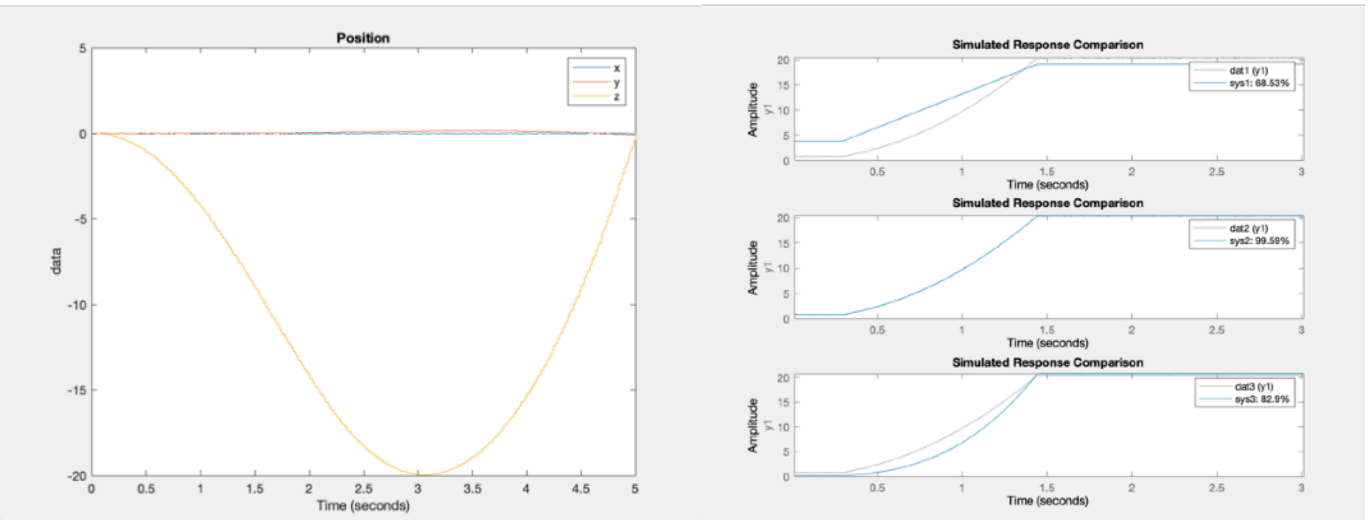
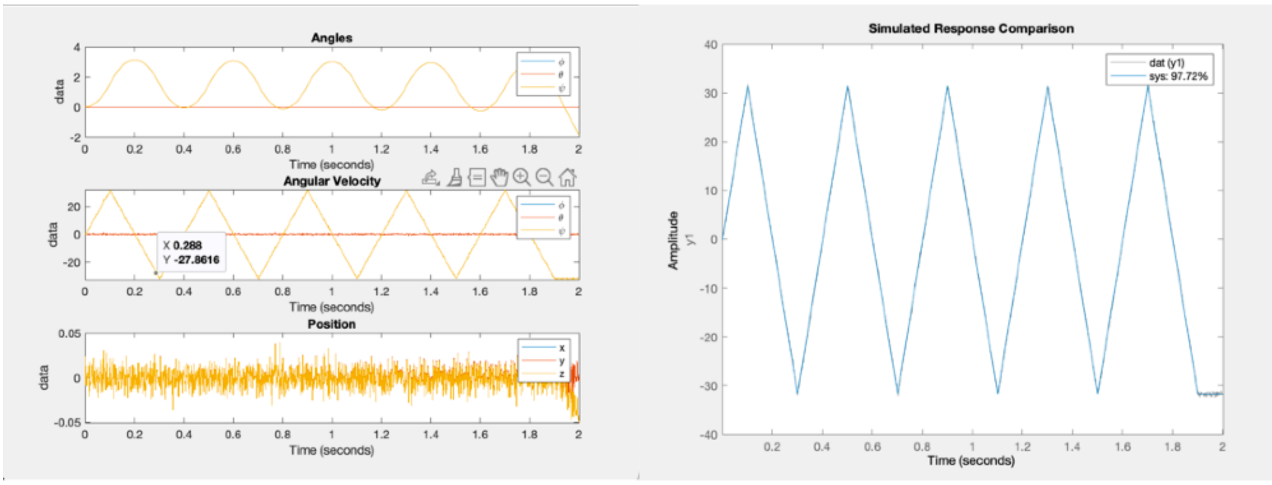
Task 1:
We begin by estimating the coefficient k for the rotor force. From the equations in section 2 one could expect the most suitable relation to be $f_i = k\Omega^2$, and it turns out to be the best fit when simulating. I simulate for rotor speeds in the interval $[0,10^5]$ which gives a satisfying excitation of states, it slowly falls in the beginning but for higher rotor speeds it starts to gain altitude. See the trajectory and simulated responses below:

