Learning Quadcopter Flight Dynamics

We denote the position of the quadcopter with x, y, z [m],
the orientation by ϕ, θ, ψ [rad]
and the angular speed of the four rotors as
$\mathbf{\Omega} = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]$ [rad/s].

Task 1, estimating k

Estimating k using $k_{est} = K_p * m$.
Which is derived from $m\ddot{z} = F = 4 \cdot k\Omega^2 - mg$
where $\Omega_i = \Omega$ for all rotors,
and by letting the $u = 4\Omega^2$ we get $\zeta = \ddot{z} + g = \frac{k}{m} * u$
This gives an estimate on the form $\zeta = K_p u$
from which we can estimate k as $k_{est} = K_p \cdot m$.
This results in code such as `k_est = sys2.Kp * m`.

Which results in $k_{est} = 2.2067e-08$, and the true $k = 2.2000e-08$.

This difference is $\approx 6.7e-11$ which is ~0.3 % of the true value. The estimation is very close to the real value and can be considered a good estimation.

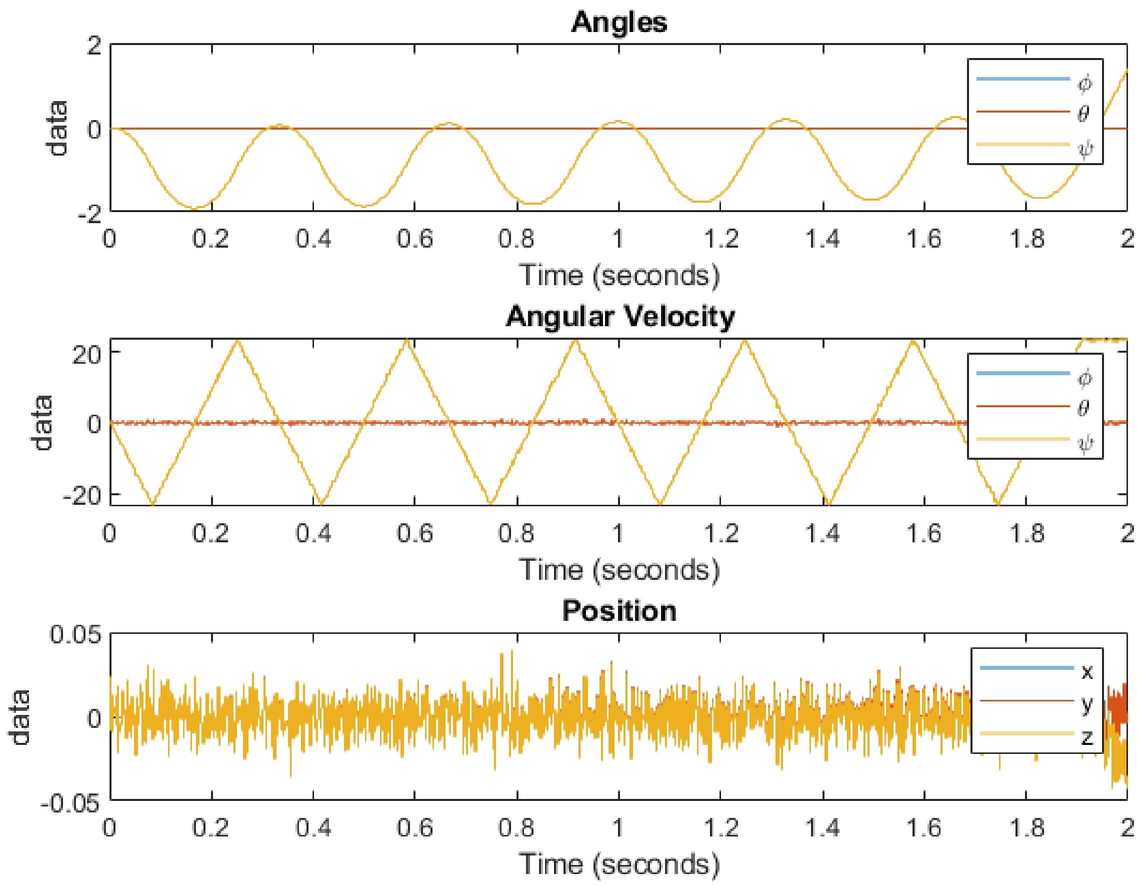
Task 2, estimating I_3 and b

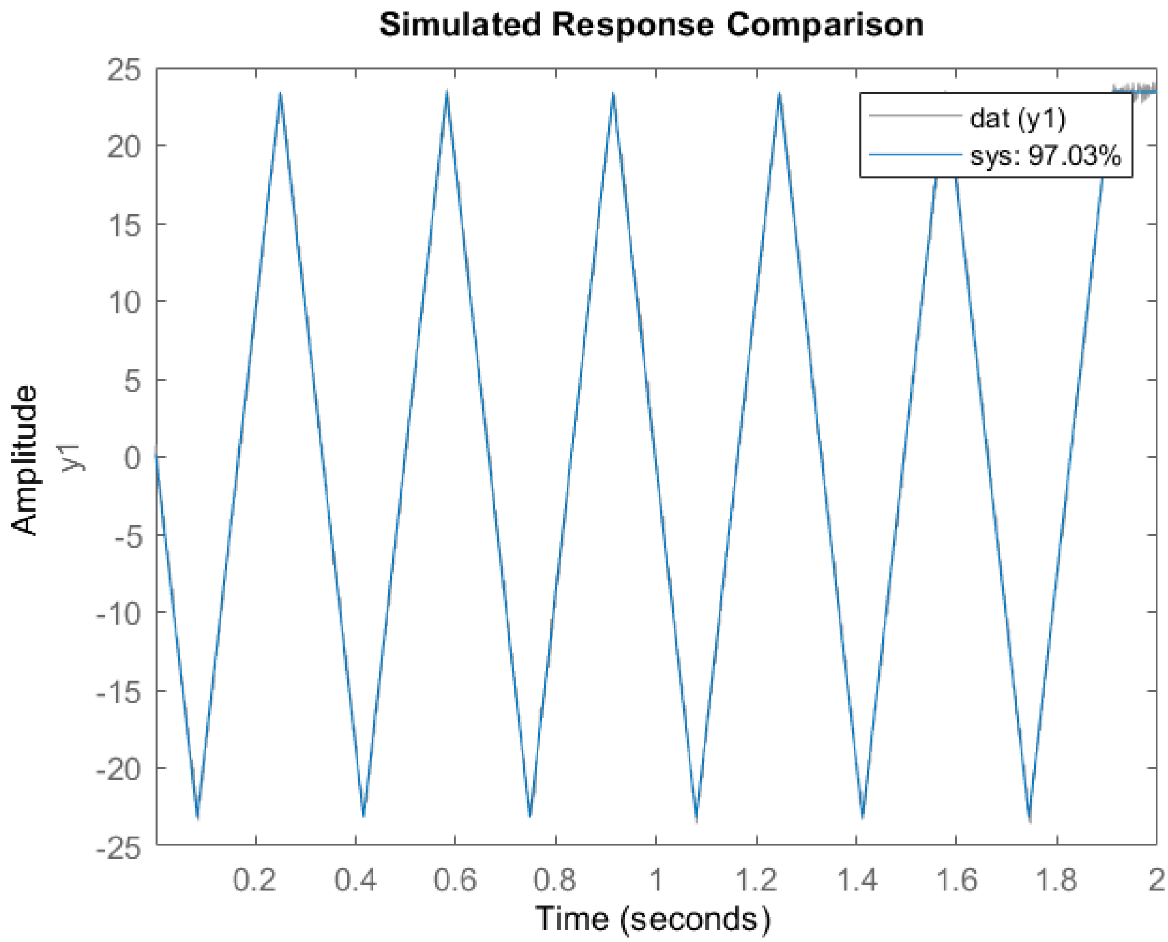
By setting $\Omega_1 = \Omega_3$ and $\Omega_2 = \Omega_4$ the resulting torque profile will be that there is only a torque around the z-axis. That gives the equation of motion for the yaw angle as
$\tau_\psi = I_3\ddot{\psi} = b_i(-2\Omega_1^2 + 2\Omega_2^2)$.

Let $u = (-2\Omega_1^2 + 2\Omega_2^2)$ then the dynamics for $\dot{\psi}$ are given by $\frac{d}{dt}\dot{\psi} = \frac{b}{I_3}u$

To estimate b and I_3, we make the quadcopter hover. This is done by letting $\Omega_2 = c\omega_1$ and setting ω_1 from the quation $mg = k(2\omega_2^2 + 2\omega_1^2)$.
Combining these equations into $\Omega_1 = \sqrt{\frac{mg}{2k(c^2+1)}}$ and using this to set c and the number of segments resulting in a hovering quadcopter, and the yaw angle ψ in the range $\pm\pi$.

