Model Predictive Control

Programming Exercise Report

Spring Semester 2017

Students:

Napat Karnchanachari (16-950-800)

Pontus Grahn (16-911-133)

Valentin Yuryev (16-931-008).

Contents

[Problem Formulation 2](#_Toc484086787)

[Reference Tracking 5](#_Toc484086788)

[First Simulation of Nonlinear Model 8](#_Toc484086789)

[Offset free MPC 10](#_Toc484086790)

[Simulations on the Nonlinear model 15](#_Toc484086791)

[Slew Rate Constraints 21](#_Toc484086792)

[Soft Constraints 22](#_Toc484086793)

[Forces Pro 24](#_Toc484086794)

## Problem Formulation

1.

The 4th and 5th row in Ac matrix correspond to equations:

Since we are linearizing around the hover point, the thrusters collective force in the z direction is equal to the gravitational force which explains the 9.81 constant. Additionally, the roll and pitch angle are very small since we have to stay near the hover point. Hence, when the quadcopter increases its roll or pitch angle (β and α) the force in the x and y direction can be approximated as:

(for small β)

(for small α)

The negative sign in y direction is explained by the different in roll rotation direction and y positive axis.

The non-zero rows in Bc matrix correspond to:

The first equation depicts thruster contribution to acceleration in z direction. This makes sense since in hover all thrusters point directly down, counteracting gravity.

The second two relate thruster output to roll and pitch angle change. Since thrusters u2 and u4 are directly opposite to each other relative to the x-axis, it is clear that they are the only ones contributing to roll. Similarly, u1 and u3 only contribute to pitch angle change.

Finally, all thrusters also have some angular momentum that contribute to quadcopters yawing. Specially, u1 and u3 spin clockwise contributing to positive rotation along the z-axis and u2, u4 contribute to counter-clockwise.

2.

Q – The values depend on the importance of achieving the origin fast. Higher parameters in Q lead to higher costs incurred over time by the controller. Hence, the controller will try to minimize (get the state to origin) as fast as possible. Since we care about quick response on pitch, roll and z’, the first 3 parameters in the diagonal Q are high. The other 4 are near 0 since we do not care how long it takes to achieve them.

R – These parameters decide how conservative the controller should be with thrusters, the idea is that we do not worry about how much power the quadcopter uses to achieve our goal so the number is close to zero. Of course, if there is a battery involved we would want to penalize input more to conserve power.

P – Is terminal cost. This matrix is 100\*Q since we want to make sure the system ends up within the terminal set. Same as in Q, the weights on roll, pitch and z’ to be higher than on the rest of the states.

A – Since we are using subsystem with only the last 7 states, our A matrix consists of the 7x7 bottom right corner of the full A matrix.

B – Similarly to A, we only want to use the portion of b matrix that corresponds to the last 7 states.

N – We set N to 19 steps which roughly corresponds to 2 seconds. The reason for this is that we want the comptroller to plan to be within the terminal set within 2 seconds.

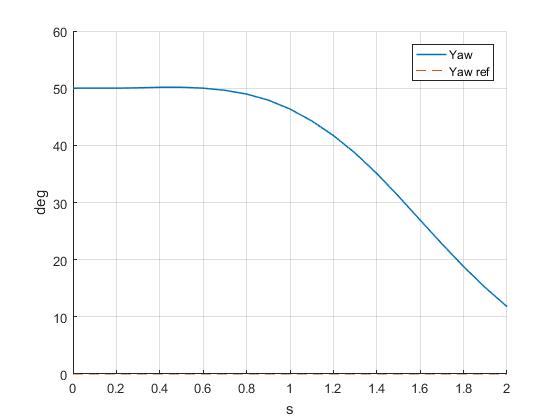
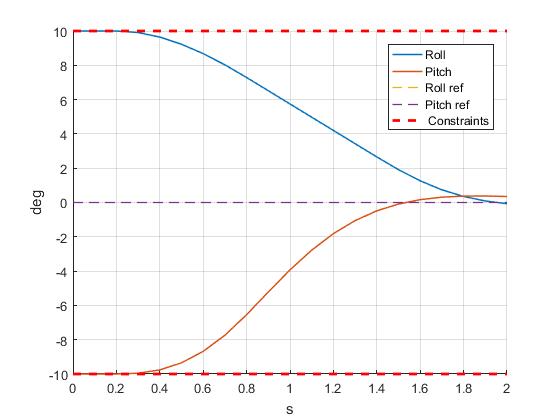
3.

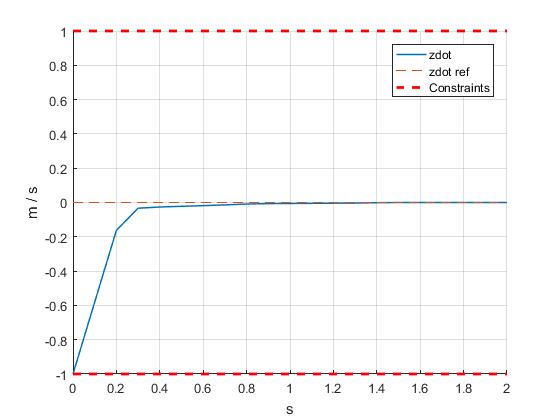
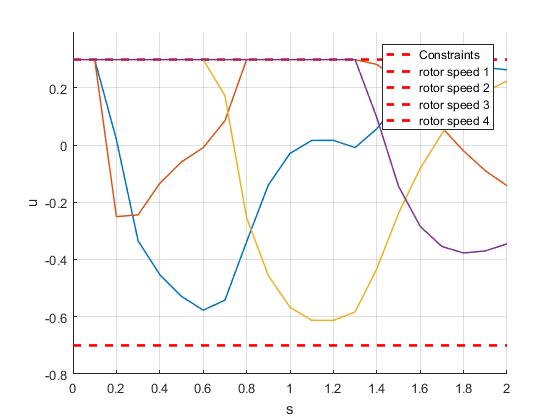
Firstly, to compute we first thought to simply compute it using the system dynamics and the terminal constraints giving us a positive invariant set within the terminal constraints. However, this did not converge.

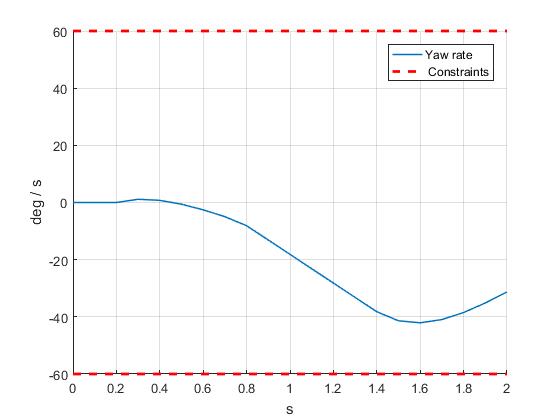
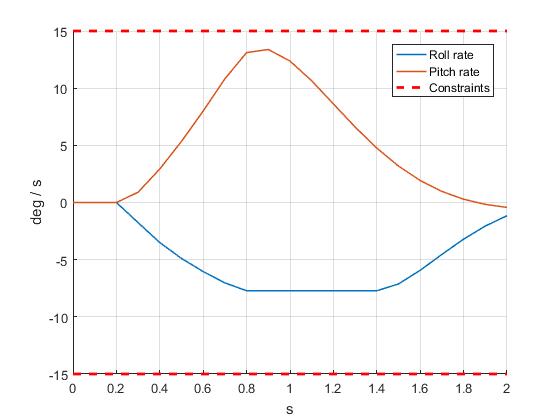
Our second approach is to compute the maximum control invariant set which is computed by considering the optimal control law and optimal cost , such that and is the solution to the discrete-time algebraic Riccati equation. Hence our system dynamics became:

However, this gave us an empty set. This where we concluded there is no invariant set contained in the terminal set hence simply computed plotted the response for a regulation problem where .