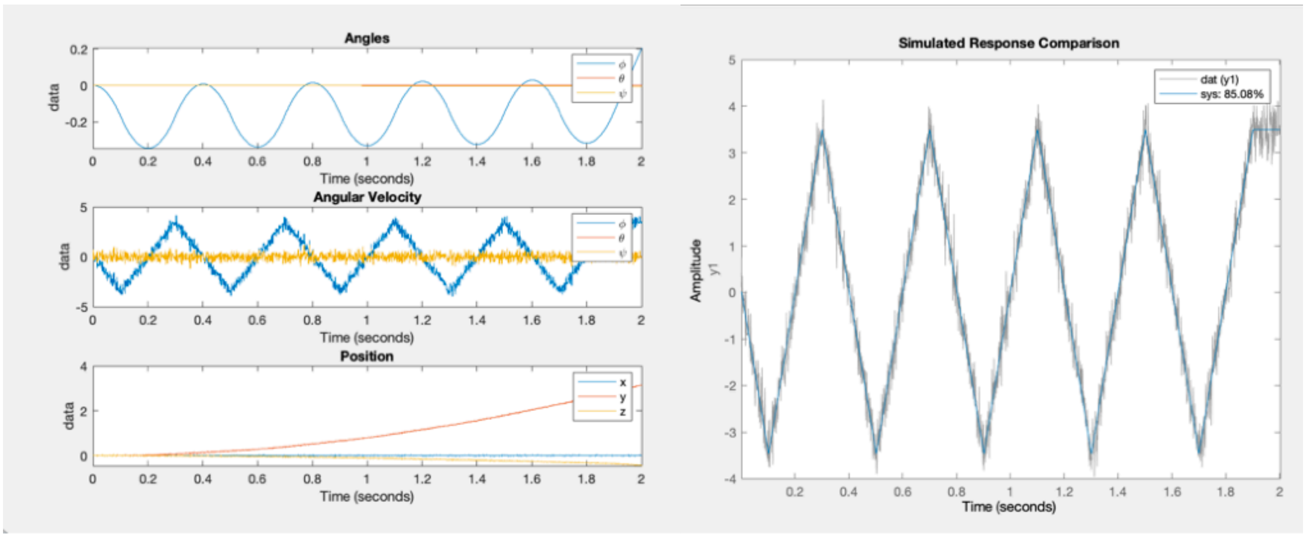


Task 2:
In order to have a hovering quadcopter, the total thrust from the rotors should equal the gravitational force. In order to make it yaw, we choose the speed of two diagonally opposite rotors to be equal, and set the speed of two of them proportional to the other two. This creates a resulting torque around the z-axis which makes the quadcopter yaw. We get $\Omega_1 = \sqrt{\dfrac{mg}{2k(1+c^2)}}$ and $\Omega_2 = c\Omega_1$. Using $c = 1.5$ and ten segments we get a simulation and response:



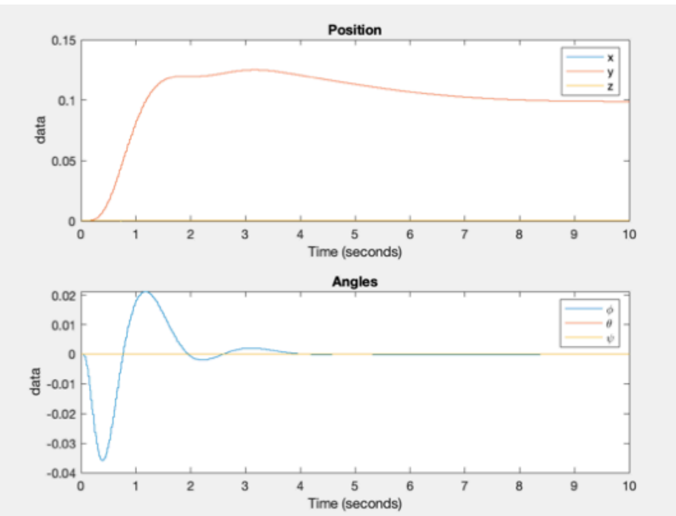
The altering in z-dimension is very small and we get satisfying yaw angles. Since we only have one equation and measured relation including b and I_3 we cannot find both of them from this experiment. Since we have the input $u = -2\Omega_1^2 + 2\Omega_2^2$ and the dynamics $\dfrac{d}{dt}\dot{\psi} = \dfrac{b}{I_3}u$ we retrieve our estimate of b as $K_p I_3$, where K_p is the estimated coefficient from simulations.

