MPC Programming Exercise Report

Students: Pat Karnchanachari, Pontus Grahn, Valentin Yuryev.

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.

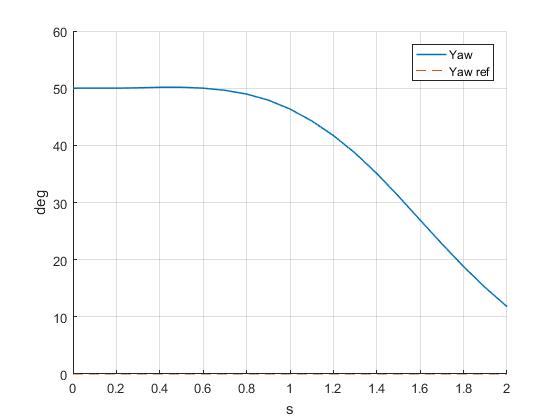
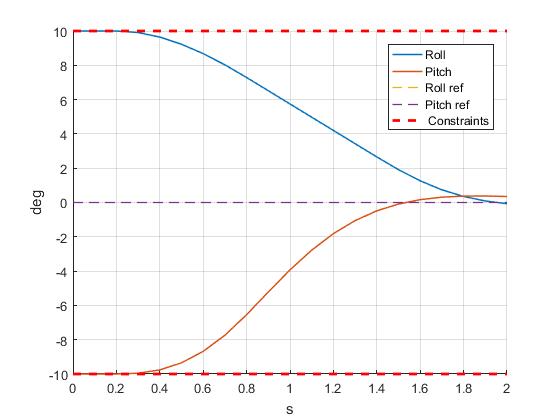
A – Since we are using subsystem with only the last 7 states, our A matrix consists of the 7x7 bottom right corner of the full A matrix.

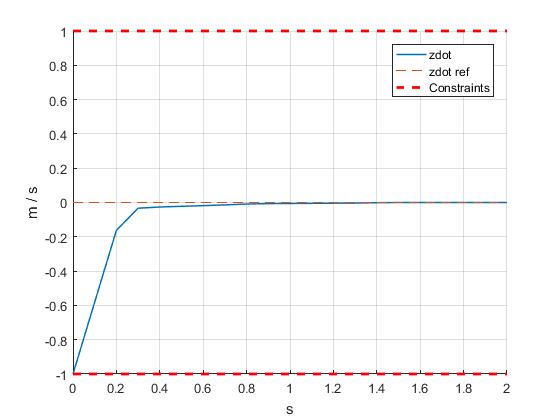
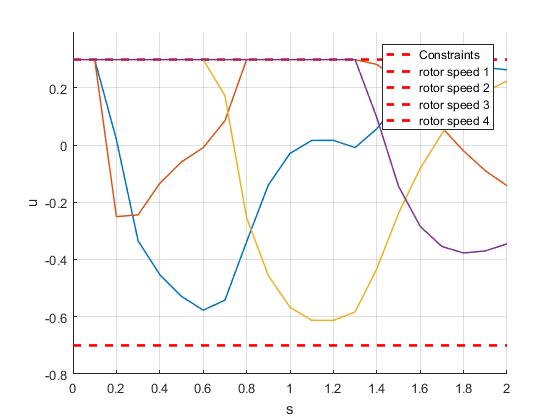
B – Similarly to A, we only want to use the portion of b matrix that corresponds to the last 7 states.

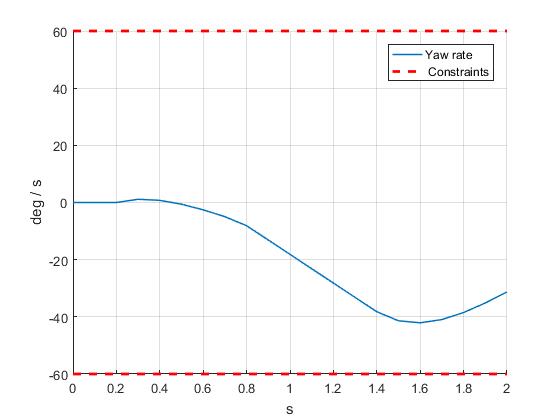
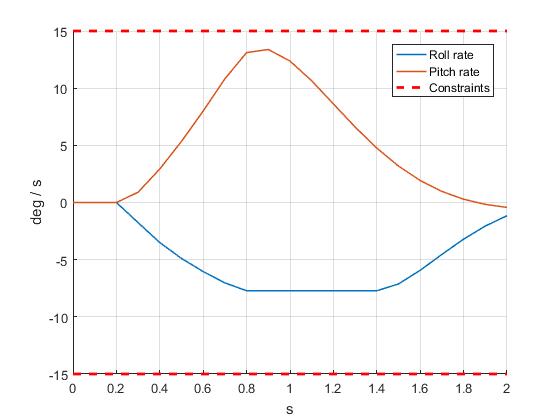
N – We set N to 19 steps which roughly corresponds to 2 seconds. The reason for this is that we want the comptroller to plan to be within the terminal set within 2 seconds.

