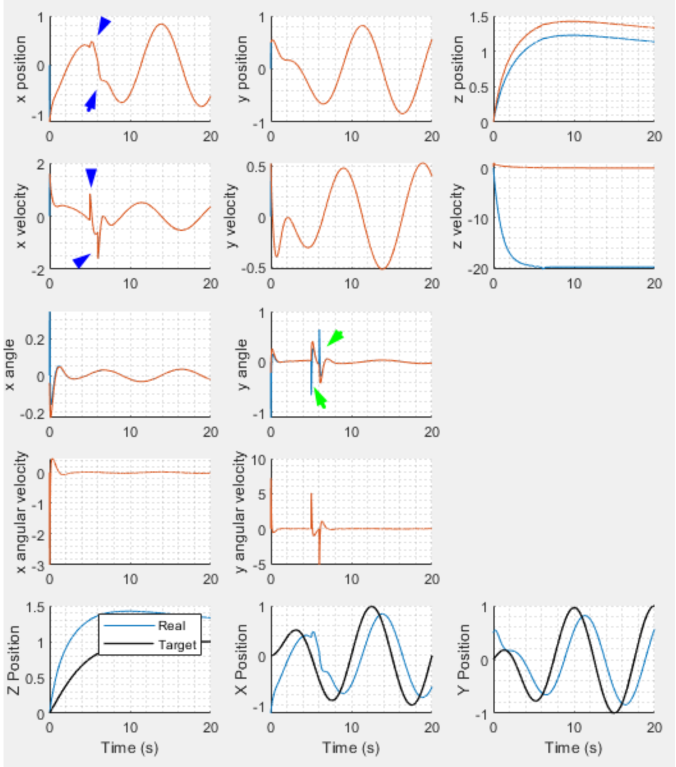


Fig. 19 Disturbance in horizontal translational direction (blue arrows). The disturbance is quickly canceled out with a y axis rotation (green arrows).

2) *Disturbance in rotational direction*

Assume a disturbance as wind making drone rotate in x rotation direction, from T=5 to T=6:

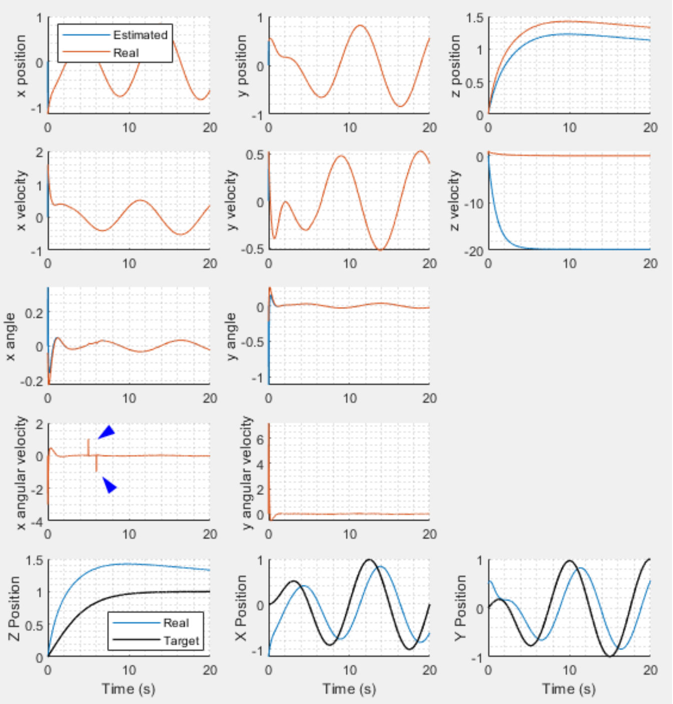


Fig. 20 Disturbance in horizontal translational direction (blue arrows). The disturbance is quickly canceled out.

We can see the disturbance is instantly compensated.

3) *Disturbance in vertical translational direction*

Assume a disturbance as wind making the drone to drift in z direction, from T=5 to T=6:

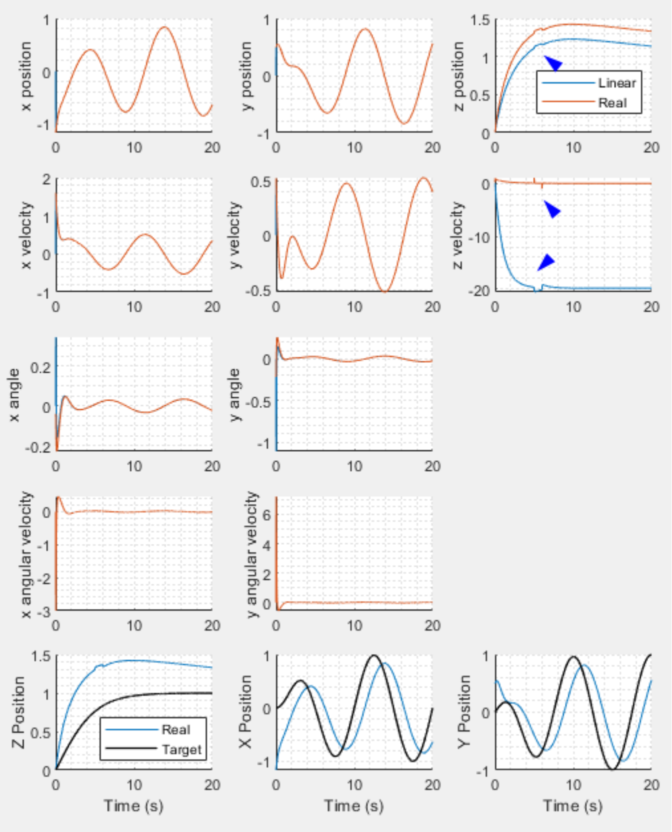


Fig. 21 Disturbance in vertical translational direction (blue arrows). The impact on velocity is visable but no huge inpact on z position.

