Model Predictive Control
Mini-Project - Quadrotor control

Hugo Grall Lucas, Théophile Schenker
Group 4

6th June 2018

Task 1: Nonlinear model and linearization

Deliverables:

1. Interpretation of the structure of matrices Ac and Bc ; explain in particular the nonzero numbers in rows 4 and 5 of Ac and the nonzero rows of Bc in connection to the nonlinear dynamics described above. (5%)

$$A_c = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.81 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9.81 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B_c = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 3.50 & 3.50 & 3.50 & 3.50 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0.56 & 0 & -0.56 \\ -0.56 & 0 & 0.56 & 0 \\ 0.73 & -0.73 & 0.73 & -0.73 \end{pmatrix}$$

The linear approximation around (X_s, u_s) in continuous time reads as follows:

$$\dot{X} = A_c(X - X_s) + B_c(u - u_s) \quad \text{with } X_s = 0 \text{ and } u_s = \begin{pmatrix} 0.7007 \\ 0.7007 \\ 0.7007 \\ 0.7007 \end{pmatrix}$$

The non-zero values of A_c and B_c can be interpreted:

- Three first values of 1 in A_c: they simply represent the fact that $(\dot{x}, \dot{y}, \dot{z})$ is the same in X and in $\dot{X}$.
- Values of ± 9.81: they mean that $\ddot{x} = 9.81 \cdot \beta$ and $\ddot{y} = -9.81 \cdot \alpha$. This can be obtained by projecting the acceleration in the direction of z_b (that has to be 9.81 m/s^2 at equilibrium point to compensate gravity) in x and y and by applying the small-angle approximation.

- Three last values of 1 in A_c: they simply represent the fact that $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})$ is the same in X and in $\dot{X}$.
- Four values of 3.50 in B_c: They represent the acceleration in the direction of z_b produced by the four motors.
- Values of ± 0.56 in B_c: They represent the effect on $\ddot{\alpha}$ of the two motors situated on the y_b axis and similarly the effect on $\ddot{\beta}$ of the two motors situated on the x_b axis.
- Values of ± 0.73 in B_c: They represent the impact of the four motors on $\ddot{\gamma}$, the yaw angle acceleration. This is possible because two propellers move clockwise and the two other move counterclockwise.

Task 2: First MPC controller

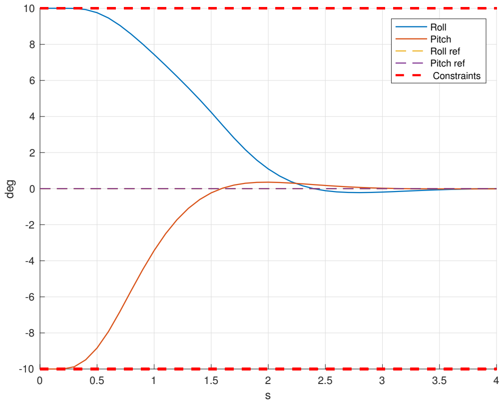
Deliverables:

1. Choice of tuning parameters and motivation for them (5%)
2. Plots of the response to an appropriate initial condition (15%

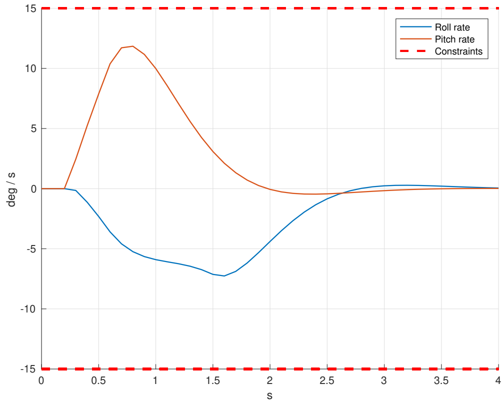
The inner controller is in charge of the orientation and the velocity along z_b of the quadcopter. The required settling time is about 2 seconds (excepted for the yaw angle). Therefore, we think that it is not useful for the inner controller to be concerned about what happens further in the future, also because the reference given by the outer controller may vary in the meantime. Since the system is sampled at $Ts = 0.1$s, 2 s correspond to an horizon length of $N = 20$. After a few trials, we noticed that the system was a bit more reactive with a shorter horizon length of $N = 10$, that we kept for the next parts.
The other parameters to tune were Q and R for the objective function. We chose them as follows:

$$Q = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

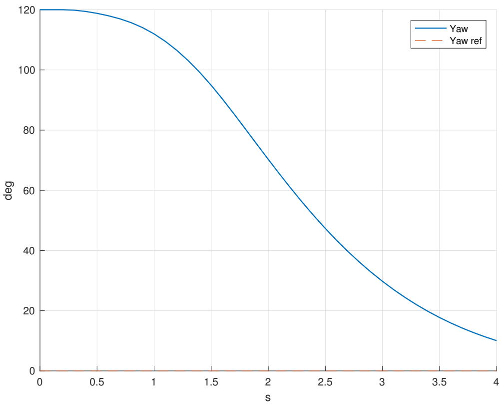
We started with identity matrices, then we increased the state cost for $\dot{z}$, α and β until we found satisfying results regarding the settling time of these variables. Figure 1 shows these results for an initial condition $X_0 = [-1 \quad 10 \quad -10 \quad 120 \quad 0 \quad 0 \quad 0]^T$ (where the angles are in degrees and need to be converted in radians before they are given to the controller). All the constraints are satisfied, the roll and pitch settling time are about 2 s and the settling time of $\dot{z}$ is much quicker, about 0.3 s. It takes a bit more than the 4 seconds of simulation for the yaw angle to go to 0, but it is still acceptable.



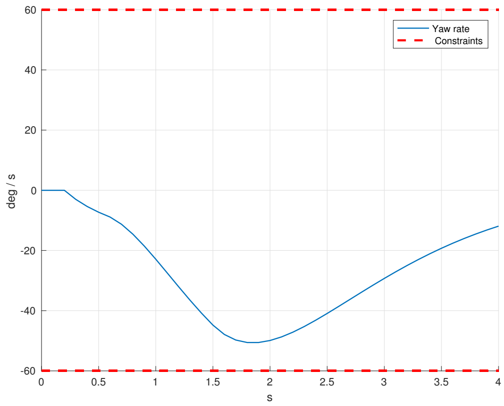
(a) Roll and pitch angles



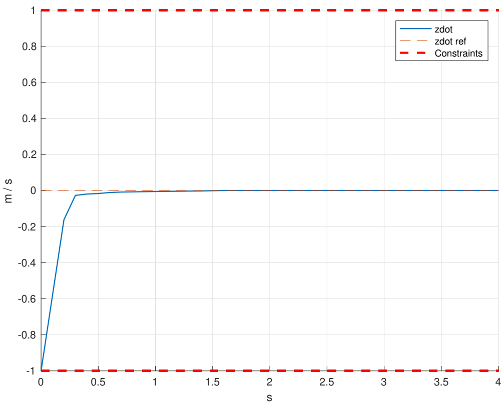
(b) Roll and pitch rates



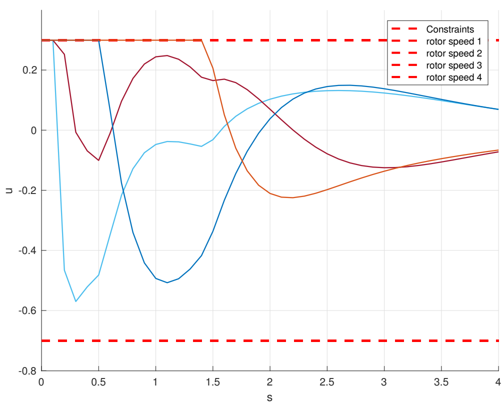
(c) Yaw angle



(d) Yaw rate



(e) $\dot{z}$



(f) Rotors speed

Figure 1: Constraints and performance for the first MPC controller, with $X_0 = [-1 \quad 0.1745 \quad -0.1745 \quad 2.0944 \quad 0 \quad 0 \quad 0]^T$ (angles in rad).