3.

Plots of response starting from given initial condition:







4.

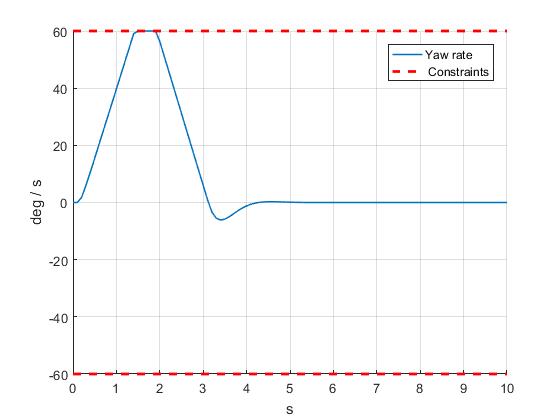
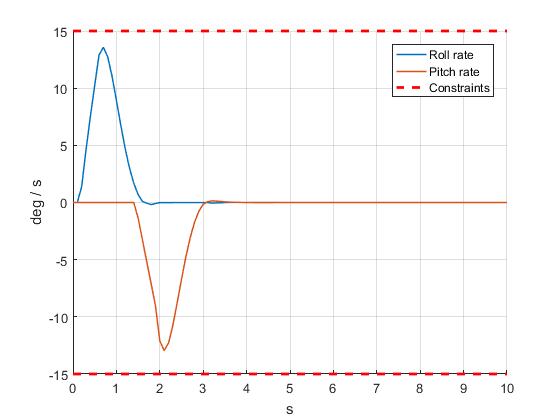
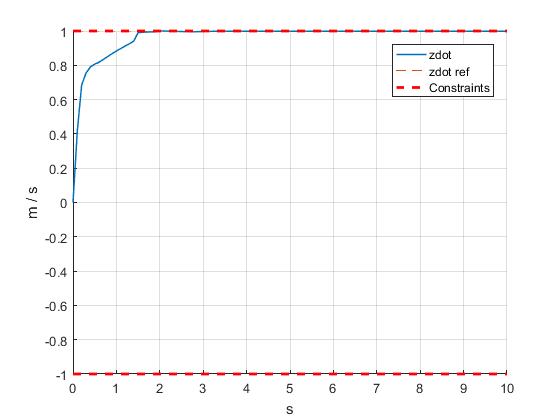
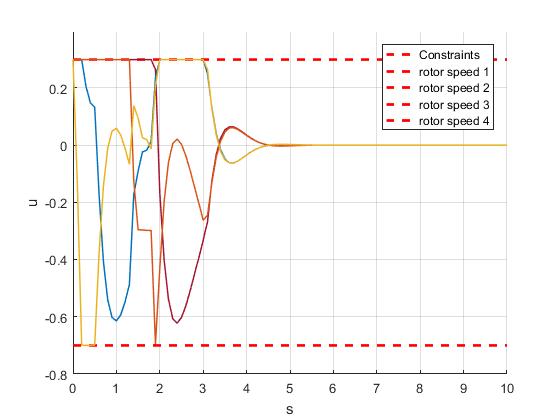
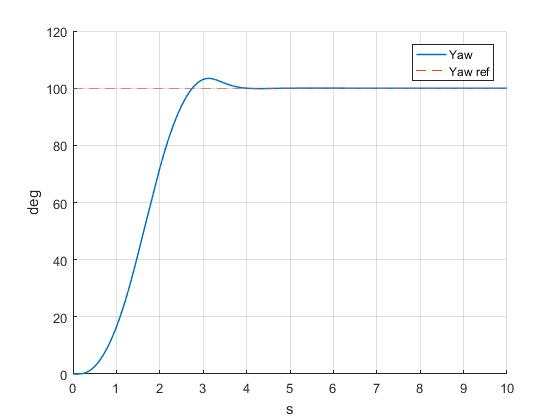
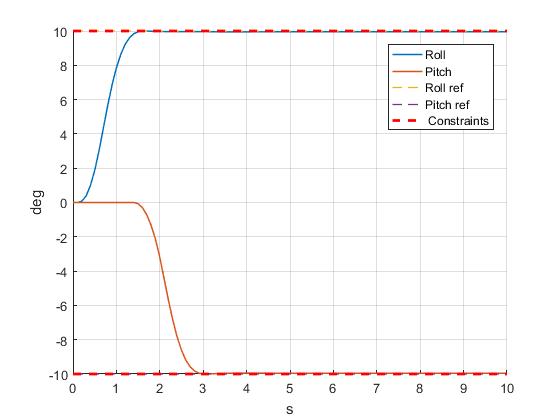
By definition:

