ME-425 Model Predictive Control Mini-project 2023

Gabriel Paffi : Groupe CD
Sciper : 310436

8 January 2024

Part 1 : System Dynamics

The objective of this project is to use all knowledge acquired during the course of Model Predictive Control to develop a controller for a rocket propeller. During the project we will first develop a linear MPC with the different subsystems, then track some references and then we will simulate a non linear MPC.

Part 2 : Lineralization

We first lineralize our system about the trim point using the code given and then we decompose our system into 4 subsystems.

Deliverable 2.1

We were able to decompose our system with the 12 states : $\omega_x, \omega_y, \omega_z, \alpha\beta, \gamma, v_x, v_y, v_z, x, y, z$, into multiple systems for two mains reasons :
- firstly : as mentioned in the report with the different equations, the link between different states, for example the angle δ_2 is related to position x, the difference throttle P_{diff} is related to γ
- secondly : because we work with a linear system, we can then manage to decompose our system into multiple parts considering the angle α and β relatively low. In fact at the end of the project we will see that we cannot decompose our system into multiple subsystems for non linear modelization.
Each input given can determine a set of states.

Part 3 : Design MPC Controllers for Each Sub-System

Deliverable 3.1

For each subsystem, we will solve the following optimisation problem :

$$J^*(x) = \min \sum_{i=1}^{N-1} x_i^T Q x_i + u_i^T R u_i + x_N^T P x_N$$

$$\text{s.t.} \quad x_{i+1} = A x_i + B u_i$$

$$C x_i + D u_i \leq b$$

$$x_n \in X_f$$

$$x_0 = x$$

We will ensure the recursive feasibility using the terminal constraints for our MPC (Proof of the results in slides *Week 5 : MPC*. We choose an horizon time of 4 second to enable the controller to reach the terminal set. We then compute the terminal value and penalty using the LQR method with the LTI system on Matlab. We will now look at the different constraints, terminals sets and cost matrices for each subsystems. In general we will choose for the deliverable 3,4 and 5 an horizon length H = 4 seconds it will enable the system to reach the terminal value and also stay reasonable in term of computational cost for the linear systems.

3.1.1 Controller X

3.1.1.1 Constraints

The controller X have 4 states : $x_X = [\omega_y, \beta, v_x, x]$ and the input relative to the system is $u_X = \delta_2$ We will first look at the physical constraints related to this subsystem. For the states, the only constraint is on β : $|\beta| <= 10$. For the input : we have $\delta_2 \in [-0.26, 0.26]$ We can then rewrite in a matricial form the set of constraints that the system is subjected to :

$Fx < f$ with $F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$ and $f = \begin{bmatrix} 0.1745 \\ 0.1745 \end{bmatrix}$ $Mu < m$ with $M = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and $m = \begin{bmatrix} 0.26 \\ 0.26 \end{bmatrix}$

3.1.1.2 Cost Matrices

We will use higher weights on the position x and the velocity w_y to have better tracking and to avoid violation on the angle beta condition, since there are no specifications for the input δ_2, R is set to 1. so :

$$Q = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \text{ and } R = 1$$

3.1.1.3 Terminal Set

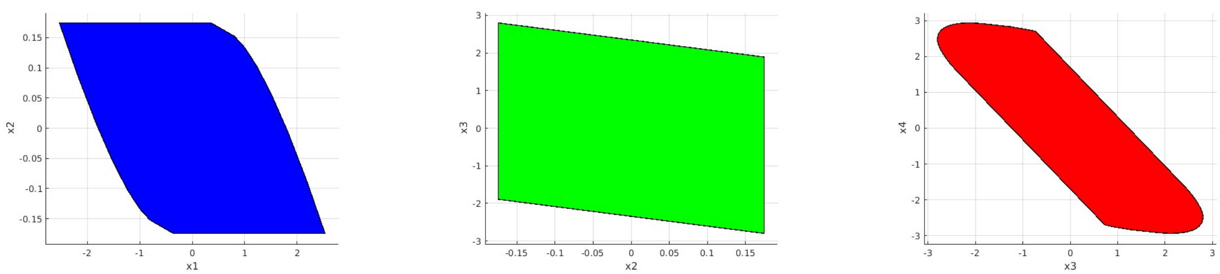

Figure 1: Terminal set's projections of the controller X

3.1.1.4 Simulation

For the X Controller, the rocket start at an initial position $x = [0, 0, 0, 3]$ relative to the subsystem or at $x_0 = [3, 0, 0]$ in the 3D plane.