Plots of response starting from given initial condition:







## Reference Tracking

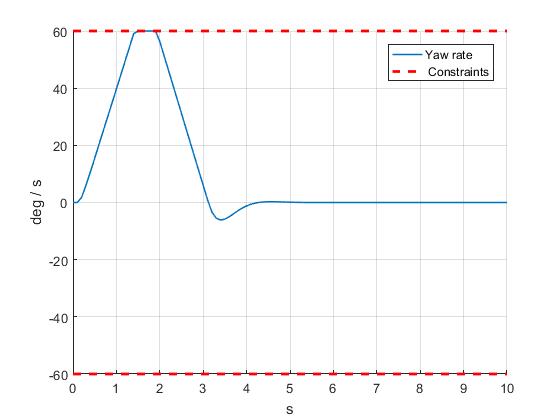
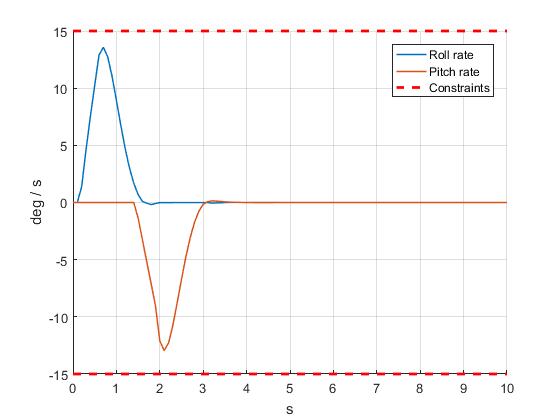
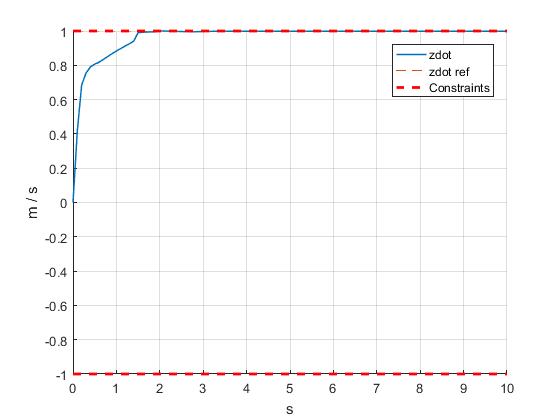
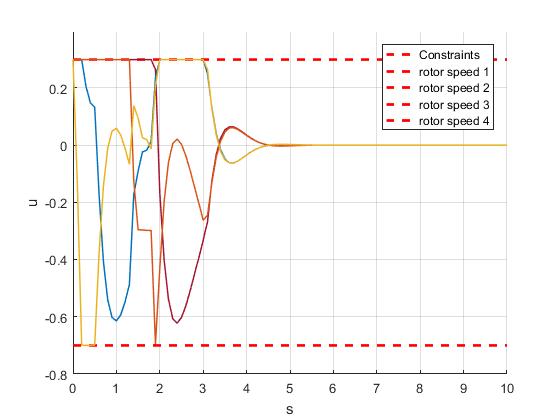
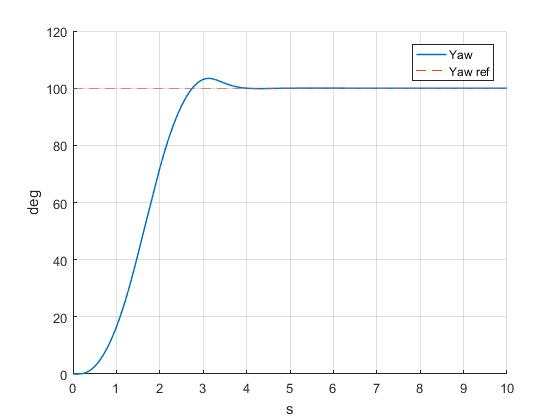
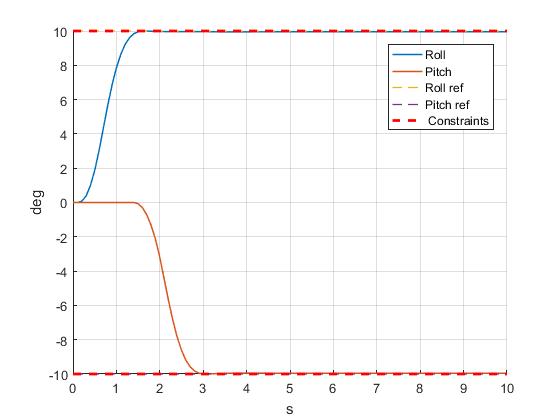
4.

By definition:

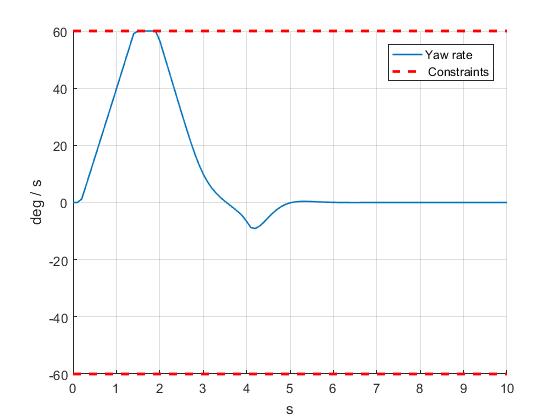
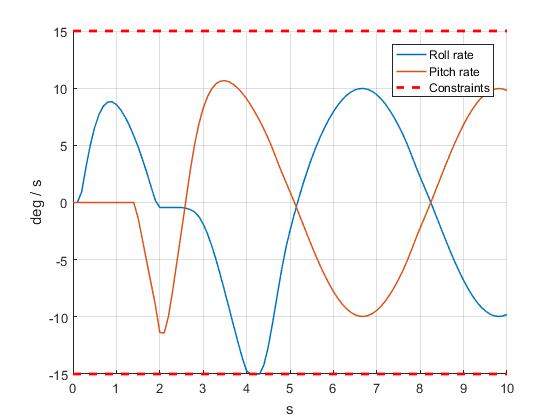
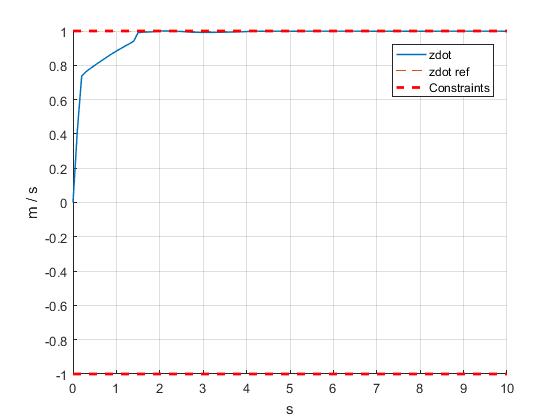
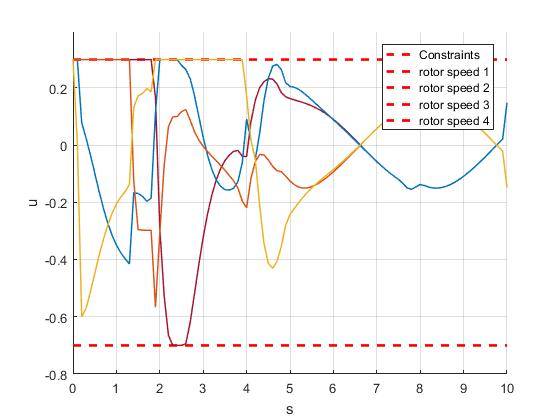
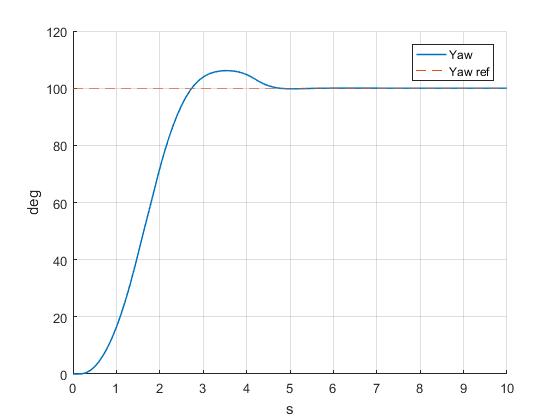
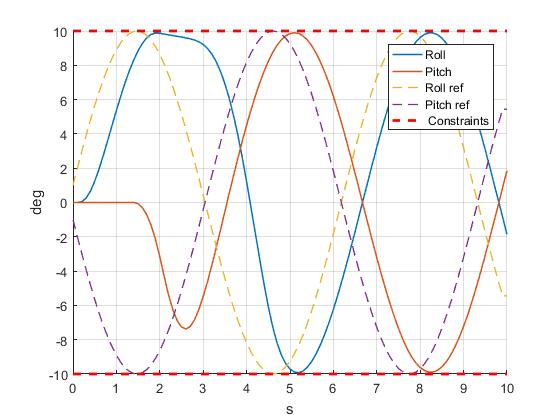
And since C is identity for the first 4 states and 0 for the rest:

The steady state for input is 0 since we have no reference there.