Task 3: Reference tracking

Deliverables

1. Interpretation of the solution (xr,ur) for arbitrary r1 (5%)
2. Plots of the response to a constant reference signal (10%)
3. Plots of the response to a slowly varying reference signal (5%)

Equilibrium states

To obtain the equilibrium states, we reverse the following equation:

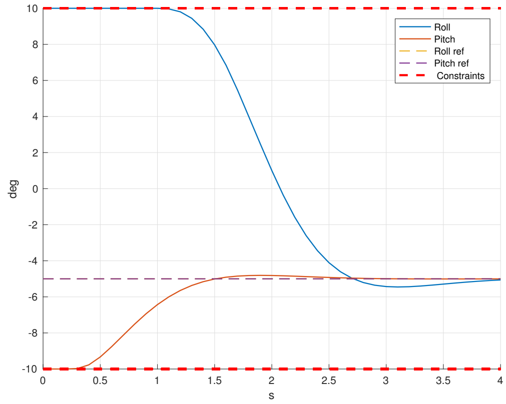
$$X_r = AX_r + Bu_r, \quad r1 = CX_r$$

Here is an example of a reference and the resulting equilibrium state:

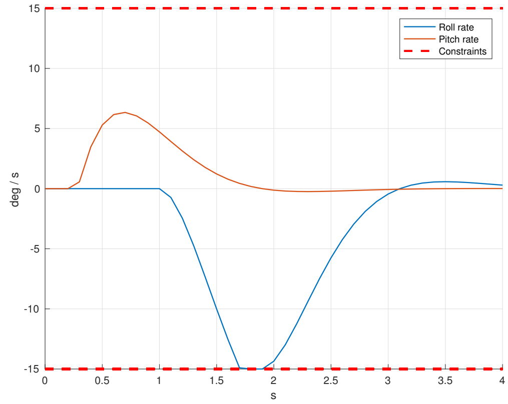
$$r1 = [1 \quad -0.1745 \quad 0.1745 \quad -2.0944]^T$$
$$X_r = [1 \quad -0.1745 \quad 0.1745 \quad -2.0944 \quad 0 \quad 0 \quad 0]^T, \quad u_r = [0 \quad 0 \quad 0 \quad 0]^T$$

Similar results are found for any reference point. It is intuitive that X_r directly follows $r1$, and that there is no angular velocities at steady state. However, the fact that the command u_r is zero is less intuitive. First, one should remember that it is not directly the command, but the variation of the command around the linearization point (equilibrium where the quadcopter is horizontal and does not move). Then, since the goal is to keep a constant velocity, the gravity should be compensated. Yet, with the small-angles approximation, the command to compensate the gravity stays the same around the linearization point. That is why the same command is always applied and we find $u_r = 0$ for any reference.

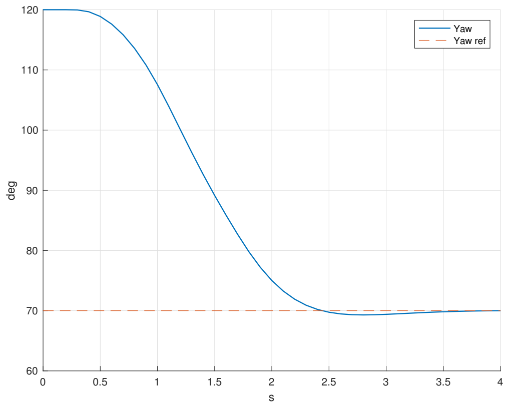
Figure 2 below shows the results for tracking a constant reference, while figure 3 shows the results for tracking a ramp signal as a reference.
As before, the constraints are always met and the performance is satisfying. For the constant reference, one can see that the command converges to zero, as explained above. For the ramp reference, a tracking error remains, in particular for the three angles. However, the quadcopter is globally able to follow the signal.



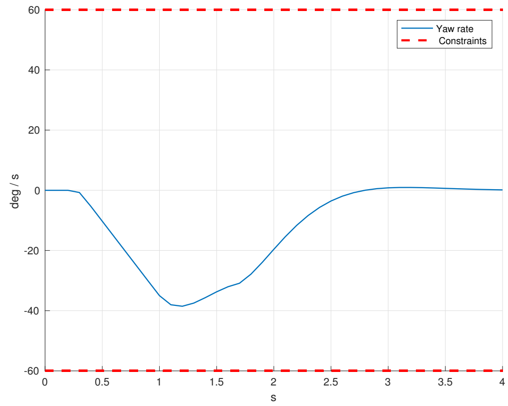
(a) Roll and pitch angles



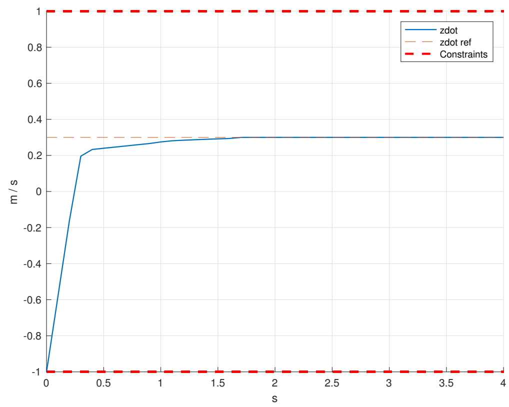
(b) Roll and pitch rates



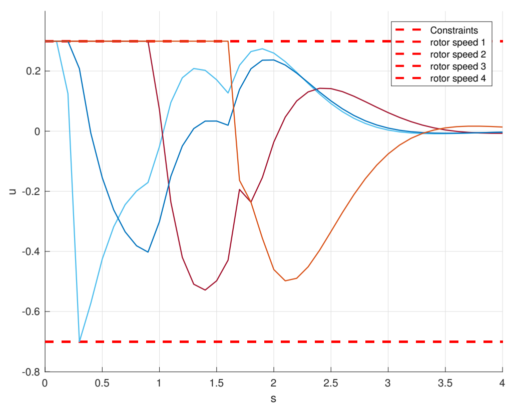
(c) Yaw angle



(d) Yaw rate

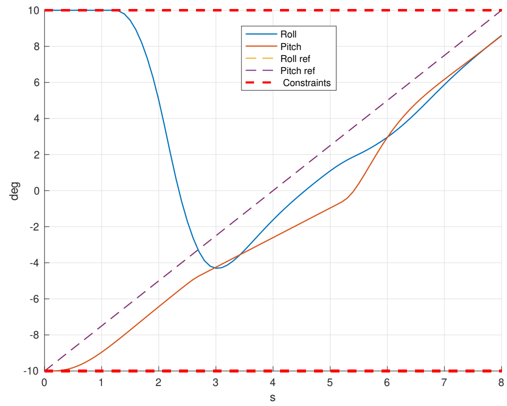


(e) $\dot{z}$

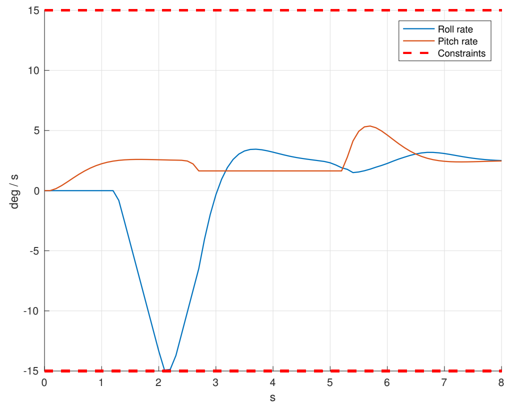


(f) Rotors speed

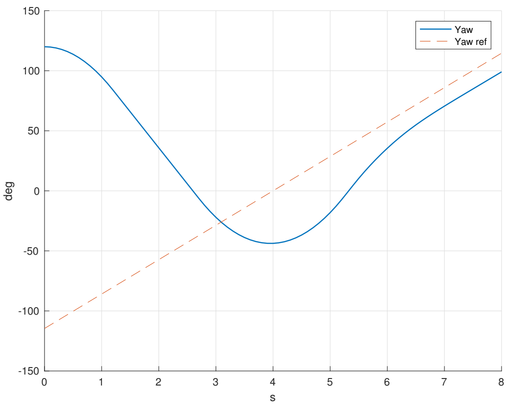
Figure 2: Constraints and performance for the reference tracking MPC controller, with $X_0 = [-1 \quad 0.1745 \quad -0.1745 \quad 2.0944 \quad 0 \quad 0 \quad 0]^T$ (angles in rad) and a constant reference $r_1 = [0.3 \quad -0.0873 \quad -0.0873 \quad 1.2217]^T$ (angles in rad).