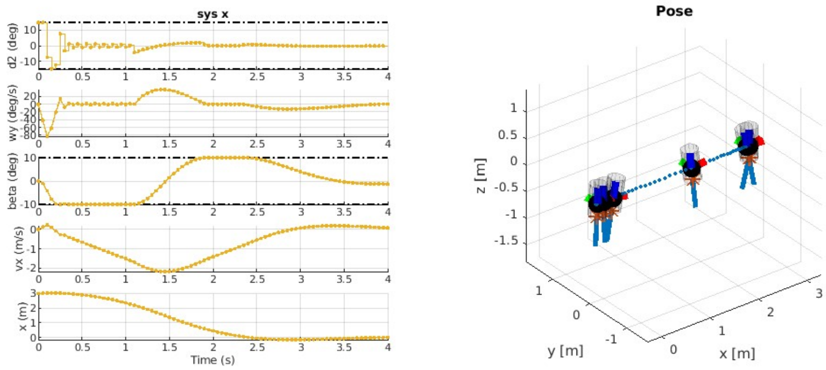

Figure 2: Open loop simulation on controller X

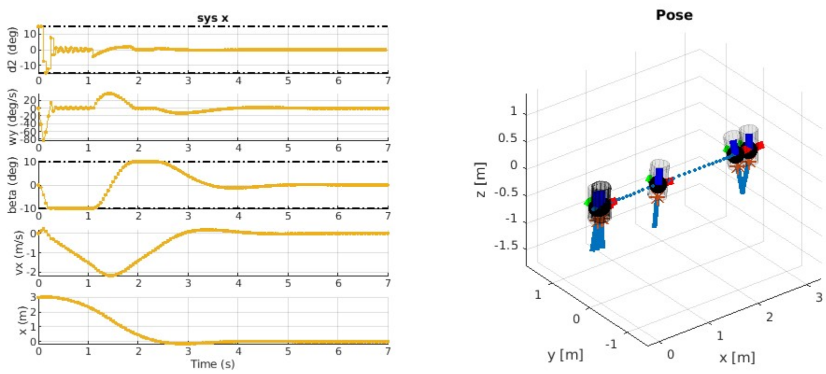

Figure 3: Closed loop simulation on controller X

3.1.2 Controller Y

3.1.2.1 Constraints

The controller Y have 4 states : $x_Y = [\omega_x, \alpha, v_y, y]$ and the input relative to the system is $u_Y = \delta_1$ We will first look at the physical constraints related to this subsystem. For the states, the only constraint is on α : $|\alpha| <= 10$. For the input : we have $\delta_1 \in [-0.26, 0.26]$ We can then rewrite in a matricial form the set of constraints that the system is subjected to :

$Fx < f$ with $F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}$ and $f = \begin{bmatrix} 0.1745 \\ 0.1745 \end{bmatrix}$ $Mu < m$ with $M = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and $m = \begin{bmatrix} 0.26 \\ 0.26 \end{bmatrix}$

3.1.2.2 Cost Matrices

We will use higher weights on the position y and the velocity w_x to have better tracking and to avoid violation on the angle beta condition, since there are no specifications for the input δ_1, R is set to 1. so :

$$Q = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \text{ and } R = 1$$

3.1.2.3 Terminal Set

For The Y controller the terminal set is identical to the X controller.

3.1.2.4 Simulation

For the Y Controller, the rocket start at an initial position $x = [0, 0, 0, 3]$ relative to the subsystem or at $x_0 = [0, 3, 0]$ in the 3D plane.