5. Constant reference:

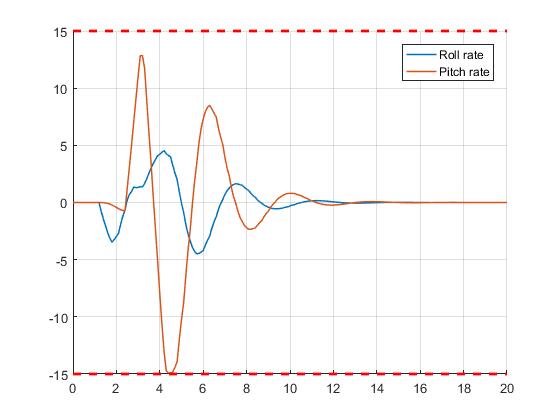
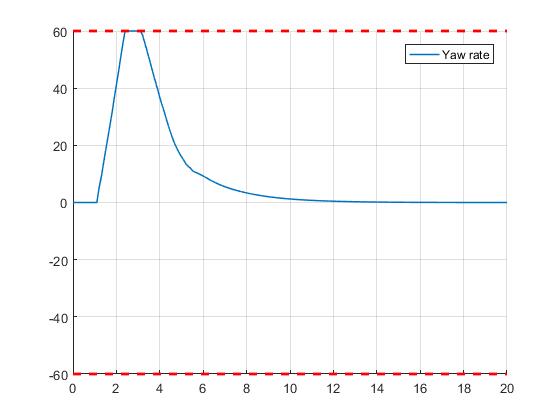
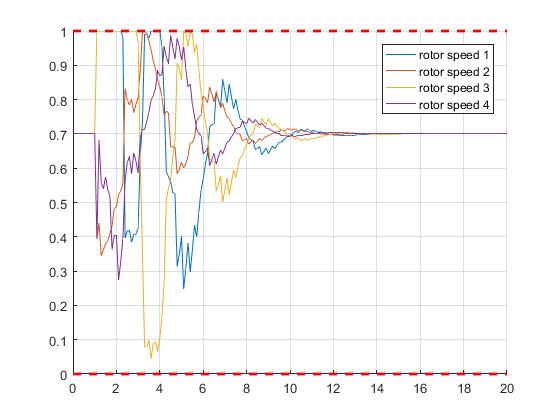
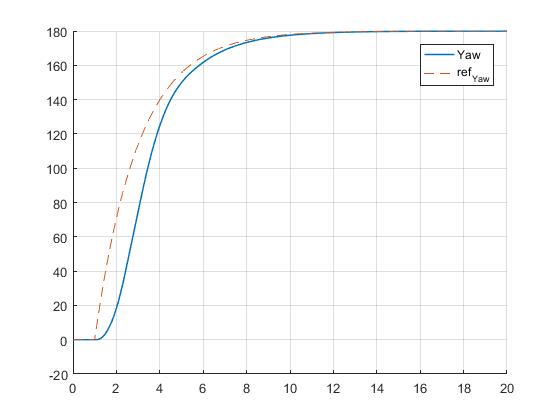
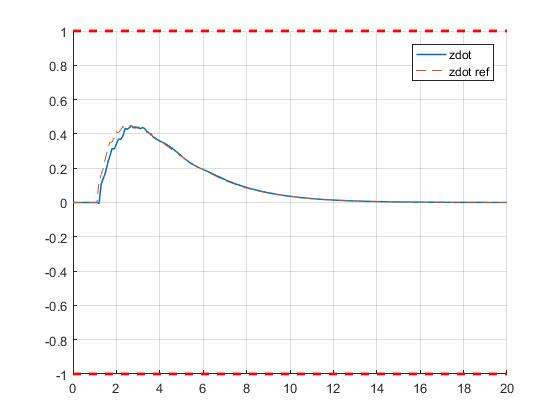
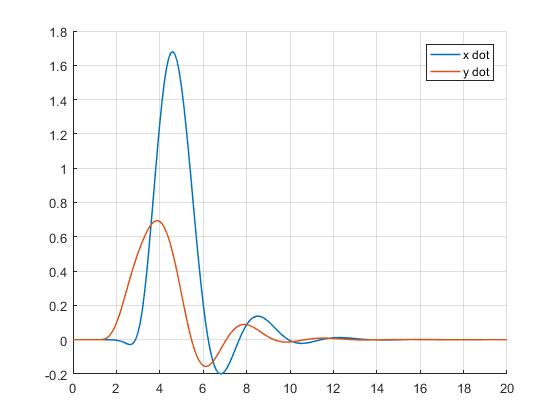
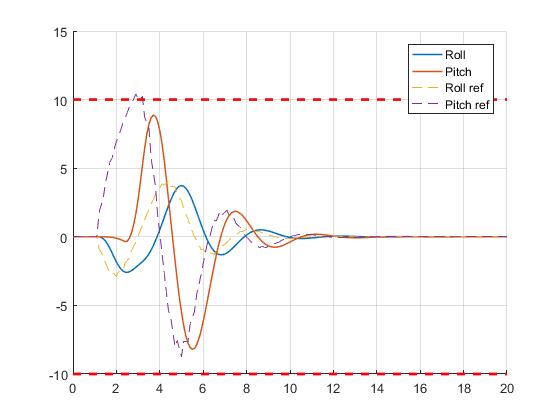
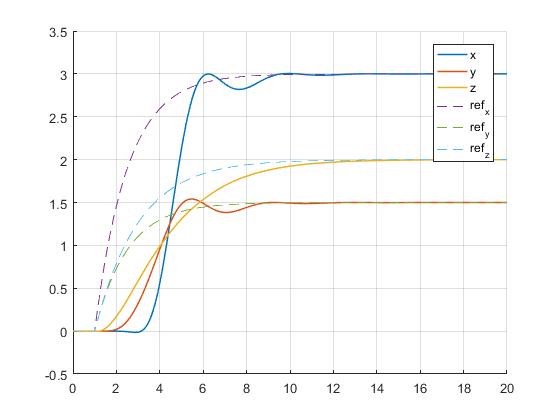
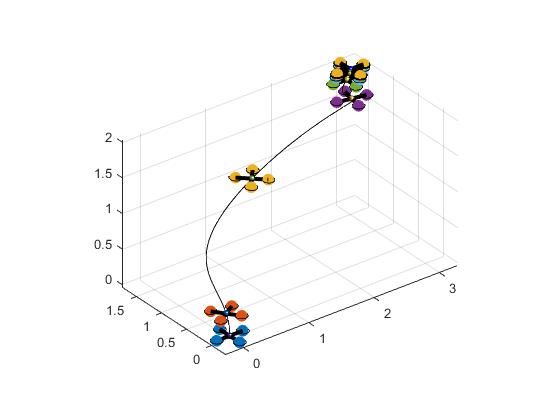