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.

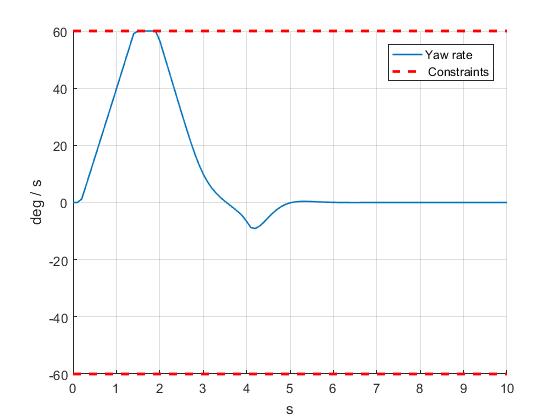
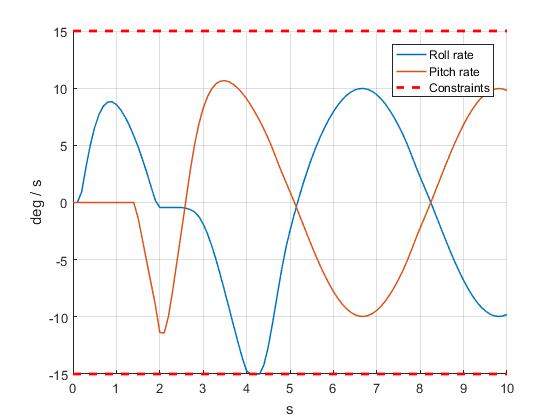
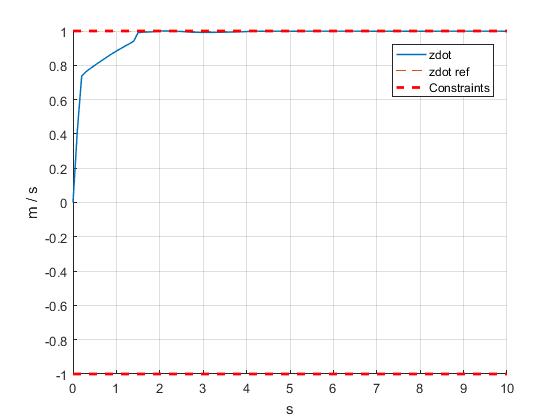
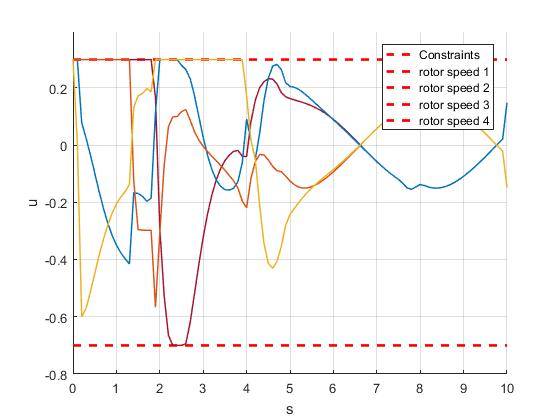
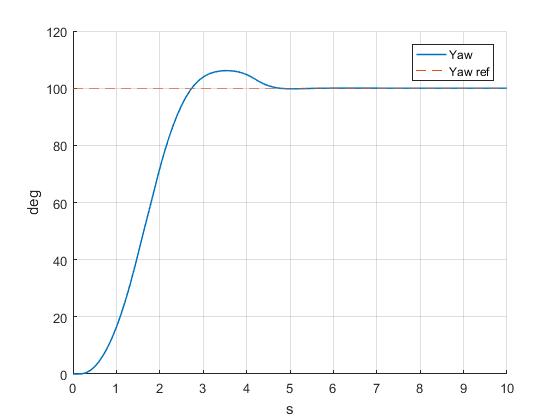
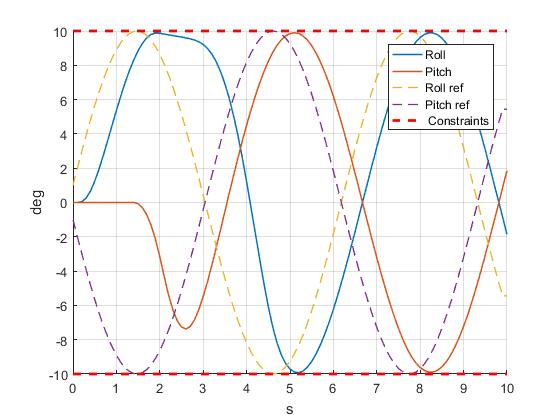
5.

