A Linear Model Predictive Controller for a Quadrotor

Astrid Lindstedt astridli@kth.se
Muhammad Zahid mzmi@kth.se
Mustafa Al-Janabi musaj@kth.se
Matea Španić spanic@kth.se

HT20

Quadrotor Dynamics

Question 1, Linearization

Linearize the system around the hovering equilibrium point, that is, $\theta \simeq 0, \phi \simeq 0$, fz = mg. The yaw angle you can choose either to linearize at $\psi \simeq 0$ or as a parameter that you can change depending on the state. Choose one and motivate it.

The system is given as

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{R_B}(\alpha)\frac{\mathbf{f_t}}{m} + \mathbf{v} \\ \dot{\alpha} &= \mathbf{T}(\alpha)\omega \\ \dot{\omega} &= \mathbf{M}^{-1}(\tau - \omega \times \mathbf{M}\omega) \end{aligned} \tag{1}$$

where the states are

$$\mathbf{x} = \begin{bmatrix} p_x & p_y & p_z & v_x & v_y & v_z & \theta & \phi & \psi & \omega_x & \omega_y & \omega_z \end{bmatrix}^T. \tag{2}$$

and the input

$$\mathbf{u} = \begin{bmatrix} f_z & \tau_x & \tau_y & \tau_z \end{bmatrix}. \tag{3}$$

From linearization of Equation (1) one could write the system as

$$\dot{x} = Ax + Bu \tag{4}$$

with $A = \frac{\partial f}{\partial x}$ and $B = \frac{\partial f}{\partial u}$ it resulted in

$$A = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & gsin(\psi) & gcos(\psi) & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -gcos(\psi) & gsin(\psi) & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega_z & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\omega_z & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \omega_y & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-\omega_z M_z + \omega_z M_y}{M_x} & \frac{-\omega_y M_z + \omega_y M_y}{M_x} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\omega_z M_z - \omega_z M_x}{M_y} & 0 & \frac{\omega_x M_z - \omega_x M_x}{M_y} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-\omega_y M_y + \omega_y M_x}{M_z} & \frac{-\omega_x M_y + \omega_x M_x}{M_z} & 0
\end{bmatrix} \tag{5}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{M_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{M_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{M_z} \end{bmatrix}^T . \tag{6}$$

In the model we set ψ as a parameter that we can change depending on the state.

Question 2: Discretize and check linearization result.

Discretize the system and control if it is correct.

To ensure that our implements of the linear model are correct, trivial control signals are used. At first, the control signal $u = [0,0,0,0]^T$ is tested. Figure 1 shows that the system stays at the initial state. Our second control signal is $u = [0.1,0,0,0]^T$, the result is an increase of the coordinate z, seen in Figure 2. This means that the quadrotor goes up. A control signal of $u = [0.1,-0.001,0,0]^T$ makes the quadrotor move up and to the positive y axis as seen in Figure 3. The last control input is $u = [0.1,0,0.001,0]^T$. This control input, in Figure 4, moves it up and to the positive x-axis. These results ensures that our model is correctly implemented.