After some trial and error the values of $c = 0.7$ and the number of segments $= 12$ were found. Producing the following two graphs where we can read that the z position barely changes.


Angles
data
Time (seconds)
ϕ
θ
ψ
Angular Velocity
data
Time (seconds)
ϕ
θ
ψ
Position
data
Time (seconds)
x
y
z


Simulated Response Comparison
Amplitude
y1
Time (seconds)
dat (y1)
sys: 97.03%

With these results, the b and I_3 can be estimated.
Due to the proportional relation between them, $\frac{b}{I_3} = K_p$, only one of them can be estimated at a time,
meaning one of them have to be estimated in another way. It is therefore assumed that I_3 is estimated elsewhere/previously. Applying this to the code results in code as `b_est = Kp * I3`.

Which results in $b_{est} = 1.9969e-09$ and the true $b = 2.0000e-09$.

This difference is $\approx 3.1e-12$ which is ~$0.2\ \%$ of the true value. The estimation is very close to the real value and can be considered a good estimation.

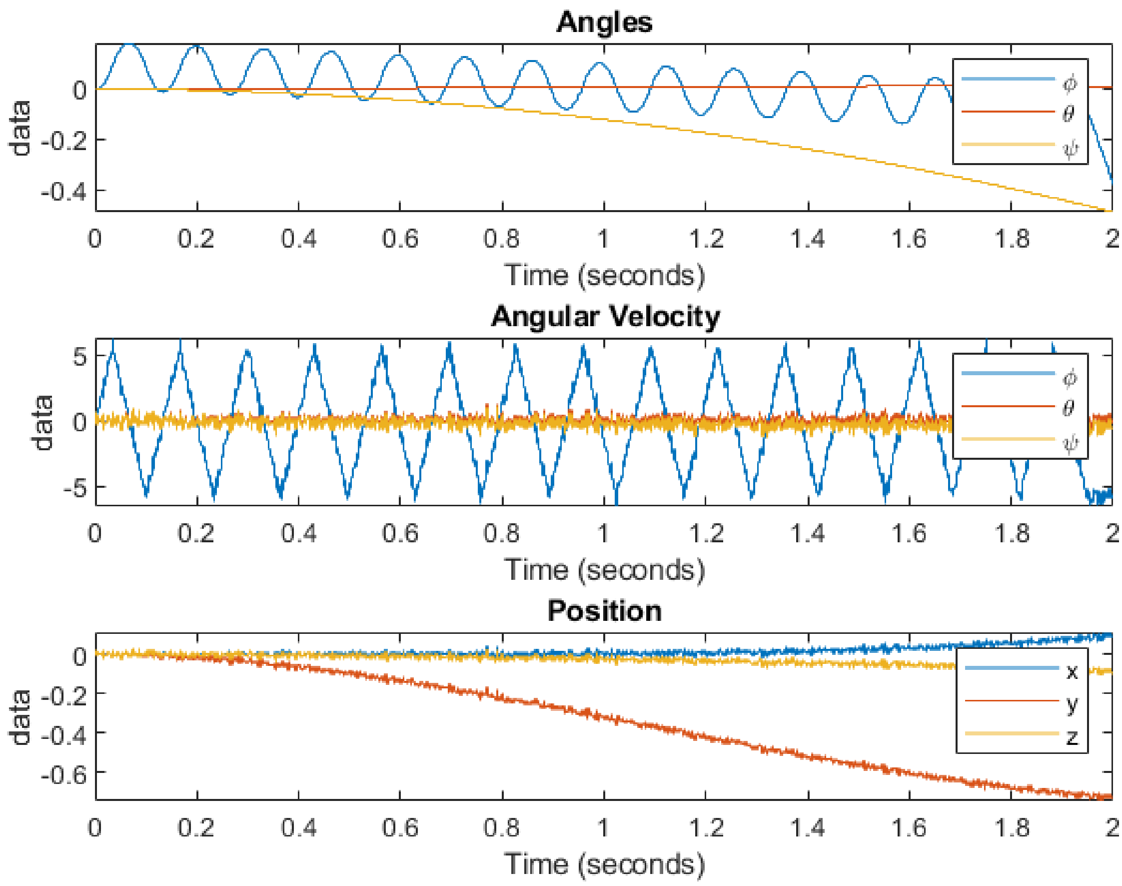
Task 3, estimating I_1 and I_2

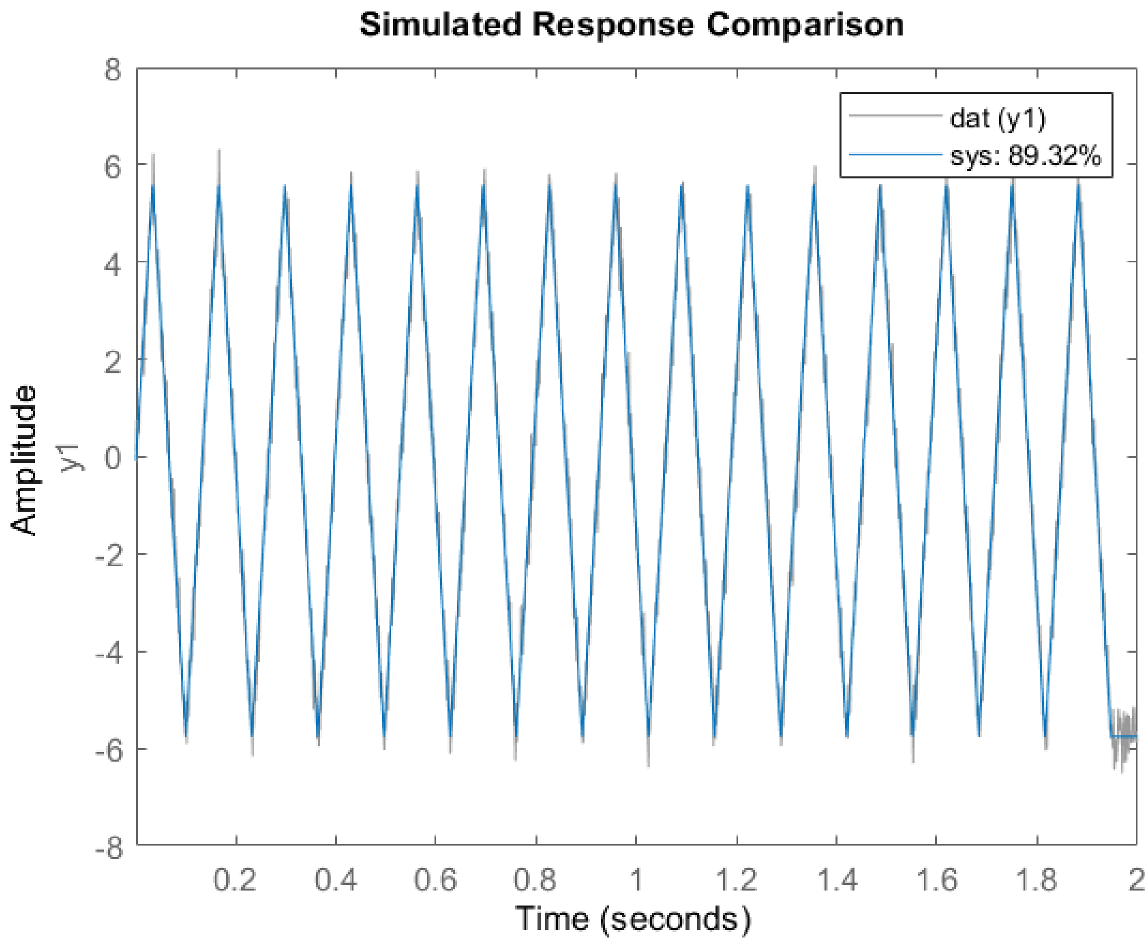
Due to the summetry of the quadcopter, it holds that $I_1 = I_2$,
so we proceed estimating only one of them.
The equations of motion for the ϕ angle is given by
$\tau_\phi = I_1\ddot{\phi}$, where for small angles, the torque is given by
$\tau_\phi = kl(-\Omega_2^2 + \Omega_4^2)$.
If $\Omega_1 = \Omega_3$ and $-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2 = 0$ there will only be a torque on ϕ.

Let $u = (-\Omega_2^2 + \Omega_4^2)$, then $\dot{\phi}$ will satisfy $\dot{\phi} = \frac{kl}{I_2}\frac{1}{s}u$.