6. Varying reference:



## First Simulation of Nonlinear Model

7. Nonlinear Response graphs:



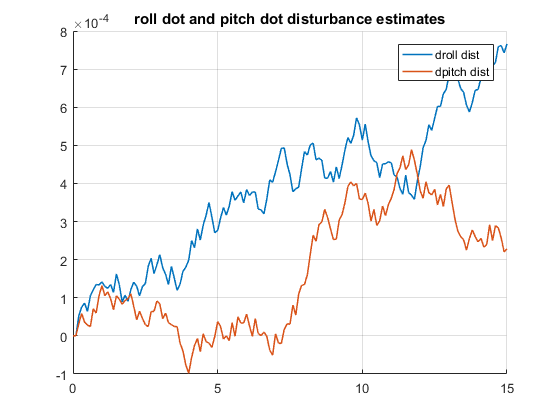
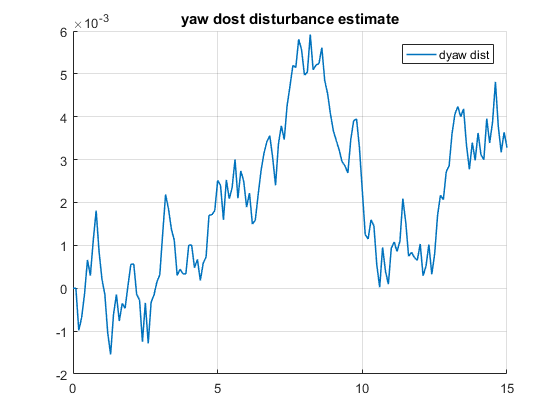
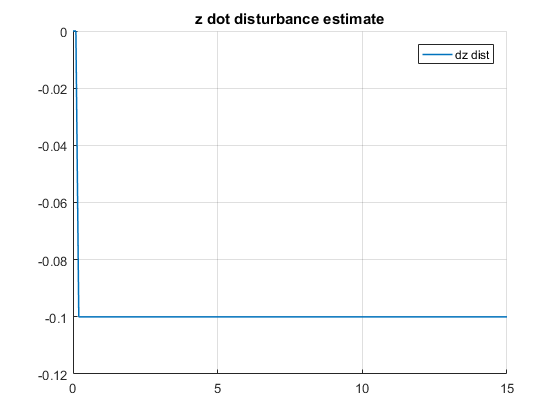
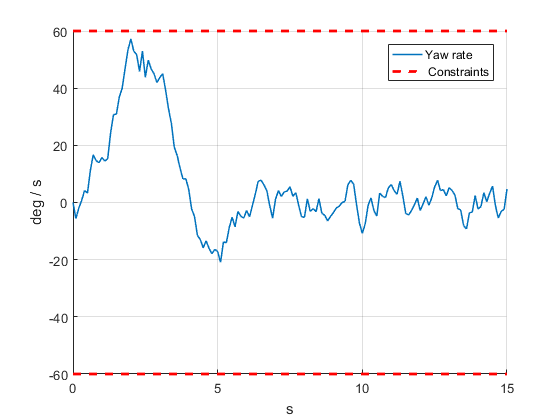
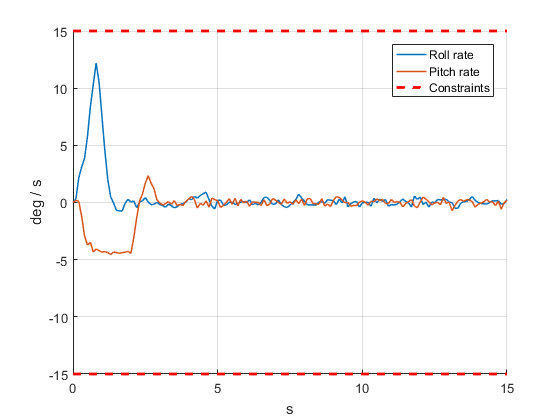
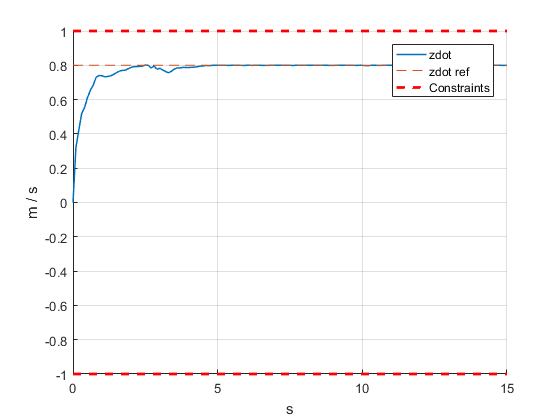
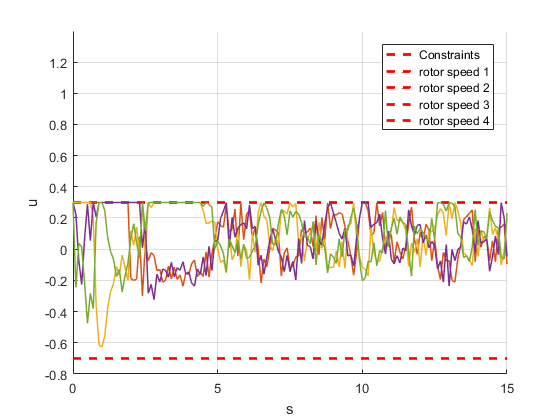
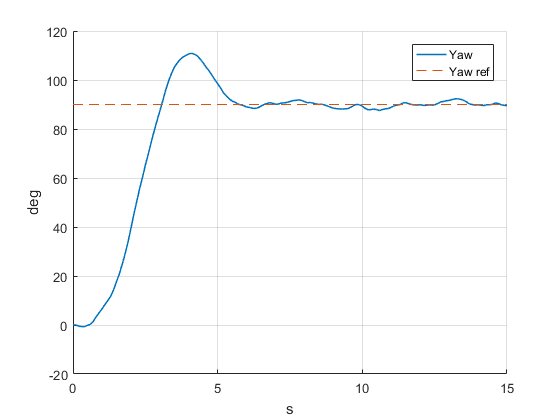
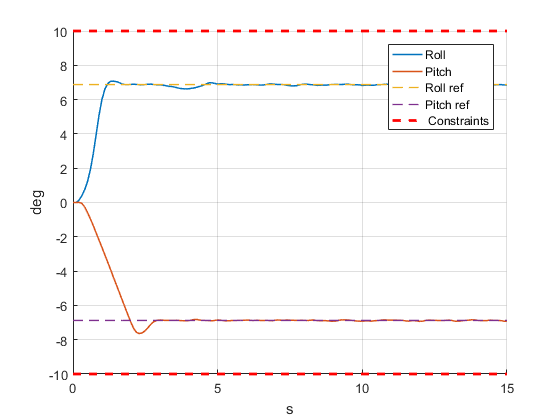
## Offset free MPC