Constant reference:



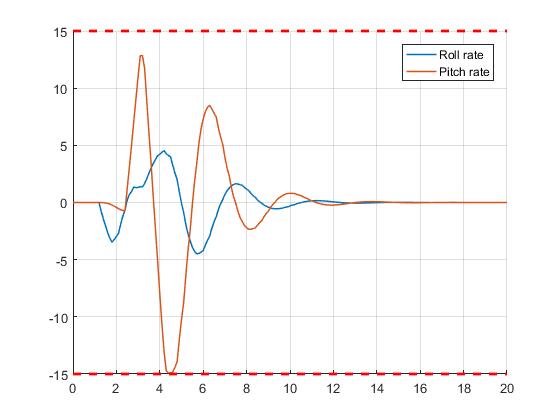
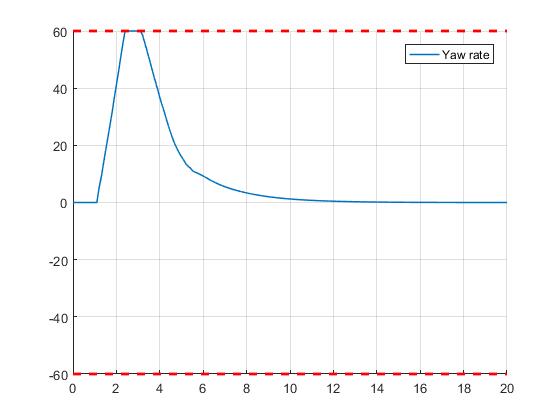
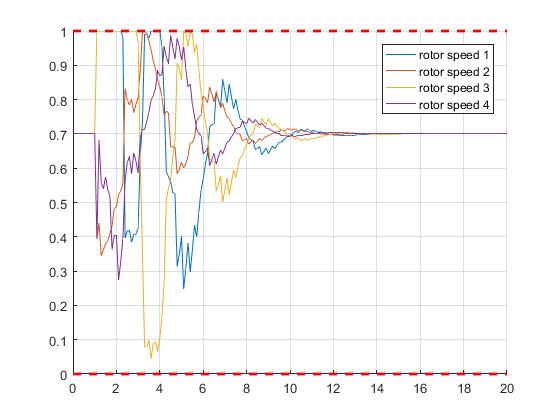
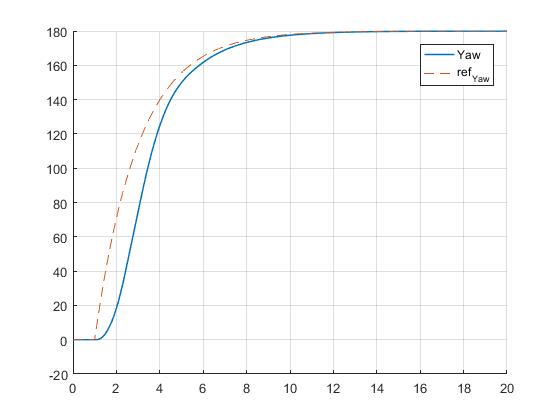
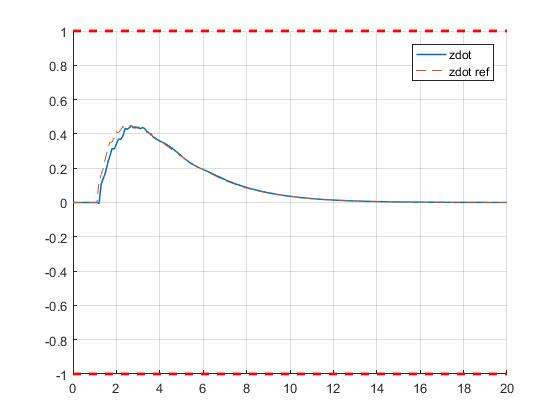
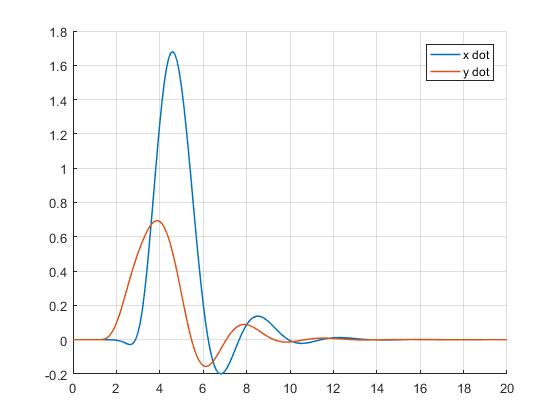