To estimate I_1 we find Ω_H where $\Omega_H = \Omega_i$ so that the quadcopter hovers. By rewriting $mg = k(2\Omega_1^2 + 2\Omega_2^2)$ using Ω_H we get
$\Omega_H = \sqrt{\frac{2\Omega_H^2}{4k}}$
With this, the quadcopter is hovering.
Adding a new constraint, $\Omega_4 = \Omega_2$, which implies that
$2\Omega_H = \Omega_2^2(1 + c^2)$ must hold in order to keep the quadcopter hovering.

This can be made into $\Omega_2 = \sqrt{\frac{2\Omega_H^2}{1+c^2}}$,
which is a function of c.
Finding a value of c and the number of segments by satisfying that the ϕ angle stays small.
Using $c = 0.6$ and the number of segments $= 30$ produces the two graphs below.


Angles
data
0
-0.2
-0.4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (seconds)
ϕ
θ
ψ
Angular Velocity
data
5
0
-5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (seconds)
ϕ
θ
ψ
Position
data
0
-0.2
-0.4
-0.6
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (seconds)
x
y
z

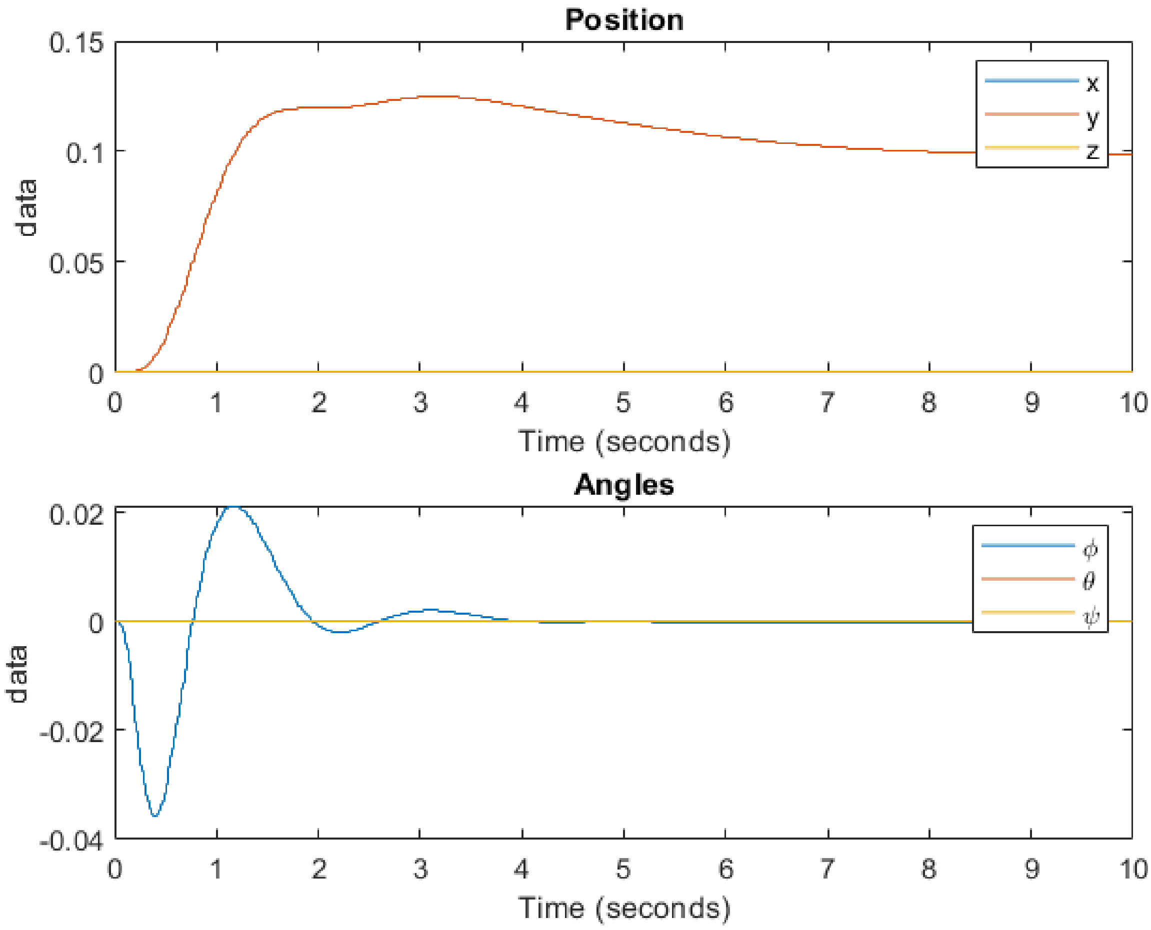

Simulated Response Comparison
Amplitude
y1
8
6
4
2
0
-2
-4
-6
-8
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Time (seconds)
dat (y1)
sys: 89.32%