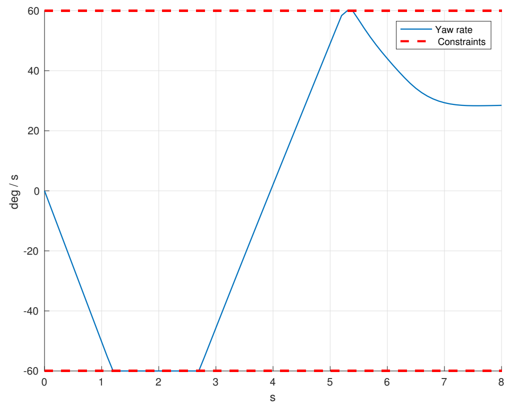
(a) Roll and pitch angles



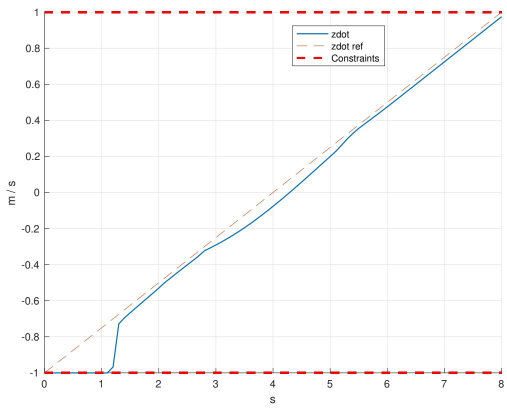
(b) Roll and pitch rates



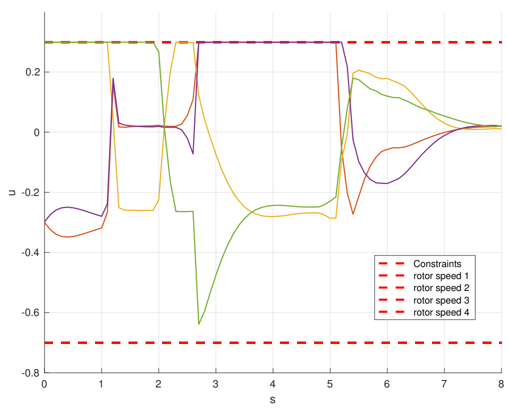
(c) Yaw angle



(d) Yaw rate



(e) $\dot{z}$



(f) Rotors speed

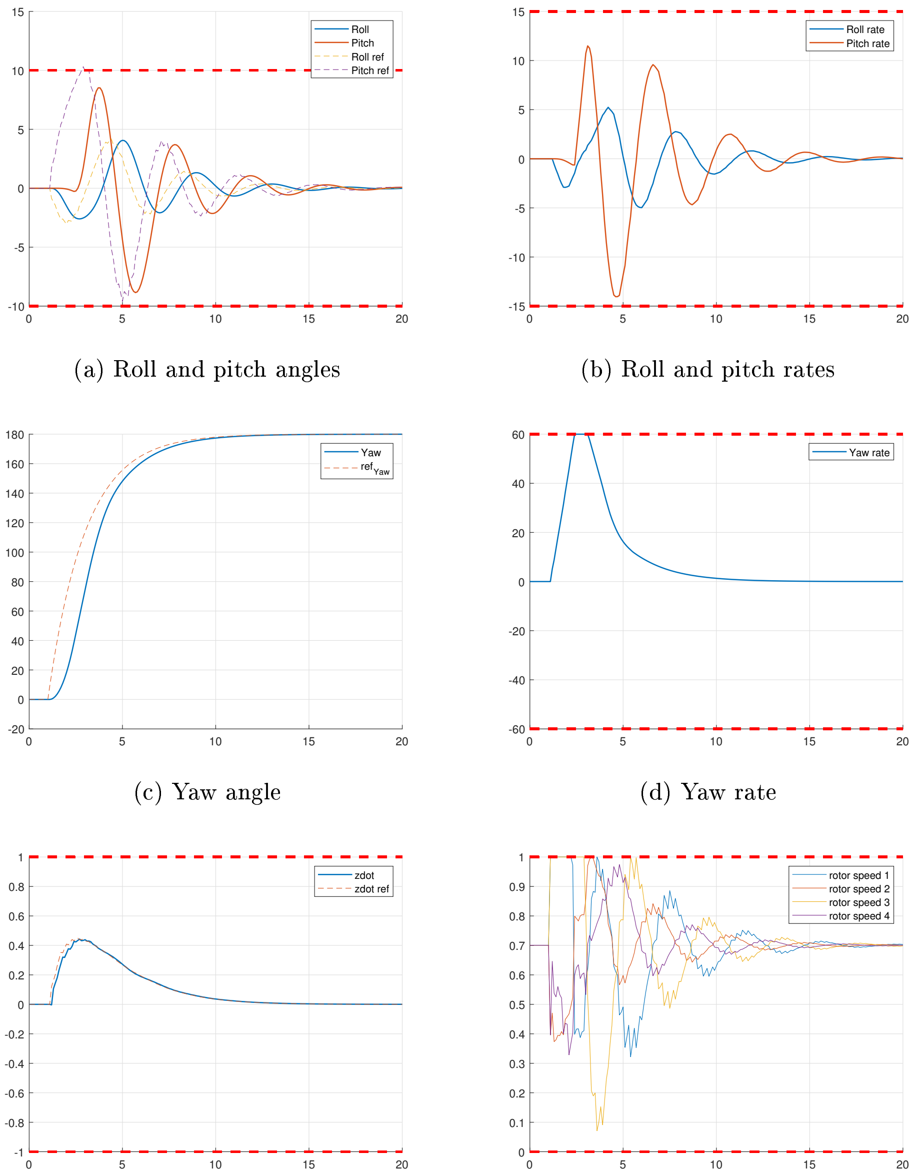
Figure 3: Constraints and performance for the reference tracking MPC controller, with $X_0 = [-1 \quad 0.1745 \quad -0.1745 \quad 2.0944 \quad 0 \quad 0 \quad 0]^T$ (angles in rad) and a ramp reference r_1 going from $[-1 \quad -0.1745 \quad -0.1745 \quad -2]^T$ to $[1 \quad 0.1745 \quad 0.1745 \quad 2]^T$ (angles in rad).

First simulation of the nonlinear model

Deliverables

Plots of a reference tracking response of the nonlinear model. (10%)

Figure 3 shows the reference tracking response of the nonlinear model. Once again, all constraints are satisfied. We can now also observe the trajectories of (x,y,z) of the drone.