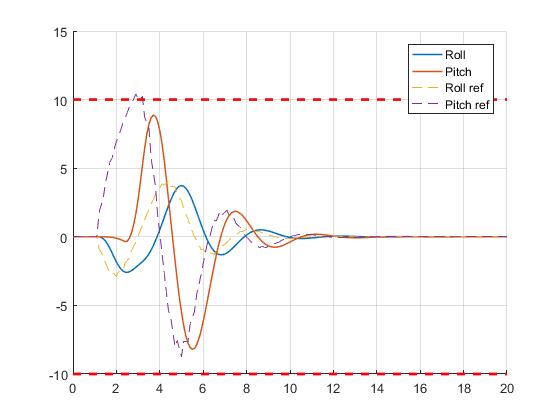
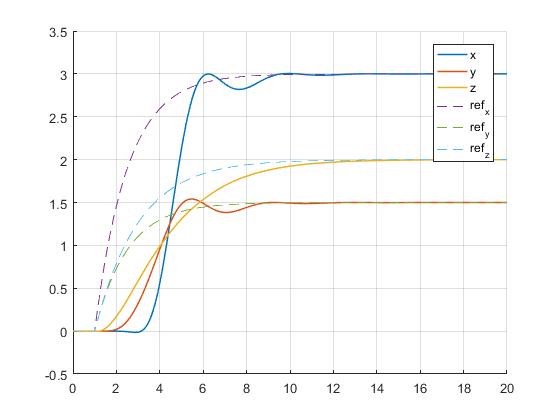
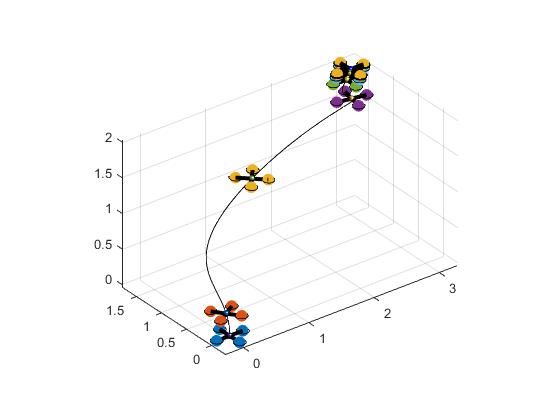
6.

Varying reference:



7.

Nonlinear Response graphs:



12.

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