8. The L-matrix is split in to two parts, Lx and Ld structured as:

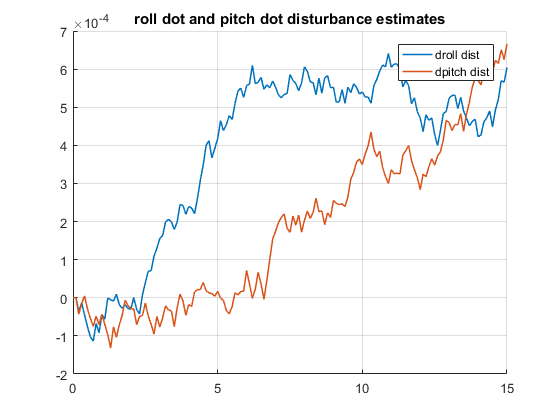
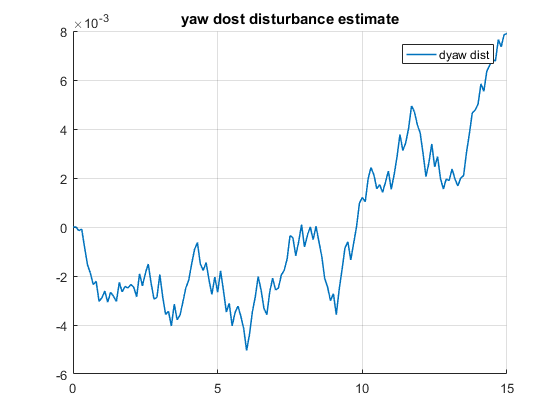
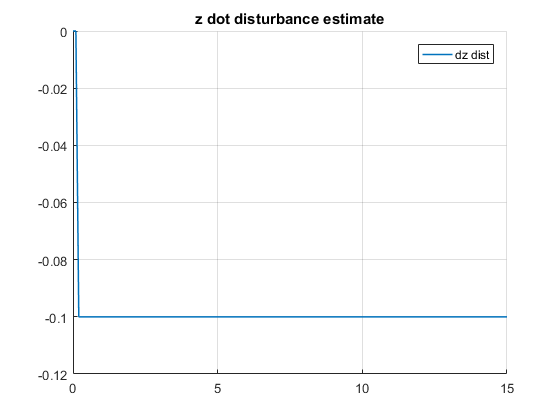
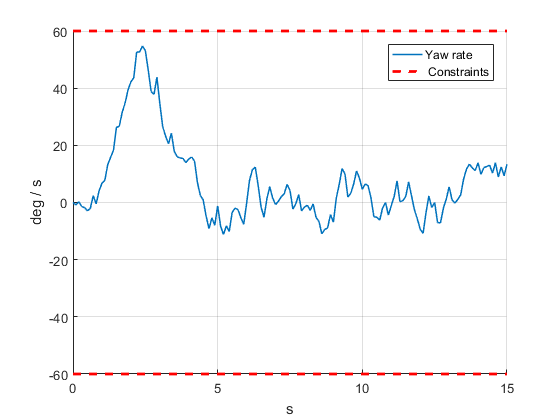
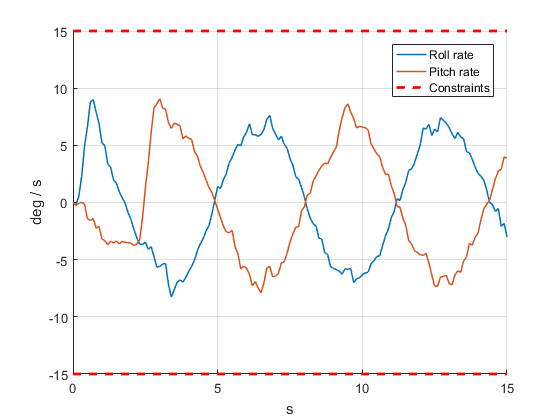
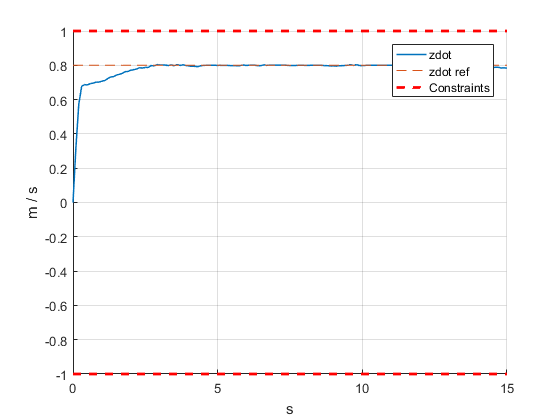
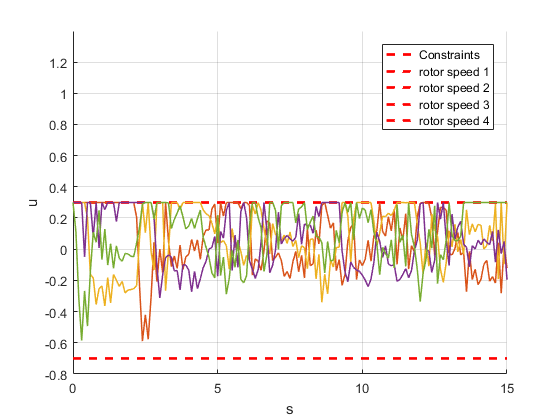
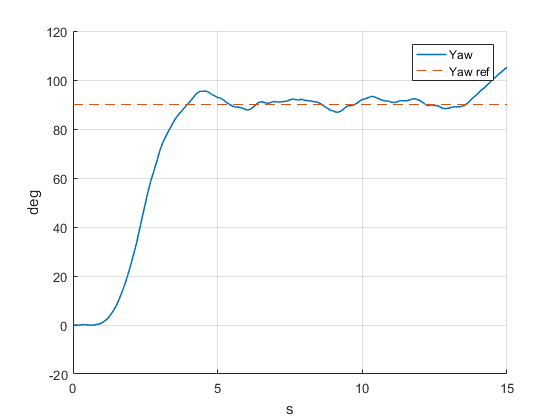
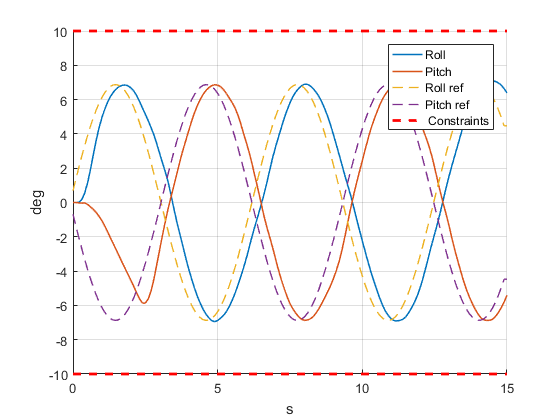
Where Lx is corresponding to the estimation of the states and Ld corresponds to the estimation of the disturbance. Lx is chosen to be the identity matrix due to the fact that we are assuming that the states are measured. Ld, on the other hand, have been chosen to be:

This matrix has been chosen in this way to acquire a stable estimator that also gives acceptable performance during linear and nonlinear simulations. To check that that the estimator is stable we can calculate the eigenvalues of the matrix Af and see that the absolute value eigenvalues are smaller than one, which our eigenvalues are.