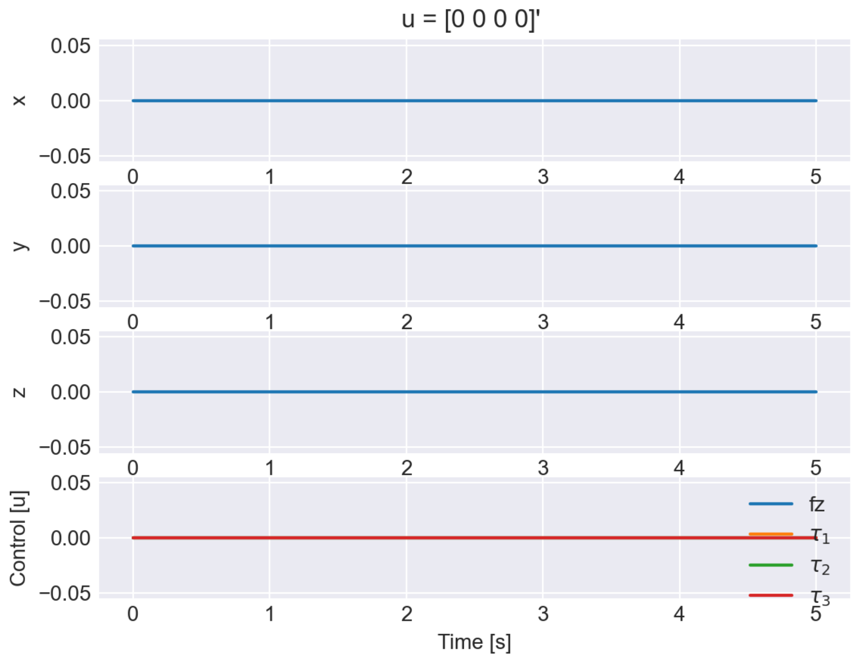


Figure 1: Stays at initial state.

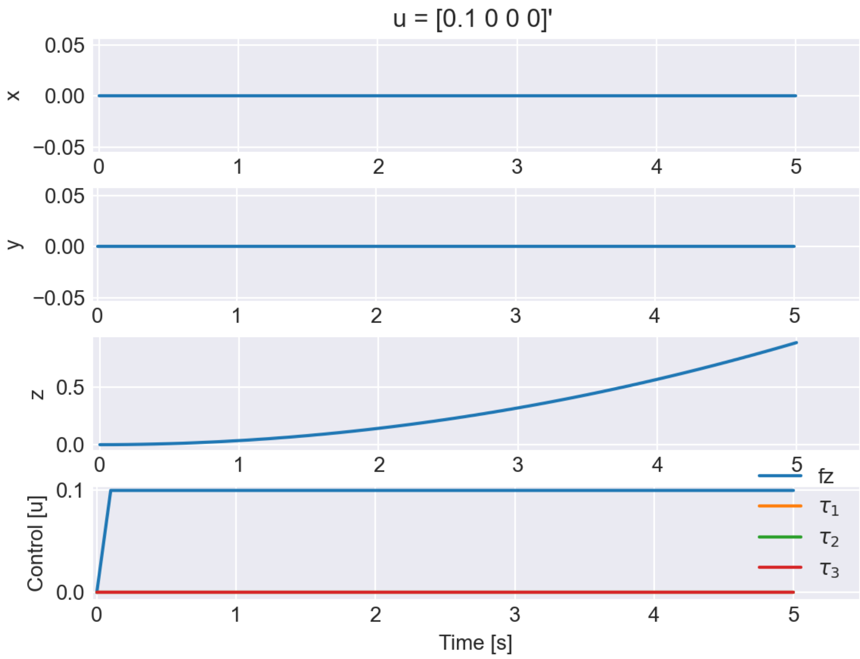


Figure 2: Goes up.

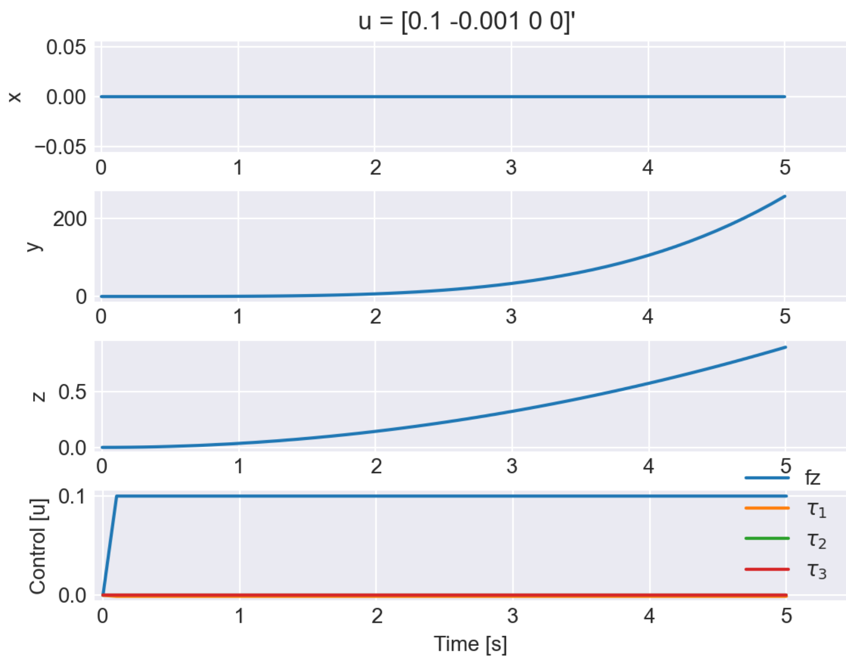


Figure 3: Moves to positive y direction meanwhile going up.