Task 3:
For hovering we again want the rotor thrust to equal the gravitational force. In order to alter the pitch, we now instead want to fulfil $\Omega_1 = \Omega_3$ and $\Omega_1 + \Omega_3 = \Omega_2 + \Omega_4$. Choosing the hover rotor velocity $\Omega_1 = \Omega_3 = \Omega_H = \sqrt{\frac{mg}{4k}}$, $\Omega_2 = \sqrt{\frac{2}{1+c^2}}\Omega_H$ and finally $\Omega_4 = c\Omega_2$. We simulate using ten segments and $c = 1.1$ we get a simulation:

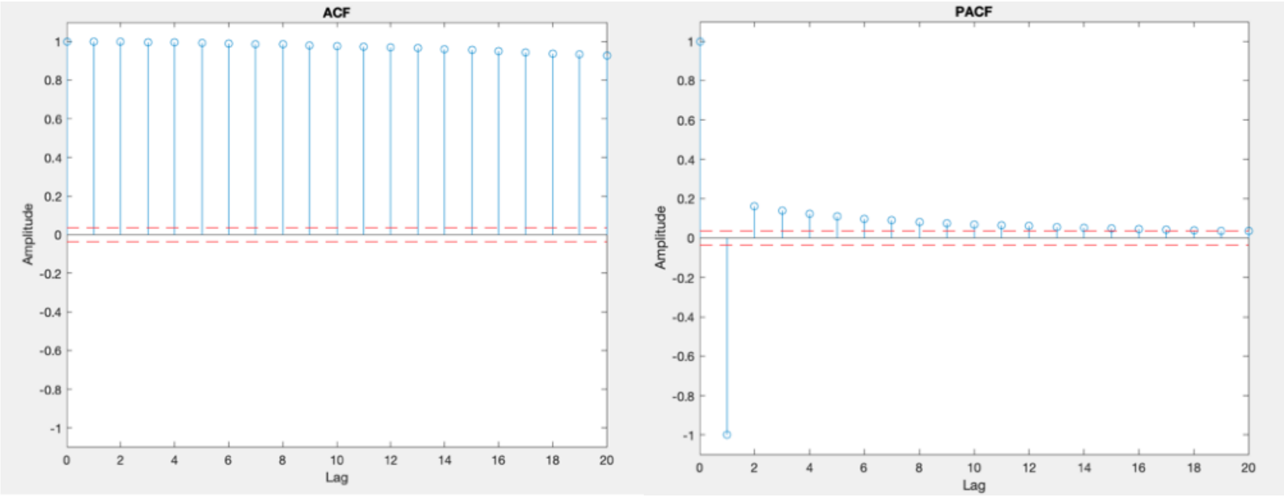


There is an apparent change of position in y-dimension and z-dimension. I believe these come from a change in pitch making the resulting force vector to have a component in the y-dimension. Since the total thrust is constant this results in a smaller component in the z-dimension, making the quadcopter lose altitude slightly.