(a) Roll and pitch angles

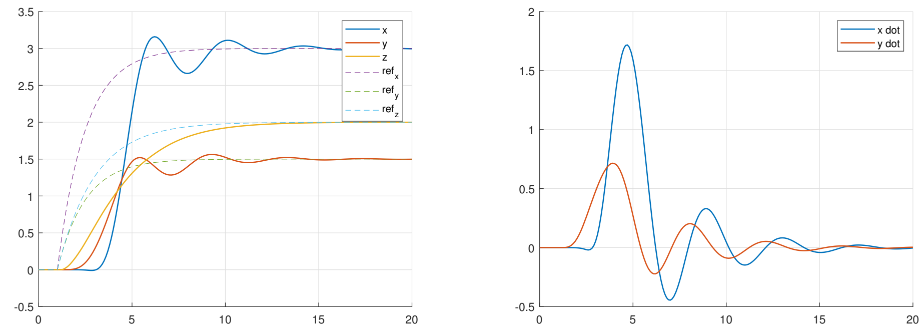
(b) Roll and pitch rates

(c) Yaw angle

(d) Yaw rate

(e) $\dot{z}$

(f) Rotors speed



(g) Position (x,y,z)

(h) $\dot{x}$ and $\dot{y}$

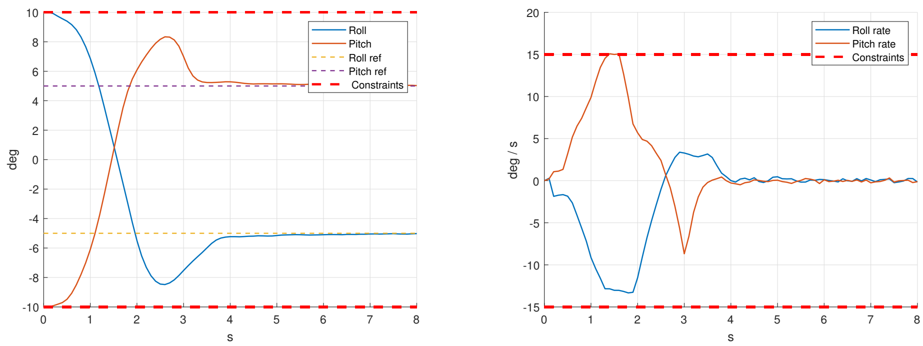
Figure 3: Constraints and performance for the nonlinear reference tracking MPC controller

Offset free MPC

Deliverables

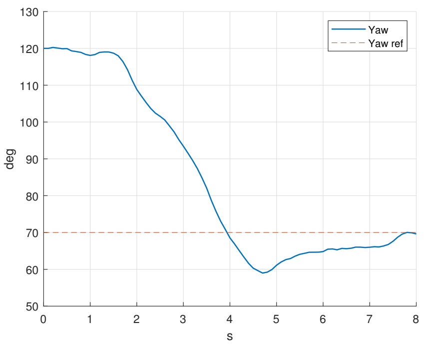
1. *Motivation for the choice of the estimation error dynamics (5%)*
2. *Step reference tracking plots in the presence of disturbance (15%)*
3. *Slowly-varying reference tracking plots in the presence of disturbance (10%)*

We used the pole placement technique for the estimator. We set the poles at 0.95 and 0.90 to avoid having an overshoot on the estimated disturbances. Indeed, the optimization problem was always infeasible for poles closer to 0, when there was an overshoot. The disadvantage of having poles close to 1 is that our estimator is a bit slow.
Figures 4 and 5 present the results for a step reference and a ramp reference, with disturbance.

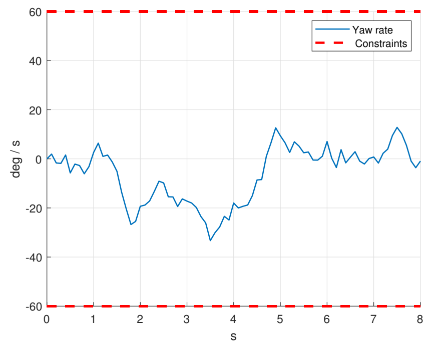


(a) Roll and pitch angles

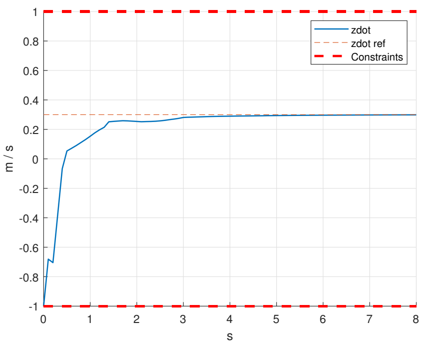
(b) Roll and pitch rates



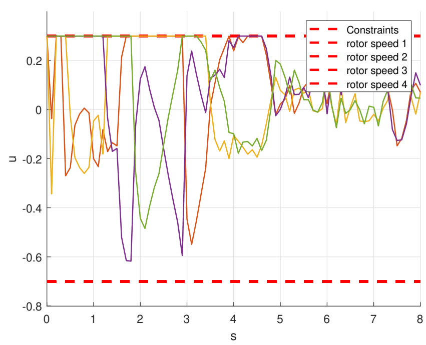
(c) Yaw angle



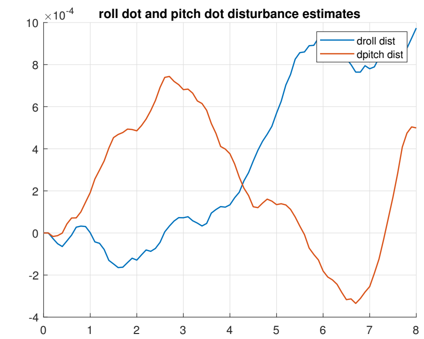
(d) Yaw rate



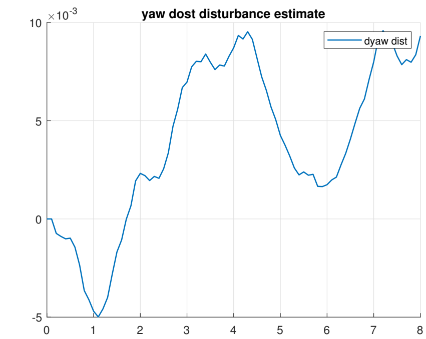
(e) $\dot{z}$



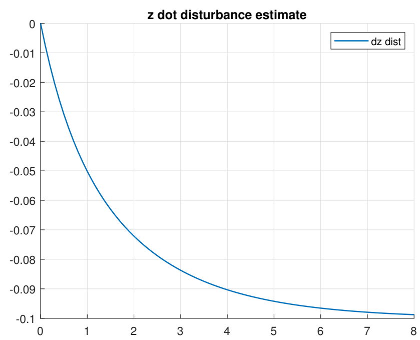
(f) Rotors speed



(g) Disturbance estimation on $\dot{\alpha}$ (roll dot) and $\dot{\beta}$ (pitch dot)

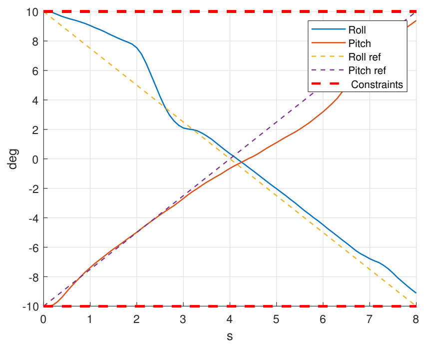


(h) Disturbance estimation on $\dot{\gamma}$ (yaw dot)

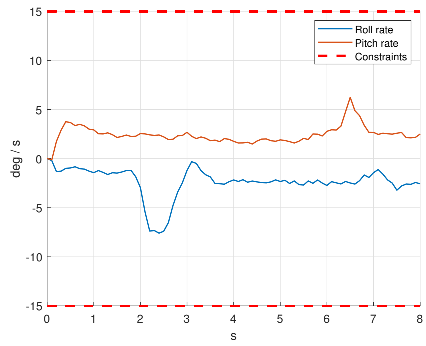


(i) Disturbance estimation on $\dot{z}$

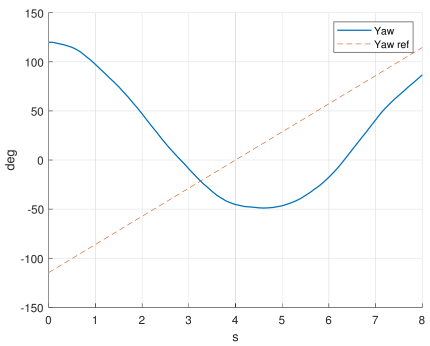
Figure 4: Constraints and performance for the reference tracking offset free MPC controller in presence of disturbance, with $X_0 = [-1 \quad 0.1745 \quad -0.1745 \quad 2.0944 \quad 0 \quad 0 \quad 0]^T$ (angles in rad) and a constant reference $r_1 = [0.3 \quad -0.0873 \quad -0.0873 \quad 1.2217]^T$ (angles in rad).