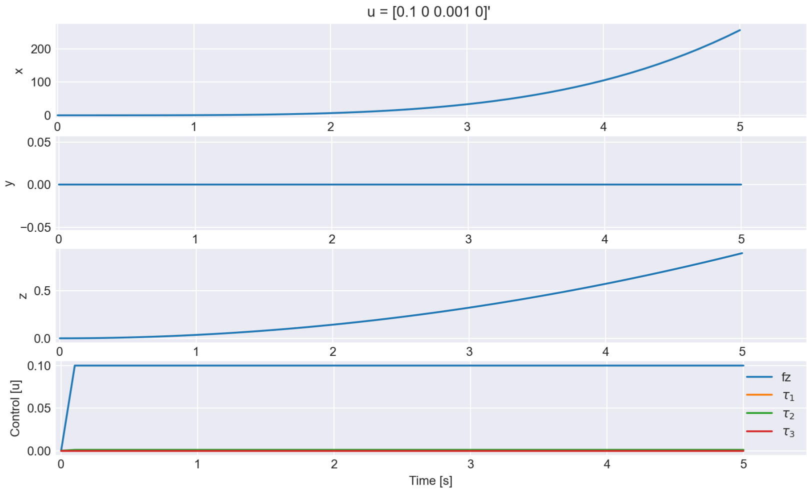


Figure 4

MPC for a Quadrotor

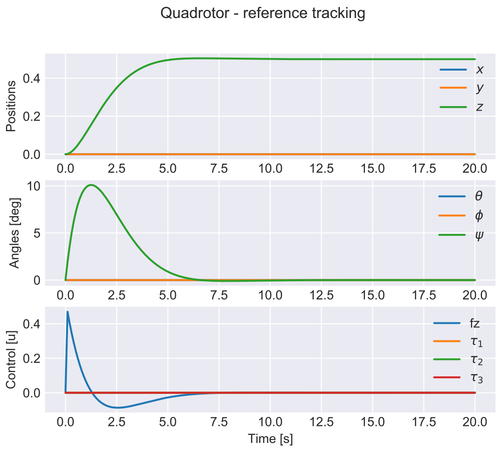
Question 3: Runnig cost

Make sure that the running cost and terminal cost functions, given a setpoint, calculate the cost given the following error. Create your MPC controller and decide your weight matrixes R and Q in a good manner. Test following setpoints:

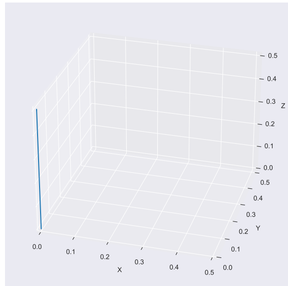
1. Move 0.5[m] in the Z direction;
2. Move 0.5[m] in the Y and Z direction;
3. Move 0.5[m] in the X,Y and Z direction.

look at how the nonlinear dynamics of the pendulum on cart example are simulated using this function and adjust it to the new dynamics. Which problems do you find? Save the system responses.

When creating the MPC controller, the weight matrices were set as $Q = I_{12}$, $R = I_4$ and $P = 1$. Figure 5, 6 and 7 are demonstrating the simulations of different set point using the linear model. As the response looks quite fast and there are no overshoots, no tuning procedure is performed on the weight matrices. The simulations of the nonlinear model in Figure 8, 9 and 10 shows a slightly slower response. Another thing that differs from the linear results compared to the nonlinear results is the input u. For the nonlinear model, u is around 14. This is the input that is required to keep the quadrotor in the air, $m \cdot g \approx 14$. In the linear model we are linearizing the system around $u \approx 14$, hence the input in the plots go to zero.