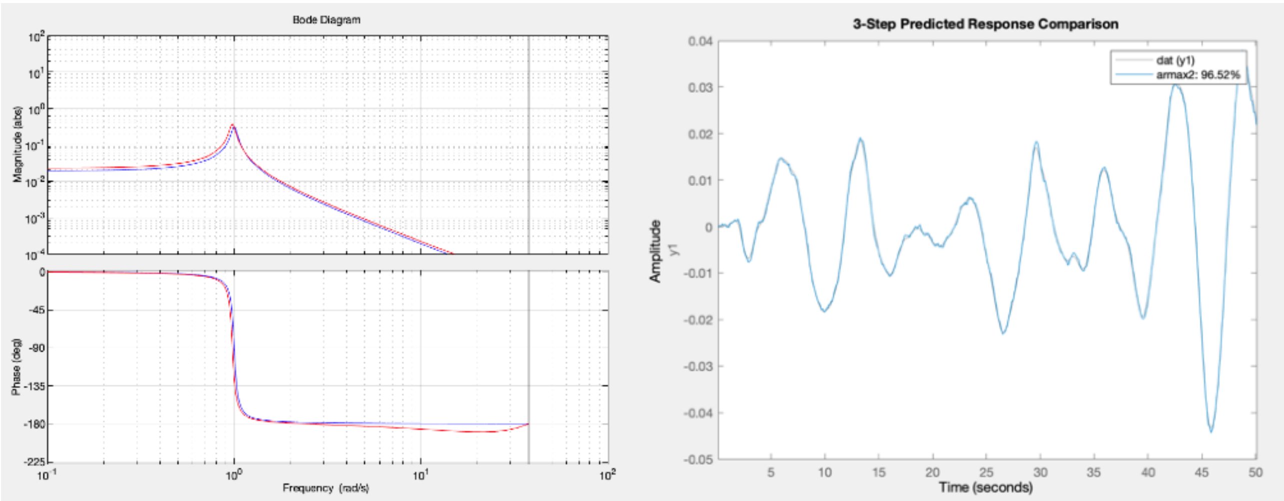
Task 4:
Running the file `test_pid.m` results in the following simulation, where the quadcopter slides slightly in the y-direction by pitching.



Task 5:
Our objective is to fit an ARMA-model to the wind. By comparing the autocorrelation function (acf) and partial autocorrelation function (pacf) below we can see that both AR- and MA-components are needed.

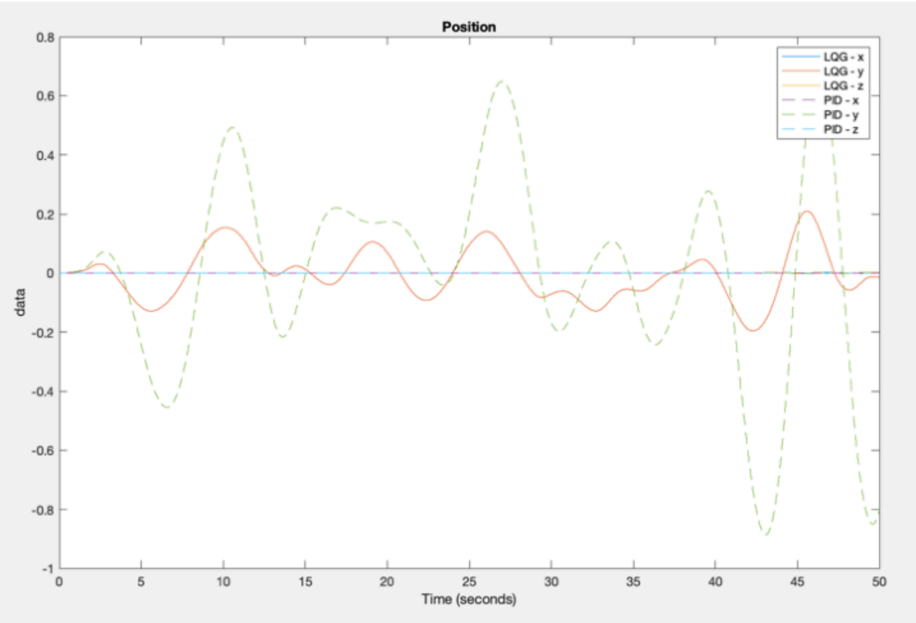


The AR should be of higher order than the MA. We try different orders and compare them to the Bode-diagram of the real process. The most suitable is an ARMA(2,1) model. We then proceed with testing different values of the decimation factor R and find R = 5 to be suitable. The resulting Bode diagrams and three step prediction response are found below.



Our resulting denominator coefficients are 0.05778 and 0.9571 which are very close to the actual 0.06 and 1.00.

Task 6:
We use our calculated coefficients together with the dynamics relations for external disturbances to find the coefficients $\omega_0 = \sqrt{0.9571} = 0.9783$ and $\epsilon = \dfrac{0.05777}{2\omega_0} = 0.0295$. When this controller is used we get a positional that varies much less than when an ordinary PID-controller is used, as evident from the graph below.



We see that the amplitude of the positional changes caused by wind disturbance in y-dimension is greater for the PID-controlled quadcopter than for the quadcopter governed by our controller. It also remains at the same position in x- and z-dimension.