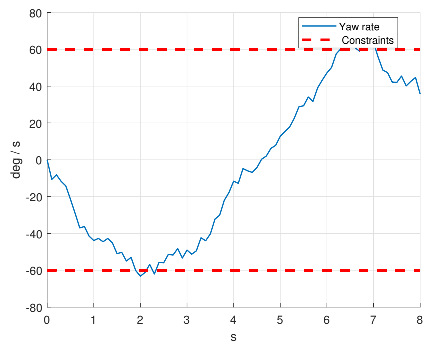
(a) Roll and pitch angles



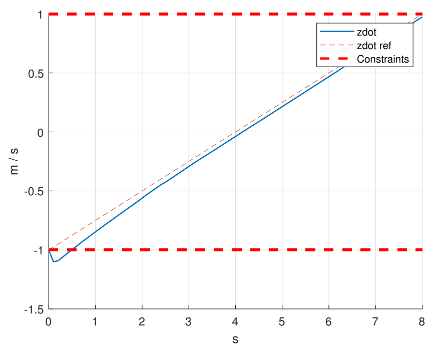
(b) Roll and pitch rates



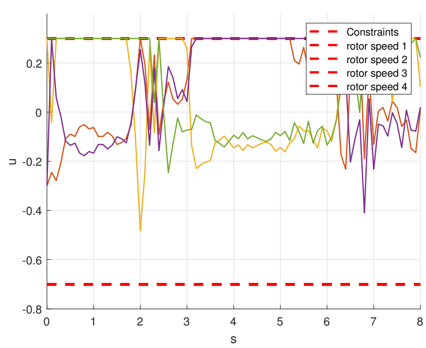
(c) Yaw angle



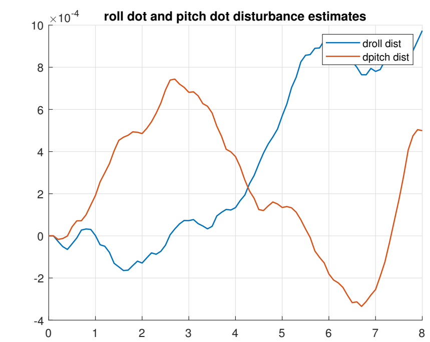
(d) Yaw rate



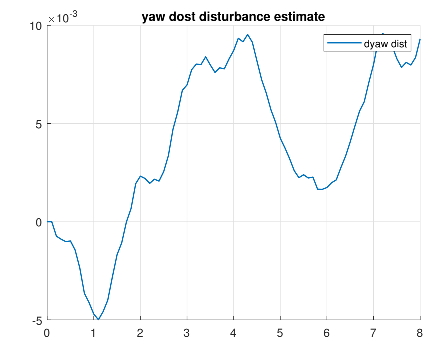
(e) $\dot{z}$



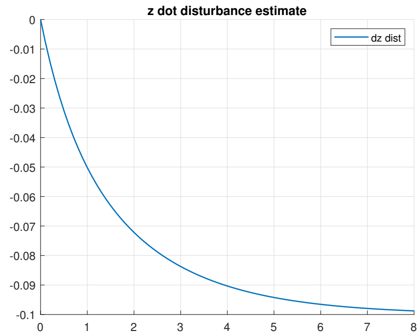
(f) Rotors speed



(g) Disturbance estimation on $\dot{\alpha}$ (roll dot) and $\dot{\beta}$ (pitch dot)



(h) Disturbance estimation on $\dot{\gamma}$ (yaw dot)



(i) Disturbance estimation on $\dot{z}$

Figure 5: Constraints and performance for the reference tracking offset free MPC controller in presence of disturbance, with $X_0 = [-1 \quad 0.1745 \quad -0.1745 \quad 2.0944 \quad 0 \quad 0 \quad 0]^T$ (angles in rad) and a ramp reference r_1 going from $[-1 \quad -0.1745 \quad -0.1745 \quad -2]^T$ to $[1 \quad 0.1745 \quad 0.1745 \quad 2]^T$ (angles in rad).

Simulations on the nonlinear model

Deliverables

1. *Plots of the reference tracking of a step signal (10%)*
2. *Plots of the reference tracking of the hexagon signal (5%)*
3. *Bonus: plots of the reference tracking of an "eight" (+10%)*
4. *Bonus: add slew rate constraints on the control inputs*

Figures 6 to 11 show the results for these 3 situations. The quadcopter is always able to follow the references, but a delay is observed, particularly for the positions (x, y, z). To obtain better results, we tuned the state cost (keeping the same command cost), to obtain these new values:

$$Q = \begin{pmatrix} 1000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

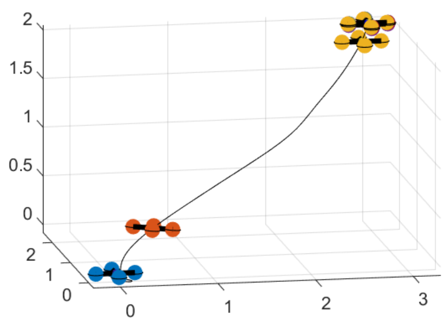
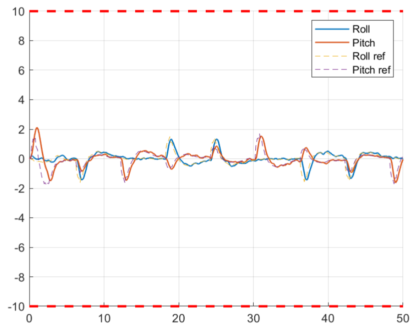
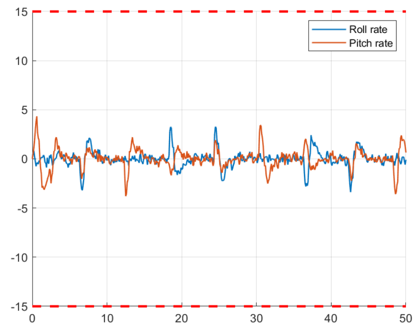


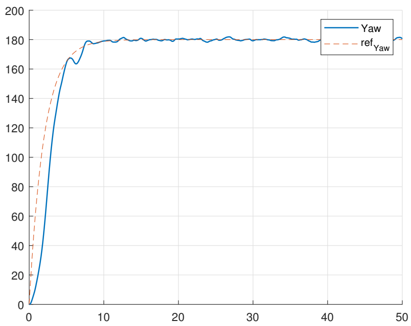
Figure 6: Trajectory for the reference tracking offset free MPC controller in presence of disturbance, for the nonlinear model with a step reference.