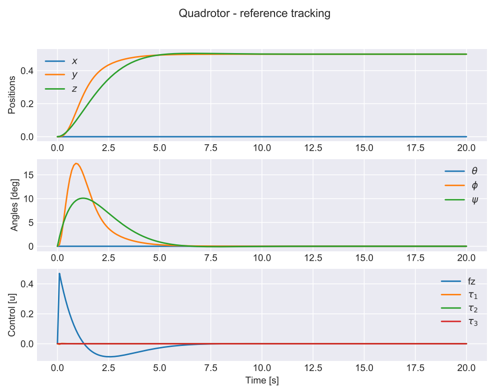


(a)

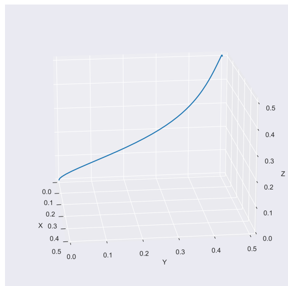


(b)

Figure 5: Setpoint 0.5[m] in Z direction, linear model.

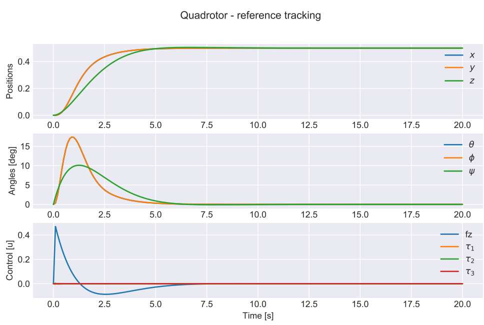


(a)

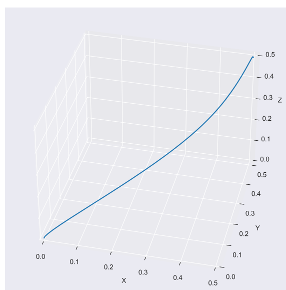


(b)

Figure 6: Setpoint 0.5[m] in Y and Z direction, linear model.



(a)

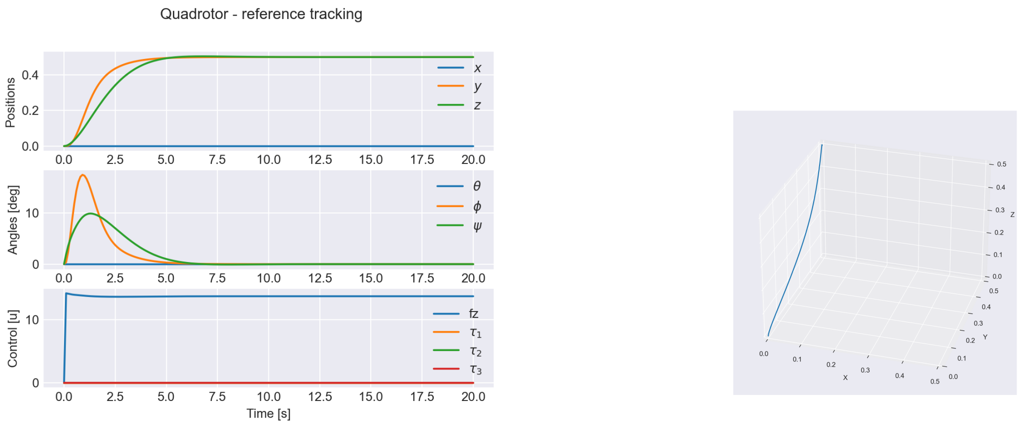


(b)

Figure 7: Setpoint 0.5[m] in X, Y and Z direction, linear model.