M. *Tracking under noise on output measurements*

Measurement noise was added to the output from the nonlinear drone system. The noise includes the accelerometer, barometer, gyroscope and GPS.

The noise of accelerometer was obtained from the datasheet of MPU6050, where the noise level is estimated as 0.04g/sqrt(Hz). Here we use 100Hz sampling rate. The noise RMS is 0.4g. The noise of gyroscope is also from the datasheet and estimated as 0.005degree/s/sqrt(Hz), which equals to 0.0087rad/s.

The noise level of barometer SPL-001 was obtained from the datasheet. The average noise level is expected to be 2.5pascal RMS, which is equivalent to 1.1m at sea level.

The noise level of GPS was obtained by experiment. The A9G module was configured with GPS mode on, and sampling rate was set to 1Hz. The module was placed on balcony and at least 10 satellites were connected. The location data was logged for 30 minutes. The standard deviation of the position data was 3.9m, larger than 2.5m that claimed in the datasheet.

The simulation result is amazing, the system is stable and steadily following the target trajectory. When look at the 3D trajectory comparison in figure 18,

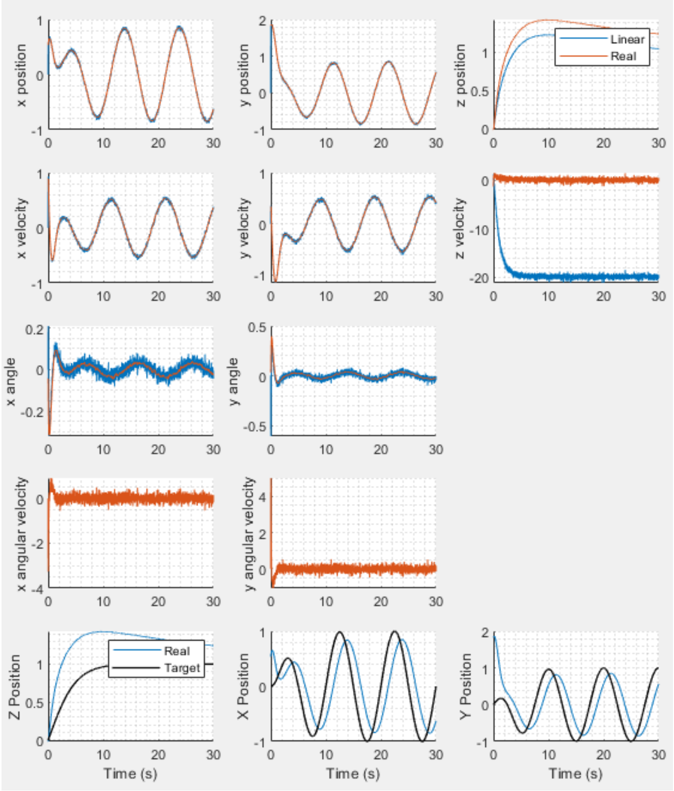


Fig. 22 Tracking the route with noise in the measurement of outputs.

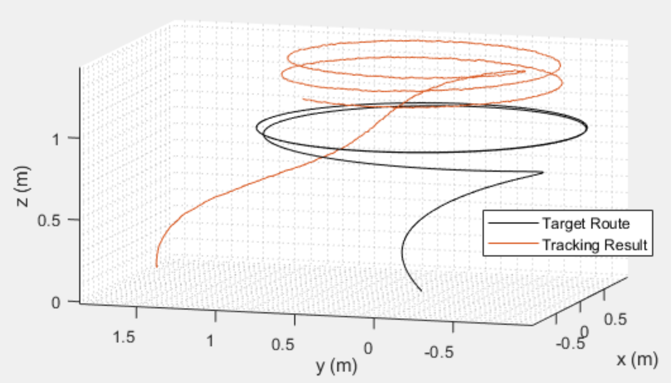


Fig. 23 3D trajectory comparison between tracking result and tracking target. We can observe a overshoot in z direction and undershoot in x and y direction. Small noise andnonzero initial condition can also be observed.

VII. Summary

A LQR controller was developed for the headless drone system based on linear observer. The project was far more complex than I expected. Yet the linear observer performed amazingly well under noise and disturbance, which is also far beyond my imagination. To summarize, this is a fruitful but tough project.

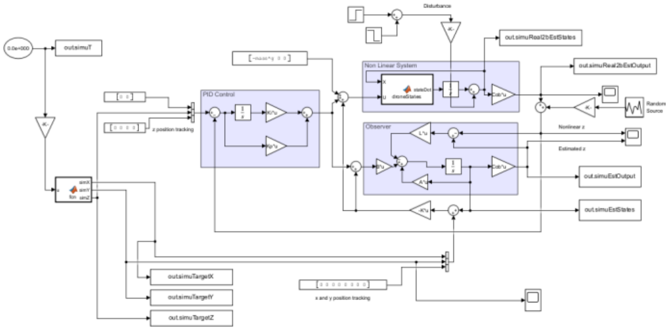


Fig. 24 The final system configuration with output and state feedback.

References

[1] Tesch, Douglas A., Diego Eckhard, and William Cechin Guarienti. "Pitch and roll control of a quadcopter using cascade iterative feedback tuning." IFAC-PapersOnLine 49.30 (2016): 30-35.

[2] Najm, Aws Abdulsalam, and Ibraheem Kasim Ibraheem. "Nonlinear PID controller design for a 6-DOF UAV quadrotor system." Engineering Science and Technology, an International Journal 22.4 (2019): 1087-1097.