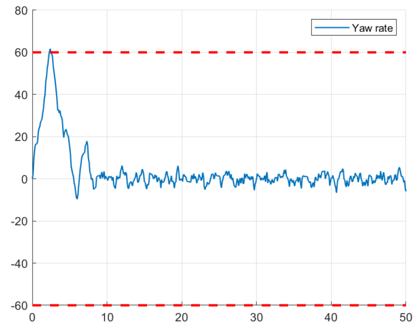
(a) Roll and pitch angles



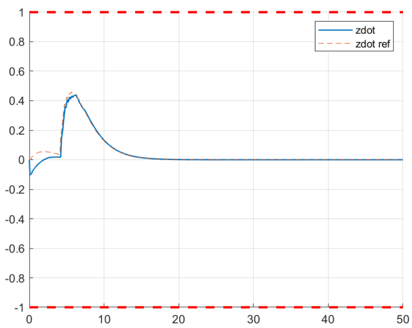
(b) Roll and pitch rates



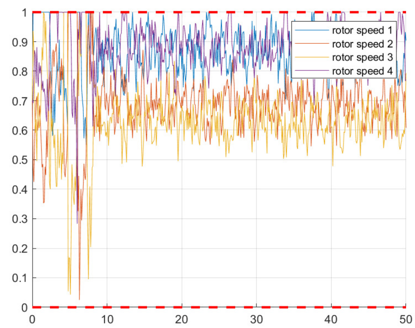
(c) Yaw angle



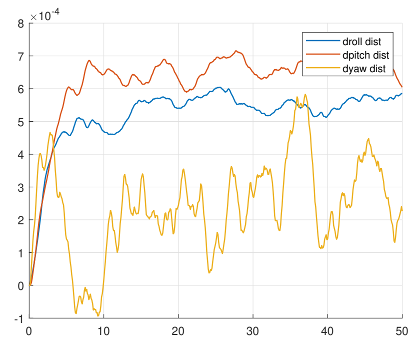
(d) Yaw rate



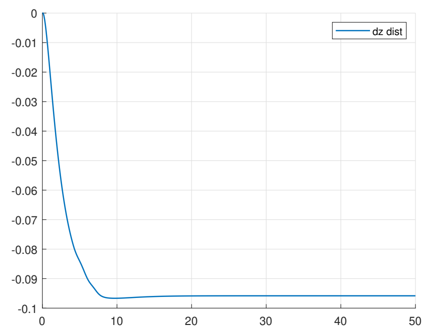
(e) $\dot{z}$



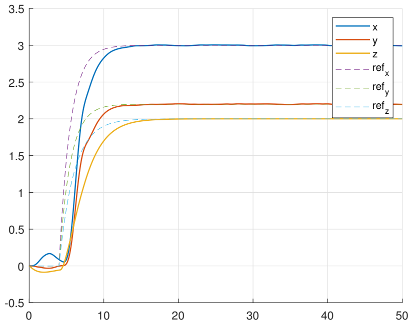
(f) Rotors speed



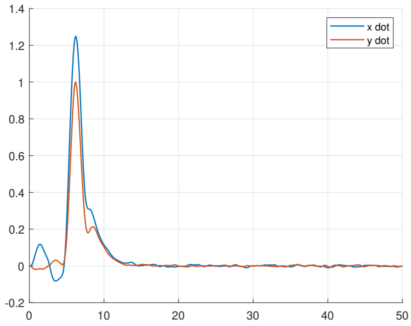
(g) Disturbance estimation on roll, pitch and yaw angles



(h) Disturbance estimation on $\dot{z}$



(i) Position (x,y,z)



(j) $\dot{x}$ and $\dot{y}$

Figure 7: Constraints and performance for the reference tracking offset free MPC controller in presence of disturbance, for the nonlinear model with a step reference.

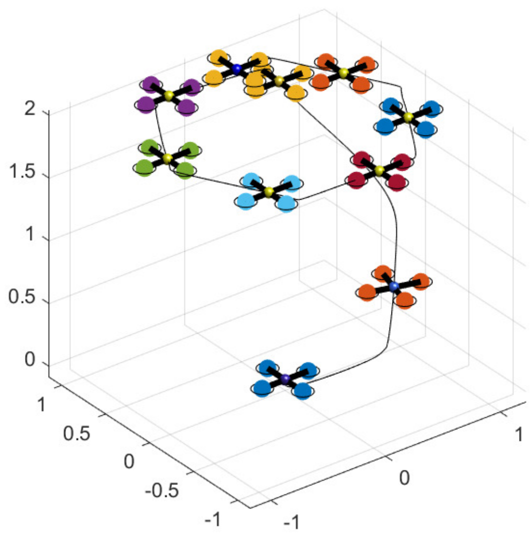
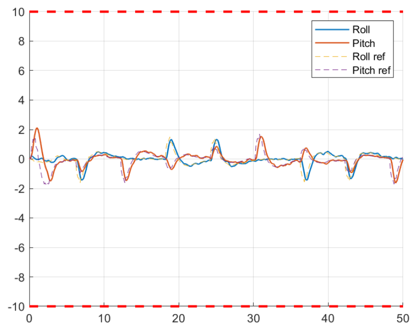
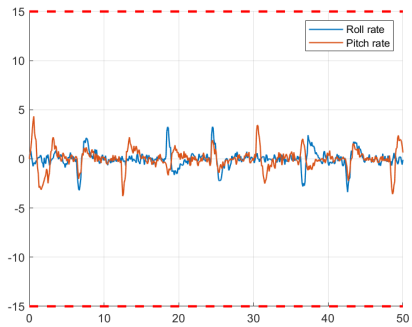


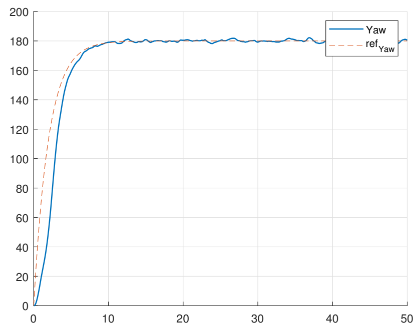
Figure 8: Trajectory for the reference tracking offset free MPC controller in presence of disturbance, for the nonlinear model with a hexagon reference.



