# Project Goal

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.

# Model Definition

A diagram of a quadcopter

Description automatically generated

Figure 1 The definition of axis and variables. Image from Douglas et. al. 2016.

A diagram of a drone

Description automatically generated

Figure 2 Explanation how yaw generated in drone. Figure from Sabatino, 2015

Sabatino described the dynamics of the actuators as Equation 1, where is the total force generated; 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.

A math equations with numbers

Description automatically generated with medium confidence

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

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

Also, to simply the model, wind effect is not considered in this report.

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

1. where **pink** line crossed variables are simplified because of .
2. where **green** line crossed variables are simplified because wind factors are not considered.
3. blue box indicates the input to the system.

A math equations on a white background

Description automatically generated

Equation 3 The quadrotor system dynamics equation defined based on  
state variable that . Sabatino, 2015.  
 for cos(); for sin(); for tan().

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

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

Define the states variables as, the input is defined as. Then the nonlinear state space system can be described as Equation 5:

Linearization

An equilibrium point is found by set :

From Eq 4.6,

From Eq 4.5,

From Eq 4.7,

From Eq 4.3,

From Eq 4.4,

Therefore, the equilibrium point is , where ,

and

The linearization can be calculated as follows:

At the equilibrium point, with the assumption that when is small, :

Output Definition

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

Therefore and

Summary

The system has 3 inputs:

The system has 10 states:

The system has 5 outputs:

(handout #2 3.1)

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

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)

A screenshot of a graph

Description automatically generated

Figure 3 Open loop simulation of the drone system. Exponential deviation from equilibrium point can be observed on the nonlinear system (orange line).

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

Step in Fm:

A screenshot of a graph

Description automatically generated

Figure 4 The step response in total pulling force. The linear and nonlinear system go to opposite directions.

Step in

A screenshot of a graph

Description automatically generated

Figure 5 A step torque input was applied on x direction. A constant increase of angular velocity can be observed in x direction.

Step in

A screenshot of a graph

Description automatically generated

Figure 6 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)

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.

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:

Dimension of ,,,.

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

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.

A screenshot of a graph

Description automatically generated

Figure 7 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.

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

A screenshot of a graph

Description automatically generated

Figure 8 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)

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

A screenshot of a graph

Description automatically generated

Figure 9

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

A screenshot of a graph

Description automatically generated

Figure 10

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

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

A screenshot of a graph

Description automatically generated

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.

A screenshot of a graph

Description automatically generated

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

The Kalman decomposition is:

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. 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 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)

A screenshot of a graph

Description automatically generated

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)

A screenshot of a graph

Description automatically generated

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

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

A screenshot of a graph

Description automatically generated

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

A screenshot of a graph

Description automatically generated

We can see the disturbance is instantly compensated.

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

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A graph of a target

Description automatically generated

# Reference

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.

Sabatino, Francesco. "Quadrotor control: modeling, nonlinearcontrol design, and simulation." (2015).