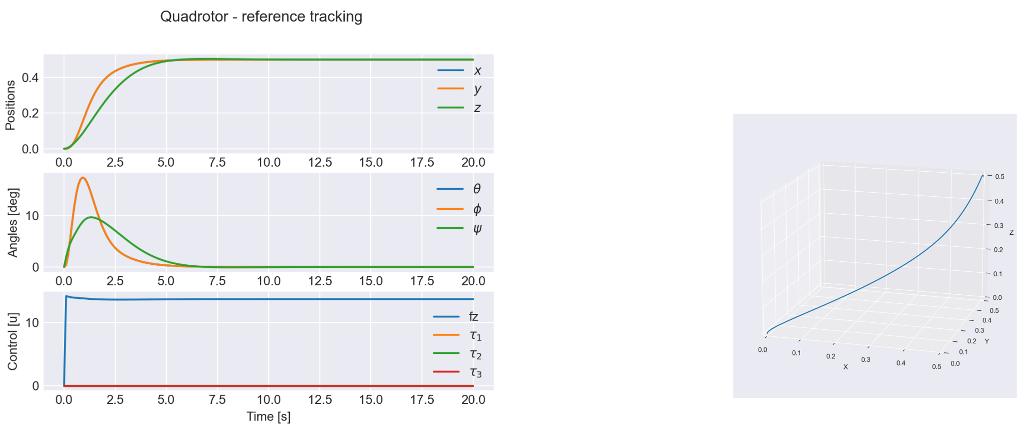
(a) (b)

Figure 8: Setpoint 0.5[m] in Z direction, nonlinear model.



(a) (b)

Figure 9: Setpoint 0.5[m] in Y and Z direction, nonlinear model.



(a) (b)

Figure 10: Setpoint 0.5[m] in X, Y and Z direction, nonlinear model.

Integrator Dynamics

Question 4

Based on the lecture notes, implement the integrator dynamics that allow our system to cope with the these differences. Then adjust the simulated model (nonlinear dynamics) to simulate3 the difference in mass with respect to the controller and simulate the system.Finally, to cope with center of mass displacement in our vehicle, integrator dynamics for the lateral movement are useful.

For the system to be able to correct steady state errors and handle modelling errors we added integrator dynamics. This was done by augmenting the state space like in assignment 3 and we did the augmented system such that it can handle lateral motions as well. The linear system was augmented with the integrator $i_{t+1} = i_t + h(r_{ref} - Cx_t)$ according to the following:

$$\begin{bmatrix} x_{t+1} \\ i_{t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A & 0 \\ -hc & \mathbf{I}_3 \end{bmatrix}}_{A_i} \begin{bmatrix} x_t \\ i_t \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0}_{3\times 4} \end{bmatrix}}_{B_i} u_t + \underbrace{\begin{bmatrix} \mathbf{0}_{12\times 4} \\ h\mathbf{I}_3 \end{bmatrix}}_{B_r} r_{ref} \tag{7}$$

with

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} \tag{8}$$

The stage cost functions was also augmented to take into account the added states. Furthermore, the new nonlinear system where augmented according to the following:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{R_B}(\alpha)\frac{\mathbf{f_t}}{m} + \mathbf{v} \\ \dot{\alpha} &= \mathbf{T}(\alpha)\omega \qquad (9) \\ \dot{\omega} &= \mathbf{M}^{-1}(\tau - \omega \times \mathbf{M}\omega) \\ \dot{\mathbf{i}} &= \mathbf{r_{ref}} - \mathbf{p} \qquad (10) \end{aligned}$$

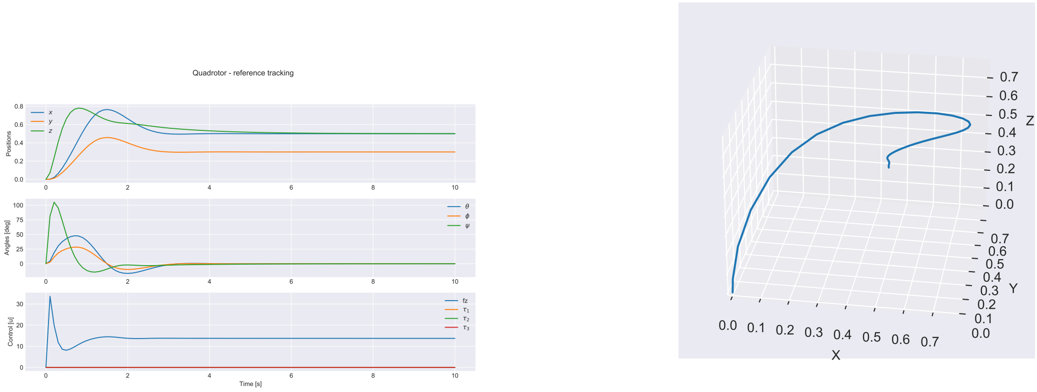
The weight matrices for the MPC controller were set so that $P = 1$,