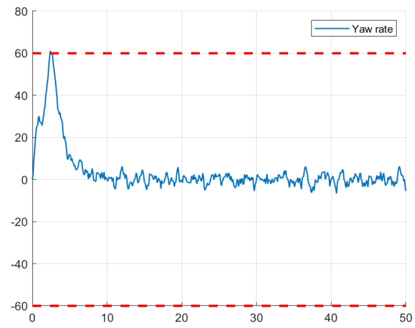
(a) Roll and pitch angles



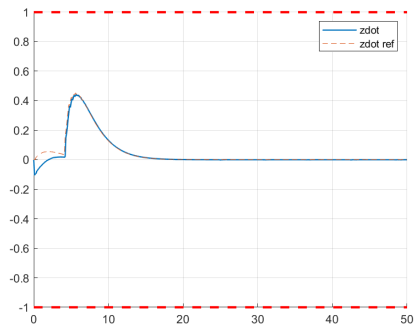
(b) Roll and pitch rates



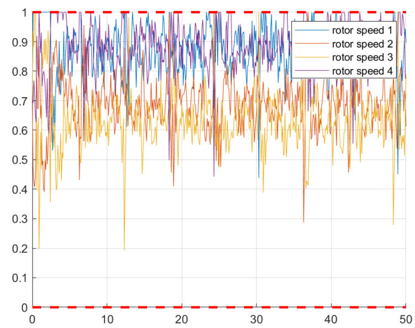
(c) Yaw angle



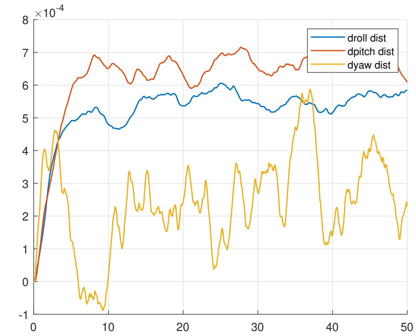
(d) Yaw rate



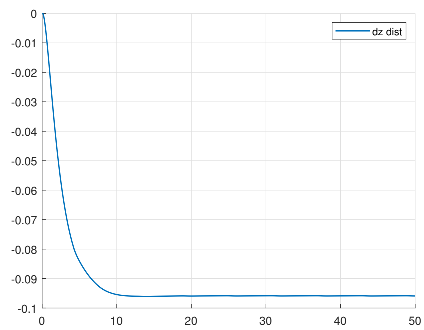
(e) $\dot{z}$



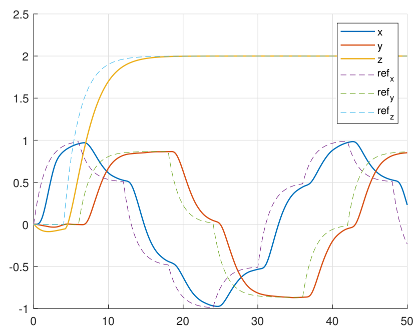
(f) Rotors speed



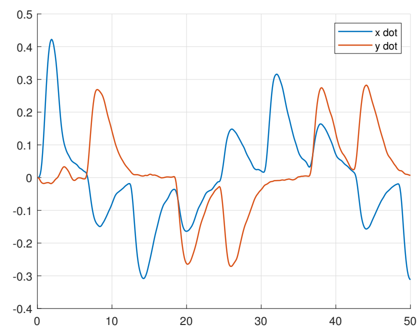
(g) Disturbance estimation on roll, pitch and yaw angles



(h) Disturbance estimation on $\dot{z}$



(i) Position (x,y,z)



(j) $\dot{x}$ and $\dot{y}$

Figure 9: Constraints and performance for the reference tracking offset free MPC controller in presence of disturbance, for the nonlinear model with a hexagon reference.

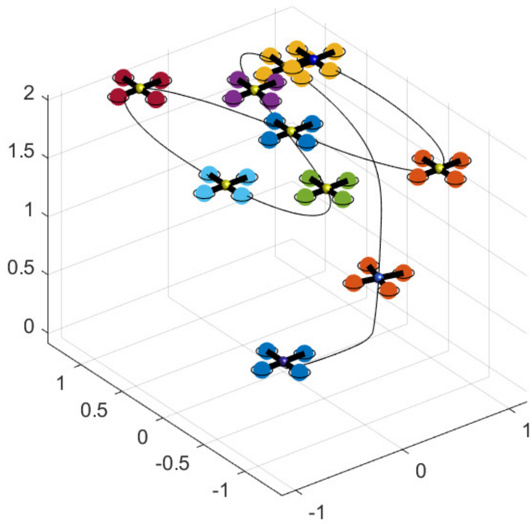
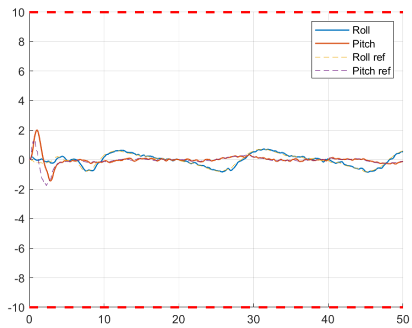
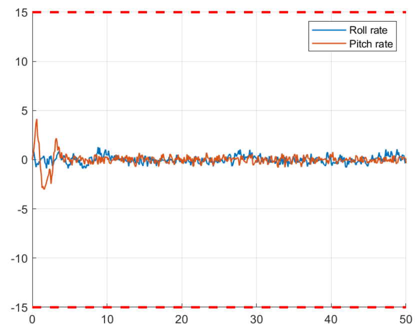


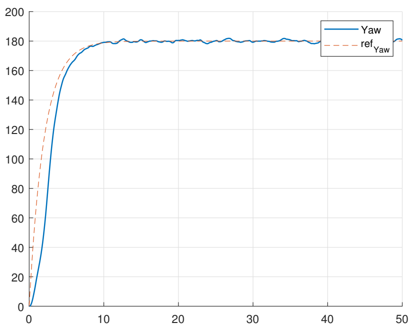
Figure 10: Trajectory for the reference tracking offset free MPC controller in presence of disturbance, for the nonlinear model with a lemniscate reference.



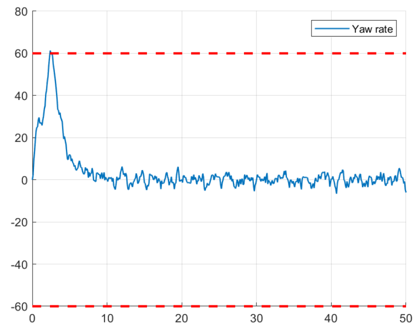
(a) Roll and pitch angles



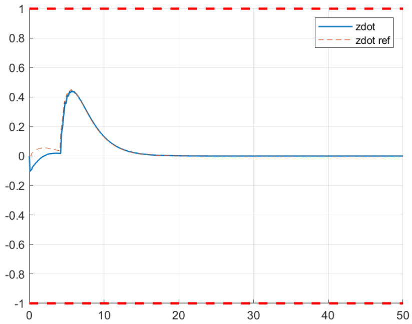
(b) Roll and pitch rates



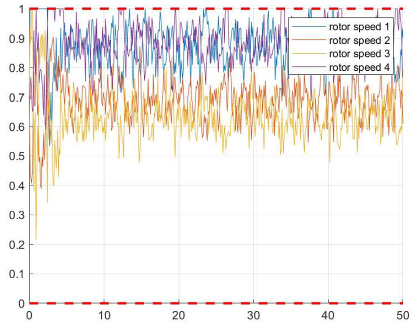
(c) Yaw angle



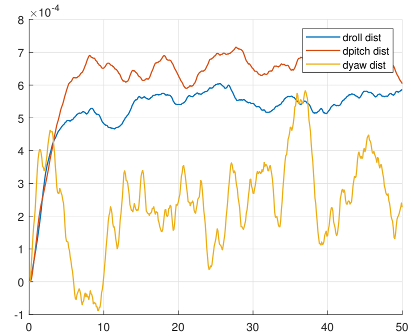
(d) Yaw rate



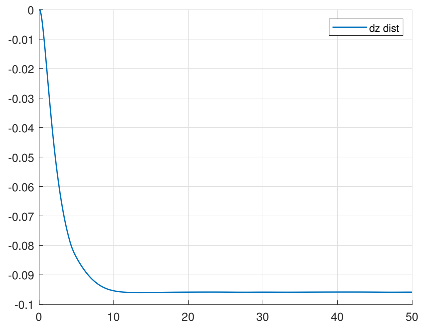
(e) $\dot{z}$



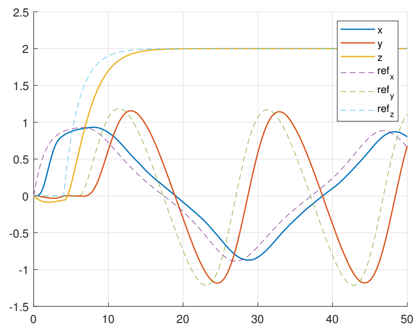
(f) Rotors speed



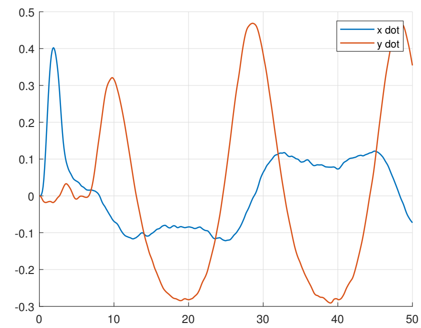
(g) Disturbance estimation on roll, pitch and yaw angles



(h) Disturbance estimation on $\dot{z}$



(i) Position (x,y,z)



(j) $\dot{x}$ and $\dot{y}$

Figure 11: Constraints and performance for the reference tracking offset free MPC controller in presence of disturbance, for the nonlinear model with a lemniscate reference.