Using the equations introduces in this task, $I_{est} = \frac{kl}{K_p}$
results in $I_{est} = 1.6667e-05$ with the true $I = 1.6600e-05$.

This difference is $\approx 6.7e-08$ which is ~0.4% of the true value.
The estimation is very close to the real value and can be considered a good estimation.

Task 4, closing the loop

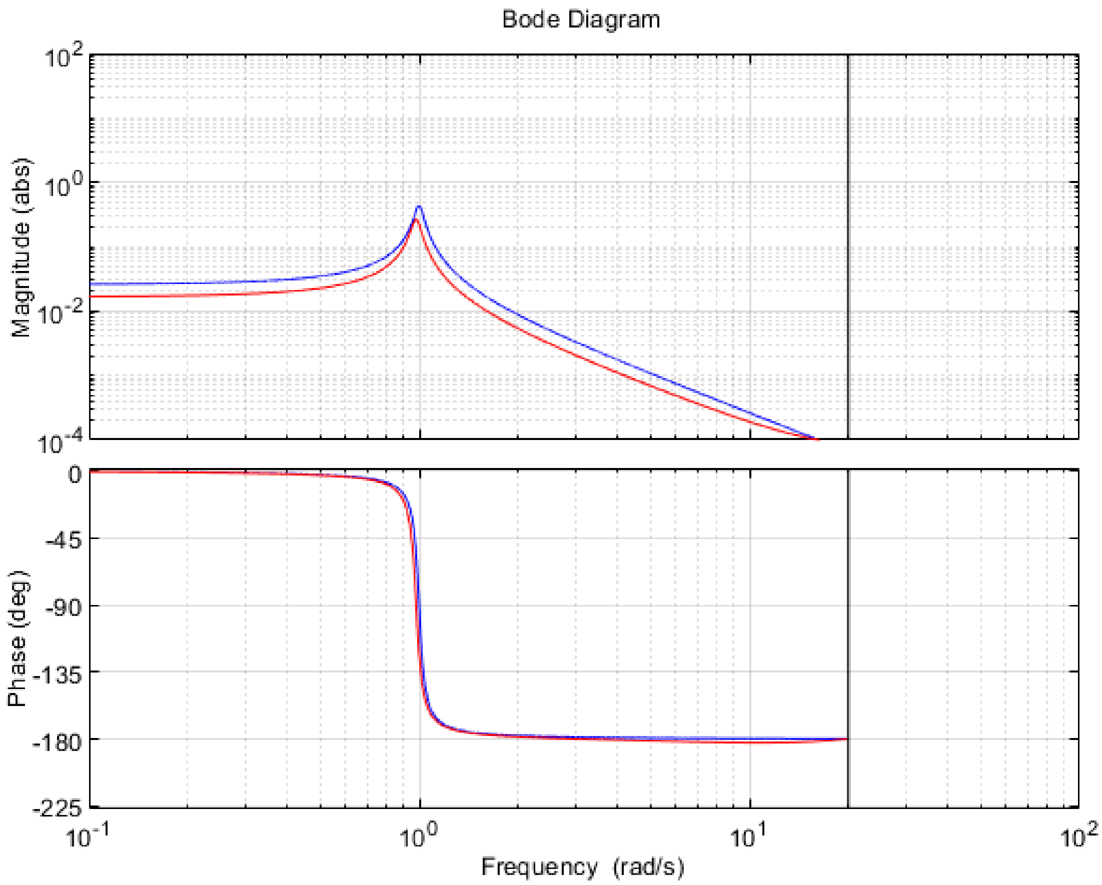
The fourth task was to evaluate and confirm that the quadcopter moved 0.1m in the y direction and maintaining the z level. Confirming this, the graph below show the position and angles.