9. Plots of the response for the constant reference signal:

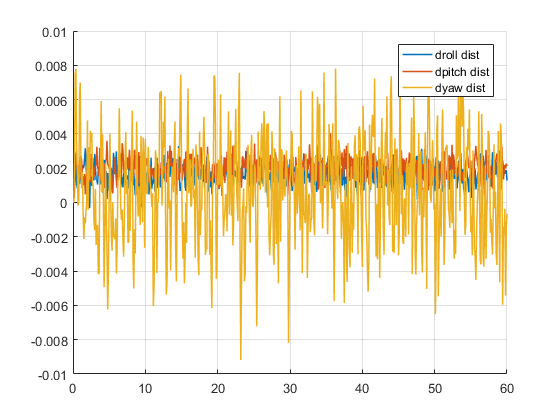
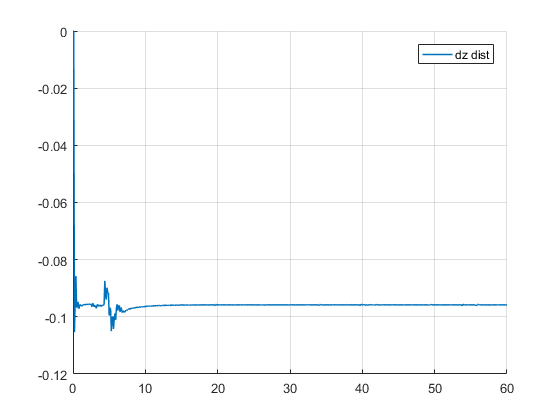
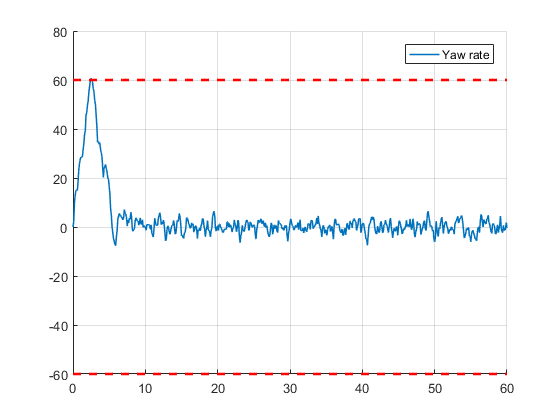
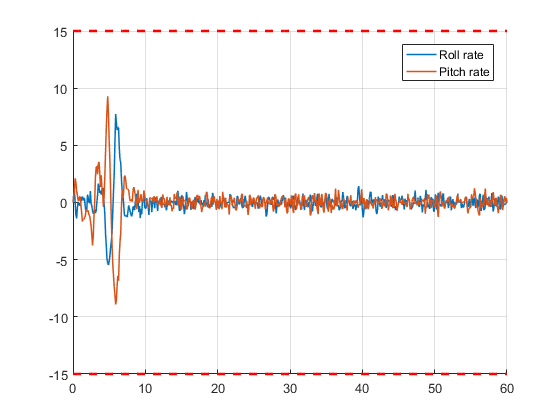
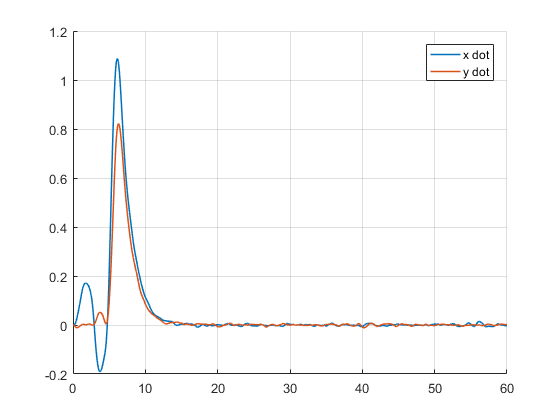
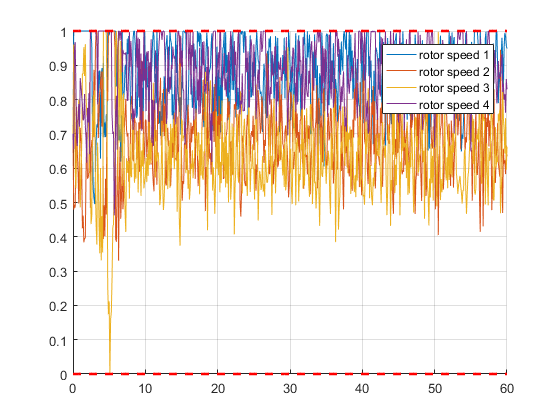
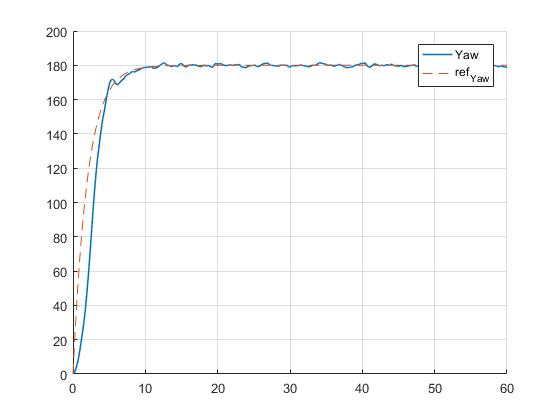
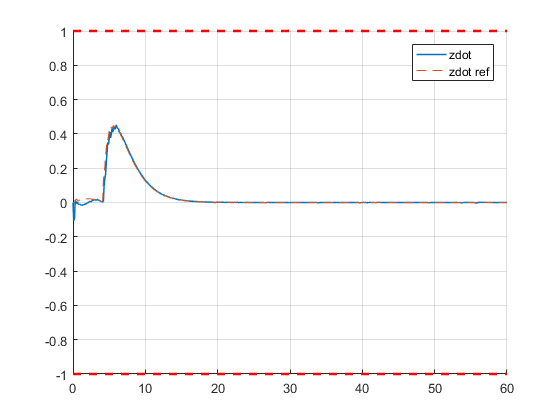
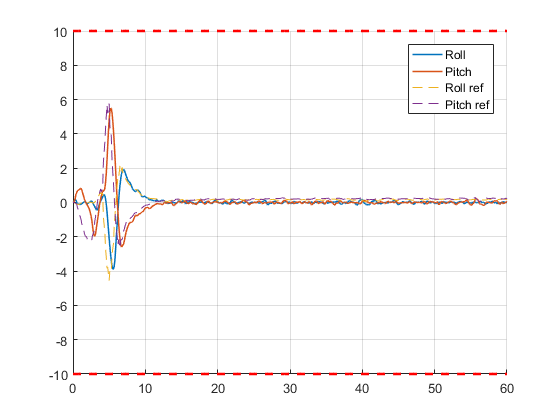
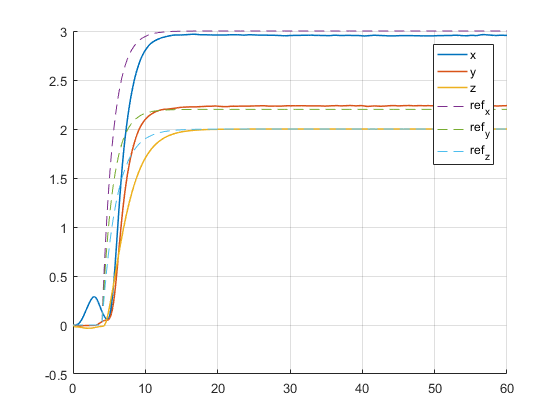
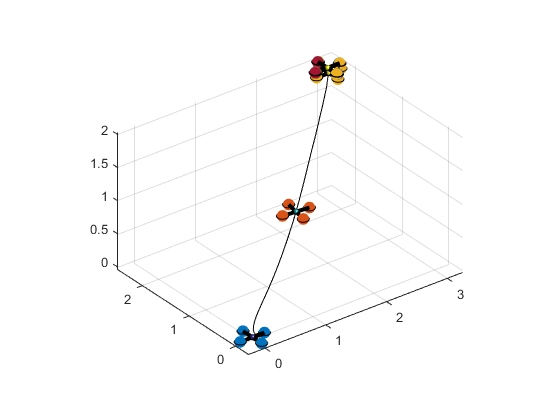