The 8-like reference trajectory was computed using the Bernoulli lemniscate curve (see the equations below). The a_i and p parameters were tuned in order to make one full "8" in 50 seconds of simulation while staying in the x and y constraints. Based on the previous simulink implementation of the hexagonal trajectory, a delay of 6 seconds has been set on the reference signals to first let the drone rise before it begins the trajectory.

$$\begin{cases} x(t) = \frac{a\cdot\sqrt{2}\cdot cos(\frac{t}{p})}{sin^2(\frac{t}{p})+1} \\ y(t) = \frac{a\cdot\sqrt{2}\cdot cos(\frac{t}{p})\cdot sin(\frac{t}{p})}{sin^2(\frac{t}{p})+1} \\ \text{with } a_1 = \frac{2}{3},\ a_2 = \frac{2}{15} \text{ and } p = 2\pi \end{cases}$$

Here is the simulink block schematic used to generate the 8-like trajectory:

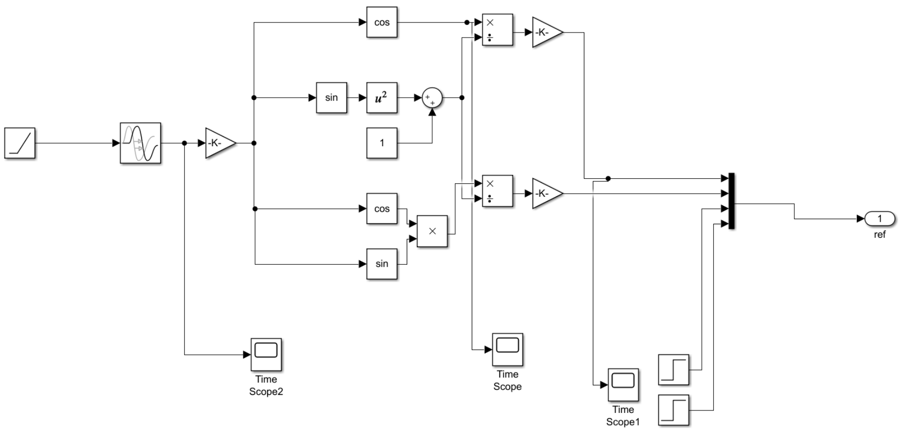
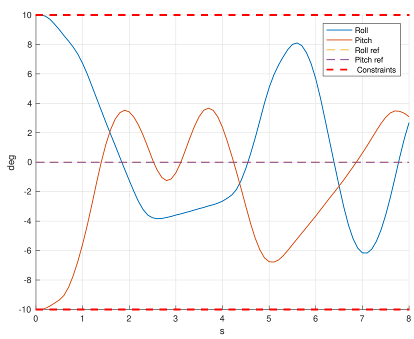


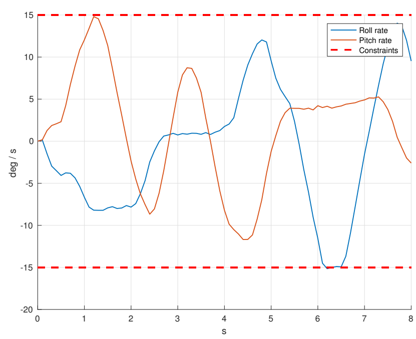
Figure 12: Simulink block schematic

Bonus: slew rate constraints

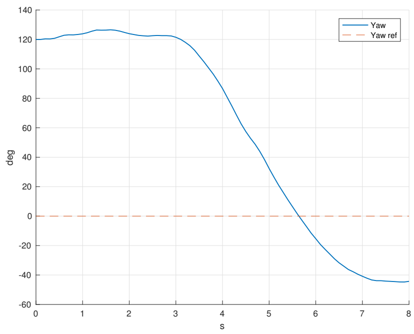
Using the linear offset free MPC, we added slew rate constraints, of the form $|u_{k+1} - u_k| < \Delta$. We used the model with default disturbance and started with a Δ of 1 (maximum possible variation to stay inside the command constraints) that we progressively decreased. Until $\Delta = 0.8$, a feasible solution was found. But already at that point, the performance was not satisfying anymore. Results are shown on figure 13.



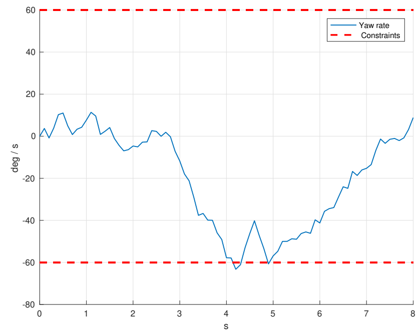
(a) Roll and pitch angles



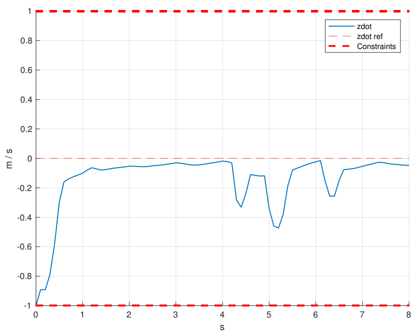
(b) Roll and pitch rates



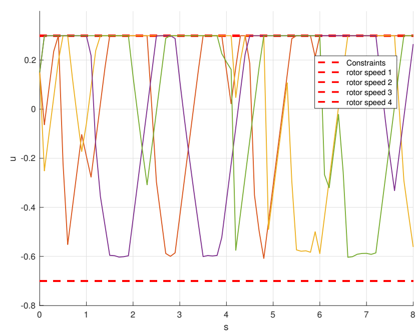
(c) Yaw angle



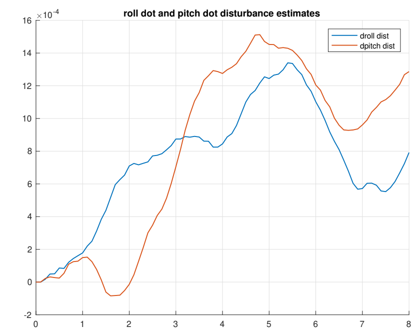
(d) Yaw rate



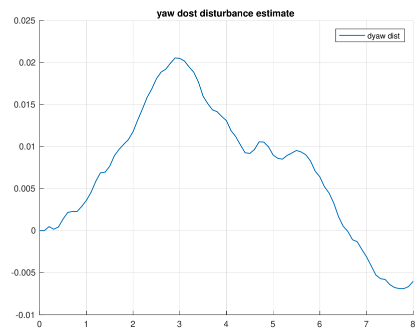
(e) $\dot{z}$



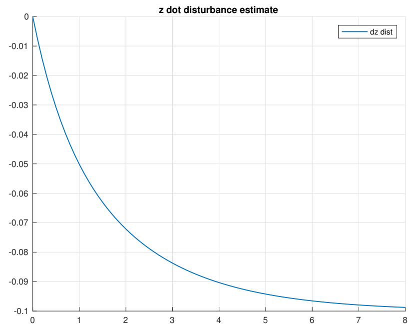
(f) Rotors speed



(g) Disturbance estimation on $\dot{\alpha}$ (roll dot) and $\dot{\beta}$ (pitch dot)



(h) Disturbance estimation on $\dot{\gamma}$ (yaw dot)



(a) Disturbance estimation on $\dot{z}$

Figure 13: Constraints and performance for the reference tracking offset free MPC controller in presence of disturbance, with $X_0 = [-1 \quad 0.1745 \quad -0.1745 \quad 2.0944 \quad 0 \quad 0 \quad 0]^T$ (angles in rad) and a ramp reference r_1 going from $[-1 \quad -0.1745 \quad -0.1745 \quad -2]^T$ to $[1 \quad 0.1745 \quad 0.1745 \quad 2]^T$ (angles in rad).

The tracking becomes very bad. As it is, adding this constraints did not really improve anything. Also, using a Δ smaller than 0.8 always led to infeasible solutions.
To improve feasibility, one should probably consider using soft constraints wherever possible (probably not on (x, y, z) if one wants to avoid collisions, but maybe on u and for the slew rate).