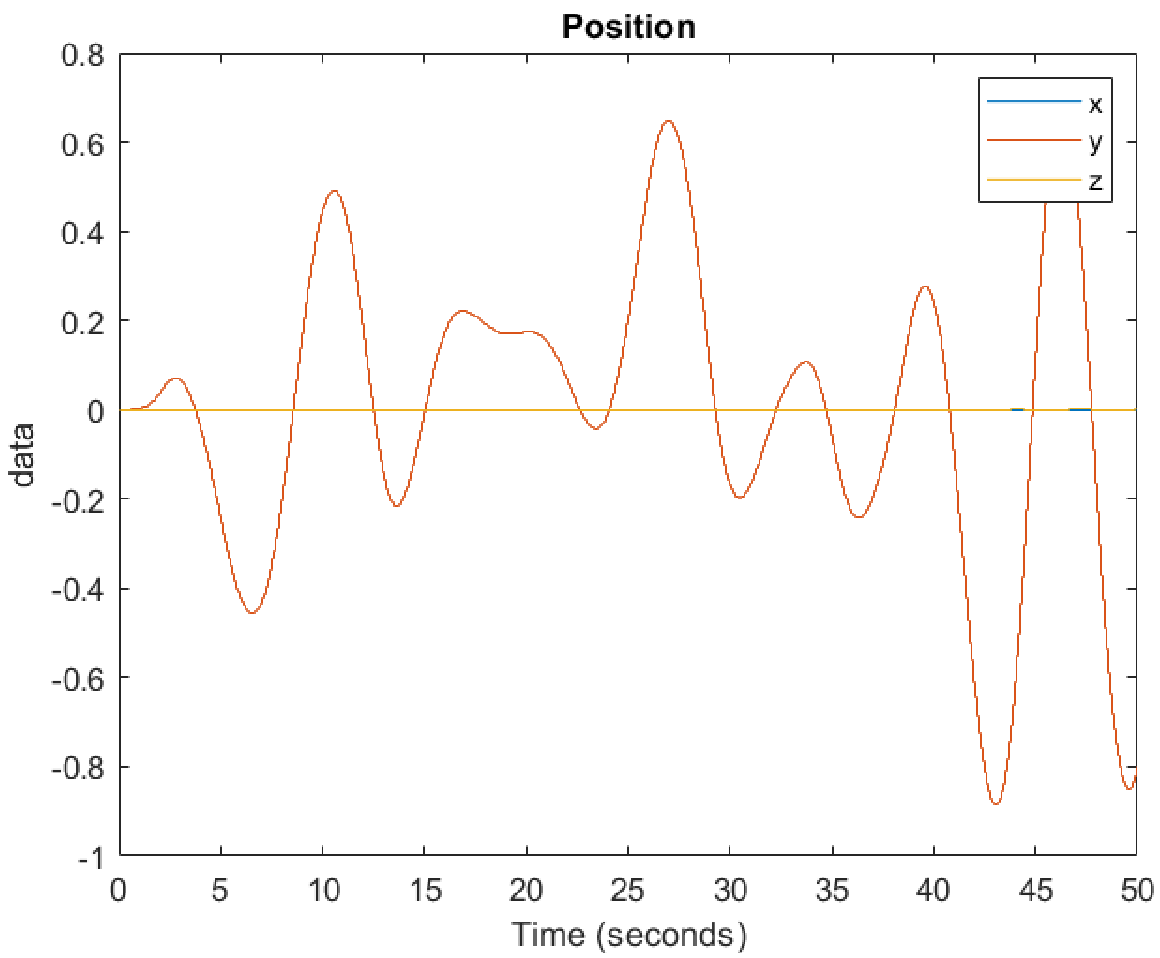


Task 5, estimating a disturbance model

We now consider when the wind is acting on the quadcopter, resulting in a force in the y direction. With the goal of keeping the quadcopter in the center/origin. To do this we estimate with an ARMA model, using the acceleration as measurement and deciding the decimating factor R.

Inspecting the illustration of the disturbance in Figure 3 in the Lab manual, the structure for the ARMA was set to $na = 2$ and $nc = 1$.
Then, by trial and error, the decimating factor R was set to reproduce an estimation of the true denominator.