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 500 \end{bmatrix} \tag{11}$$

and $R = I_4$. To test the robustness of our controller an offset of the model is set. If the controller can put the quadrotor to the initial set point, the controller is said to be stable and robust. Our controller takes lateral motion of the vehicle into account. The set point is set to [0.5, 0.3, 0.5] and the system is simulated with three different offsets. Figure 11 shows an offset of the mass, from 1.4 to 2. The system stabilizes at the given set points. Figure 12 is showing the simulation results for an offset of the thrust. The trust has in this case a disturbance of 10. Figure 13 is demonstrating an offset of the inertia matrix, the matrix is set to,

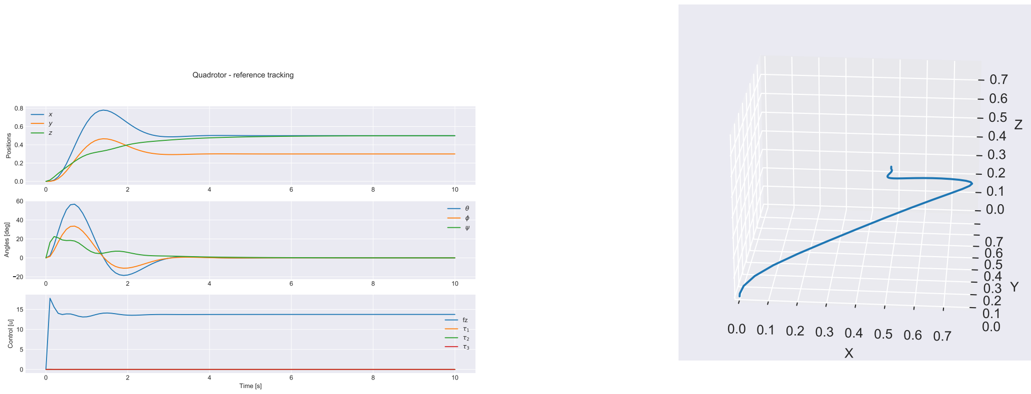
$$M = \begin{bmatrix} 0.001 & 0.004 & 0 \\ 0.004 & 0.001 & 0 \\ 0 & 0 & 0.005 \end{bmatrix} \tag{12}$$

that is off-diagonal. The results are still stable.



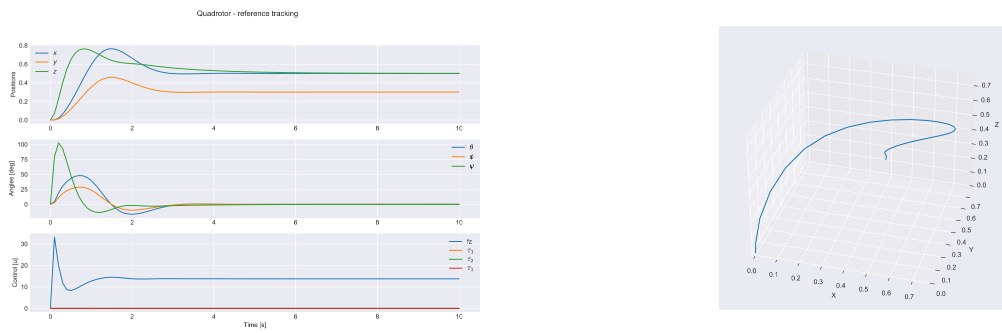
(a) (b)

Figure 11: Offset in mass, nonlinear model.



(a) (b)

Figure 12: Offset in trust, nonlinear model.



(a) (b)

Figure 13: Offset as an off-diagonal inertia matrix, nonlinear model.