10. Plot of the response for the slowly varying reference signal:

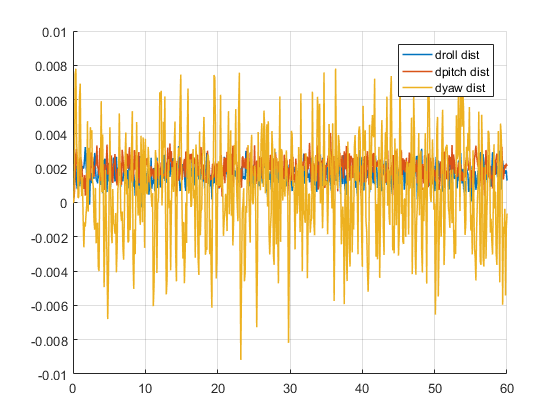
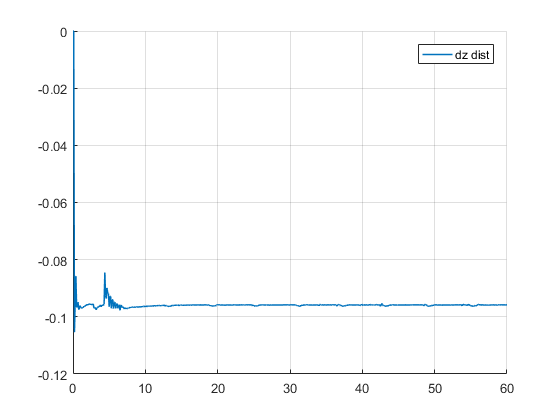
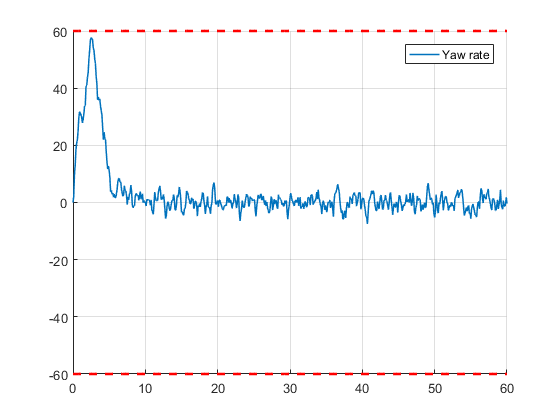
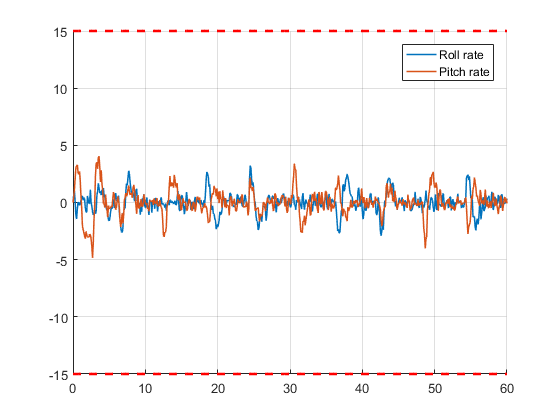
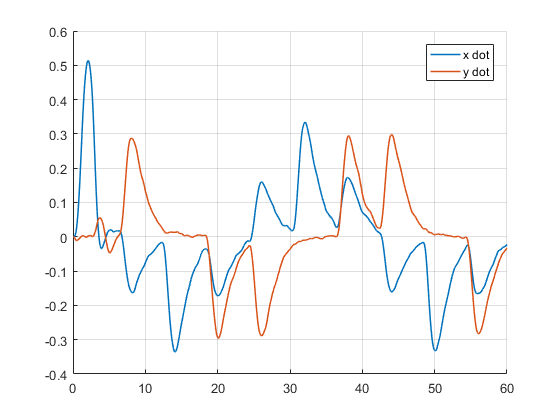
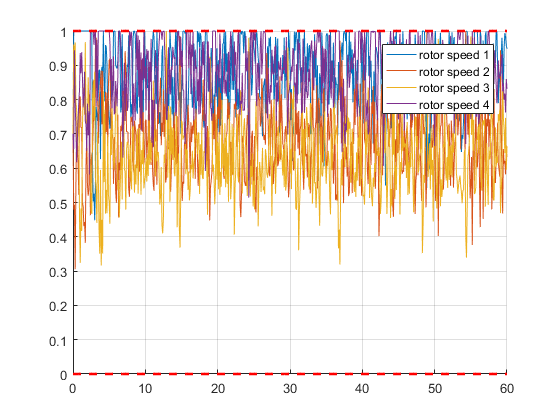
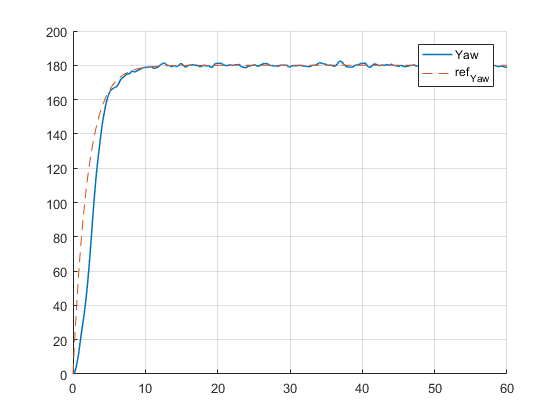
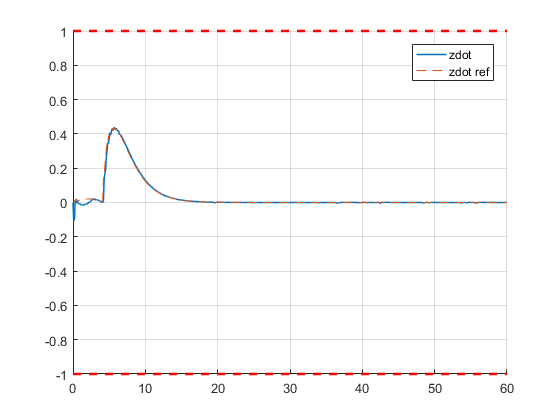
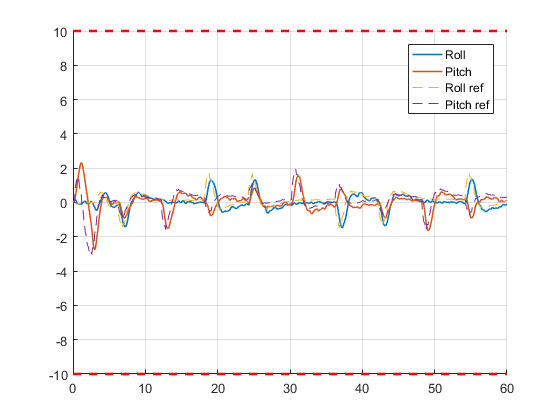
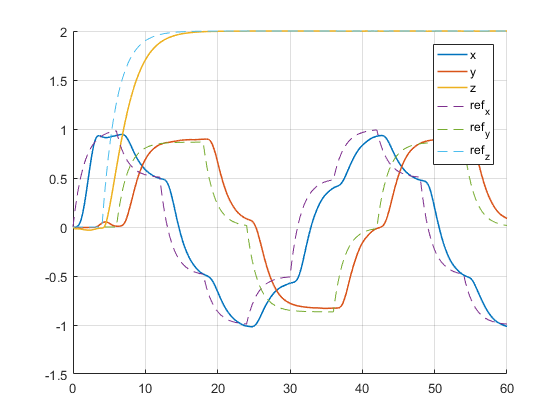
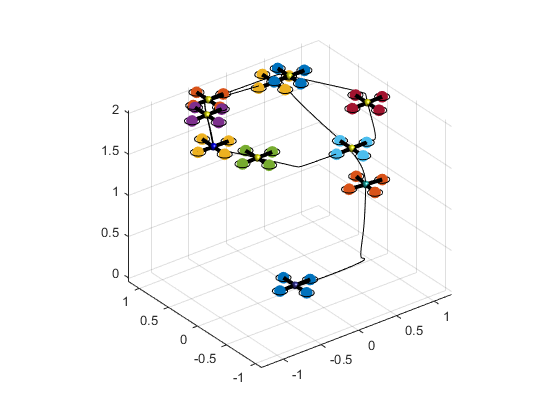