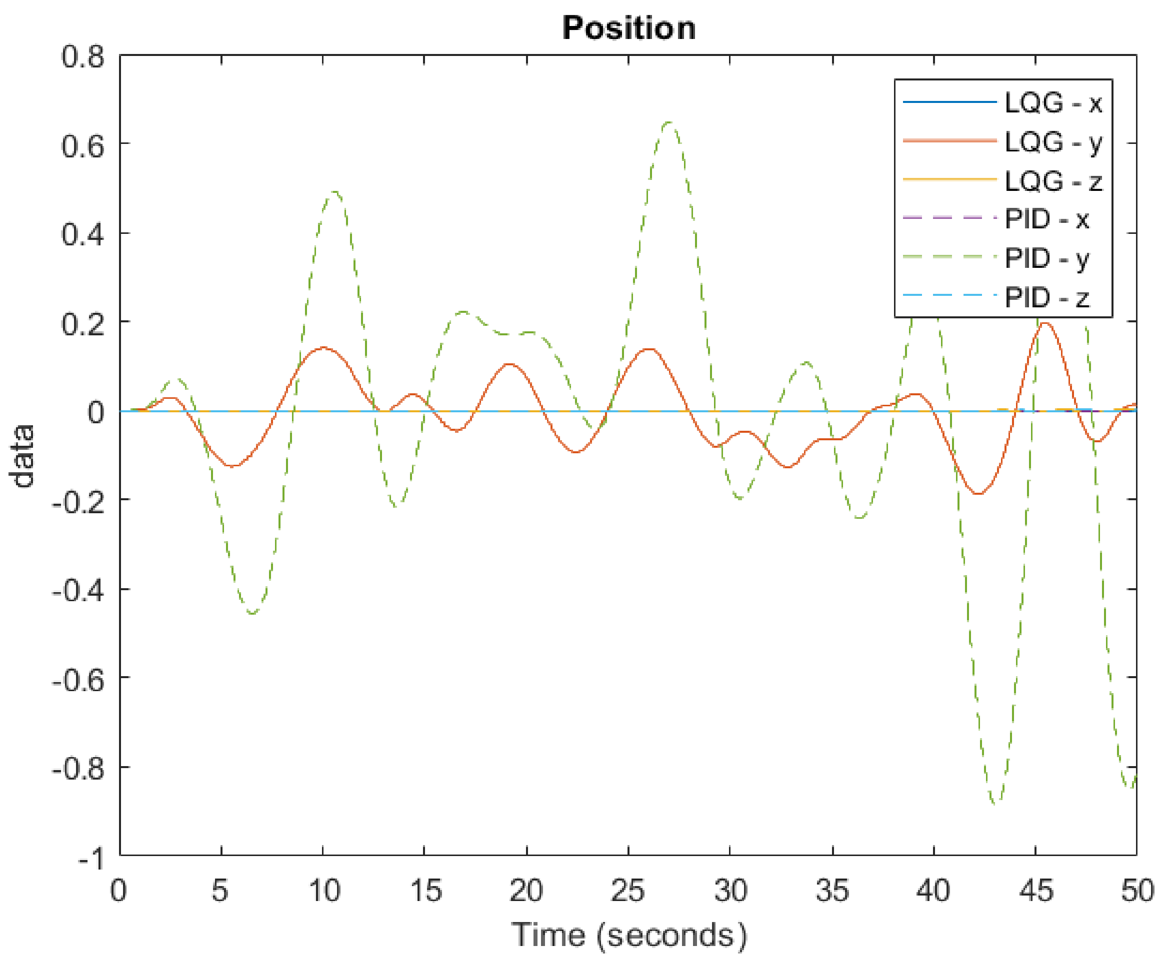

Bode Diagram
Magnitude (abs)
Phase (deg)
Frequency (rad/s)
10^2
10^0
10^-2
10^-4
0
-45
-90
-135
-180
-225
10^-1
10^0
10^1
10^2


Position
data
Time (seconds)
x
y
z
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25
30
35
40
45
50

Which results in $R_{est} = 9.45$ and estimated denominator $s^s + 0.06036s + 0.96$ with the true denominator $s^2 + 0.060s + 1$. This produces the two graphs above, showing that it was a good estimation. Also note in the second graph that the position is quite affected by the wind.

Task 6, using the disturbance model in a control design

Implementing an LQG controller was the last task of this lab. This was done by inserting previously calculated values for ω_0 and ϵ, which results in $\omega_0 = \sqrt{0.96} = 0.9798$ and $\frac{0.06036}{\omega_0} = 0.0308$.



Using this produces the graph above, in which we can see that using the disturbance in the control design drastically improves the controller.