## Simulations on the Nonlinear model

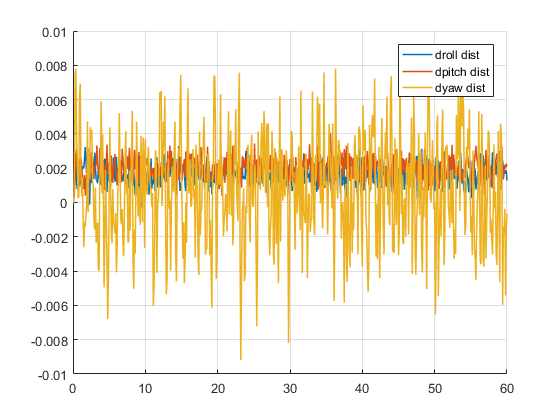
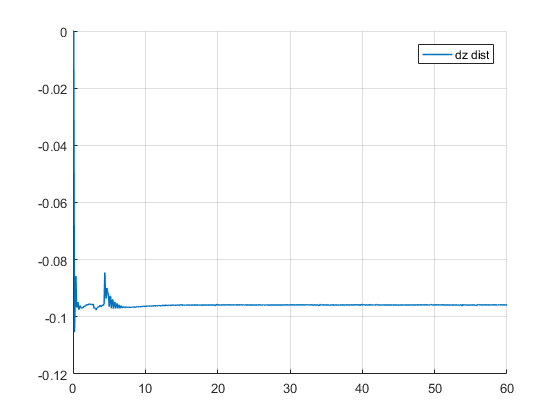
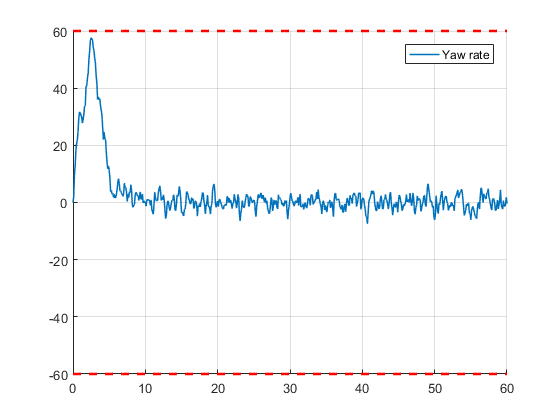
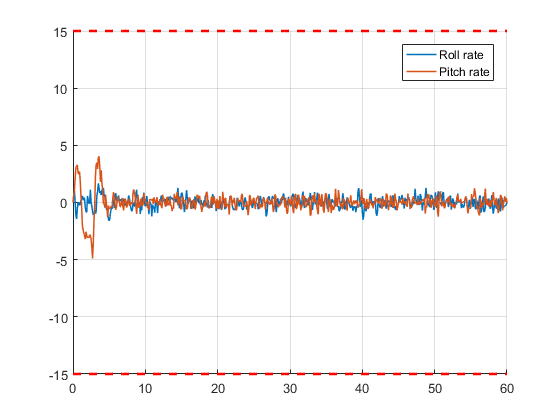
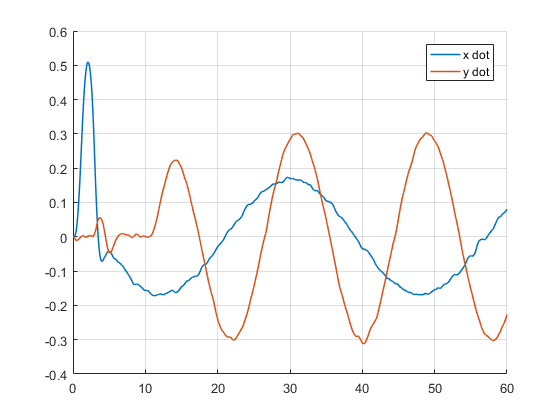
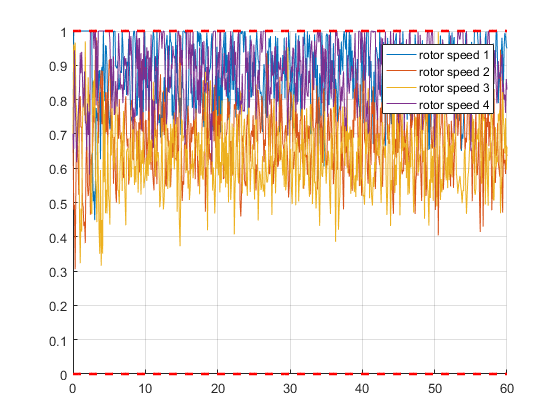
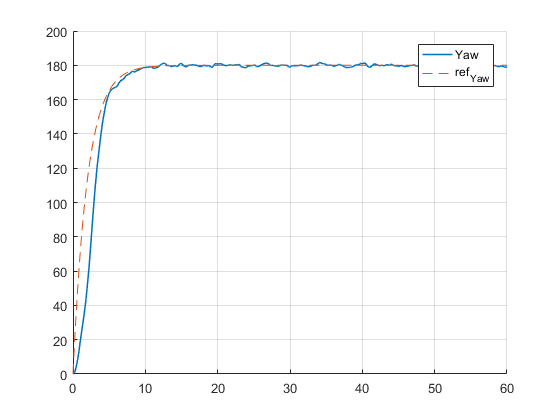
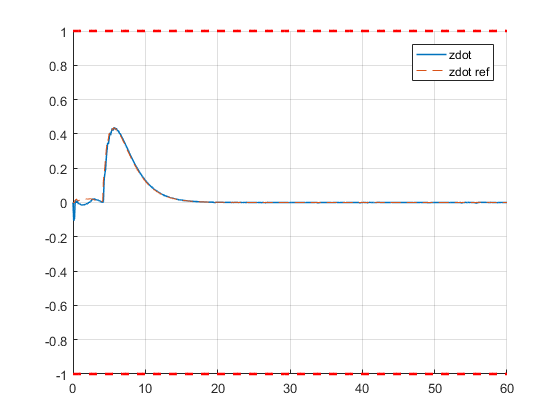
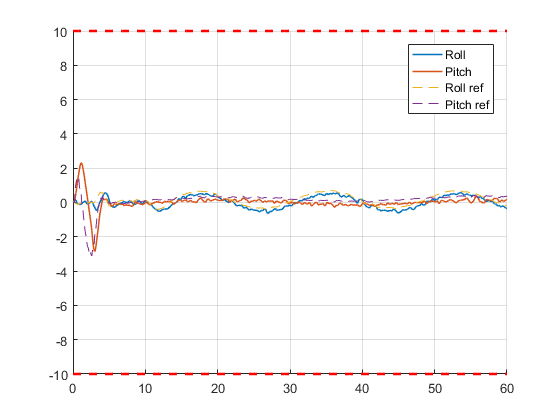
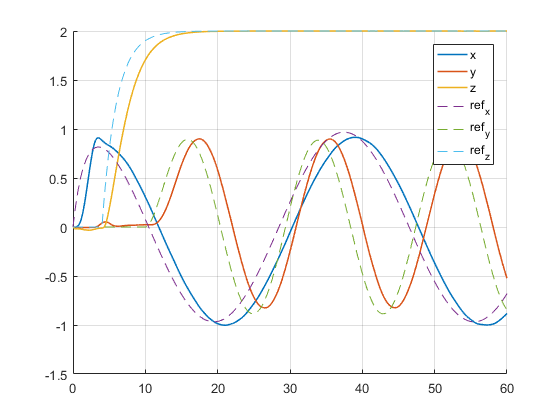
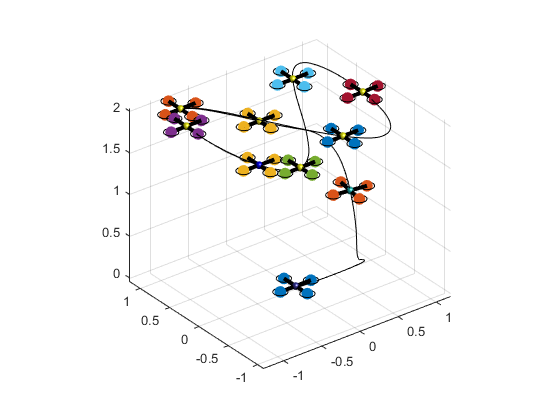
11. Plots of the reference tracking of a step signal:



12. Plots of the reference tracking of the hexagon signal:



13. Plots of the reference tracking of an “eight”:



## Slew Rate Constraints

14. Step Response:

The lowest delta value that we could achieve without causing infeasibility is 0.3.

## Soft Constraints

15. Soft Constraints Cost Function:

The reason that we chose these weight is a result of iterative design. We wanted the controller to slightly exceed the slew constraints when needed but penalise it enough such that it does not violate the constraints too often.

16. Step response:





## Forces Pro

17.

Part 5. Reference tracking using N = 20:

ForcesPro: rt = 0.0391 = 39 ms

QuadProg: rt = 0.0422 = 42 ms

Part 9. Disturbance using N = 20:

ForcesPro: rt = 0.0533 = 53ms

QuadProg: rt = 0.0607 = 61ms

As expected, the ForcesPRO time is faster since it is a dedicated C code optimized for control purposes.

It is worthy to note that the ForcePRO servers were unable to compute the controller with N = 100, which is possible for quadprog. While this is probably implementation issue with the servers of EmboTech, it is still a downside to the program.