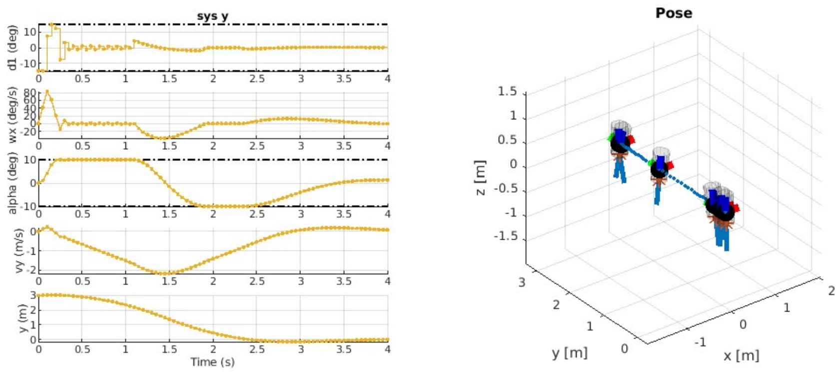

Figure 4: Open loop simulation on controller Y

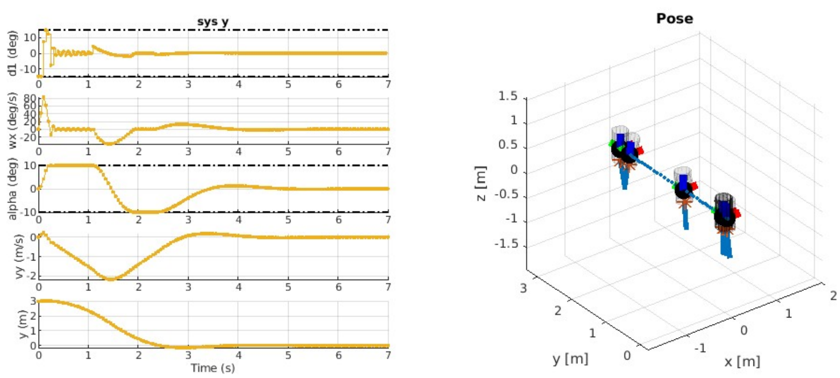

Figure 5: Closed loop simulation on controller Y

3.1.3 Controller Z

3.1.3.1 Constraints

The controller Z have 2 states : $x_Z = [v_z, z]$ and the input relative to the system is $u_Z = P_{avg}$ We will first look at the physical constraints related to this subsystem. For the states, there is no constraints on it. For the input : we have $P_{avg} \in [50, 80]$ We shifted our constraints towards 0 using the steady state value of the throttle : $P_{avg} = 56.6667$ We can then rewrite in a matricial form the set of constraints that the system is subjected to :

$$Mu < m \text{ with } M = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{ and } m = \begin{bmatrix} 23.66 \\ 6.33 \end{bmatrix}$$

3.1.3.2 Cost Matrices

We will use higher weight on the position z to have a fast tracking of the position with R set to 1 so :

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix} \text{ and } R = 0.1$$

3.1.3.3 Terminal Set

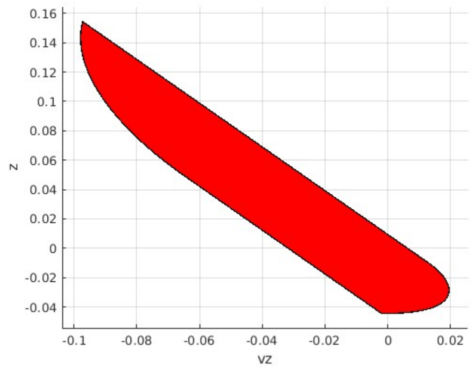

Figure 6: Terminal Set on controller Z

3.1.3.4 Simulation

For the Z Controller, the rocket start at an initial position $x = [0, 3]$ relative to the subsystem or at $x_0 = [0, 0, 3]$ in the 3D plane.

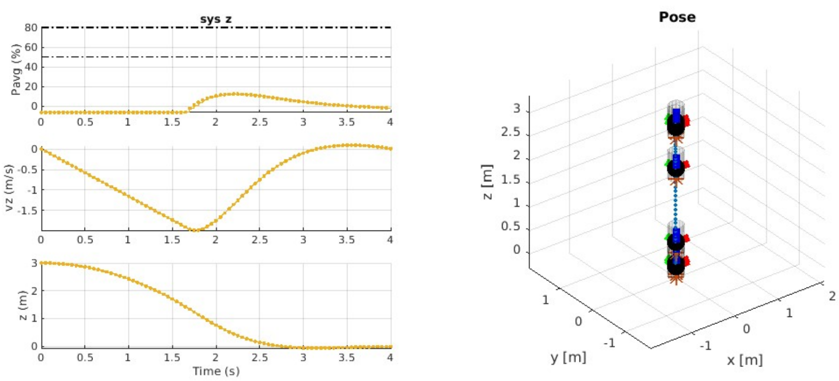

Figure 7: Open loop simulation on controller Z

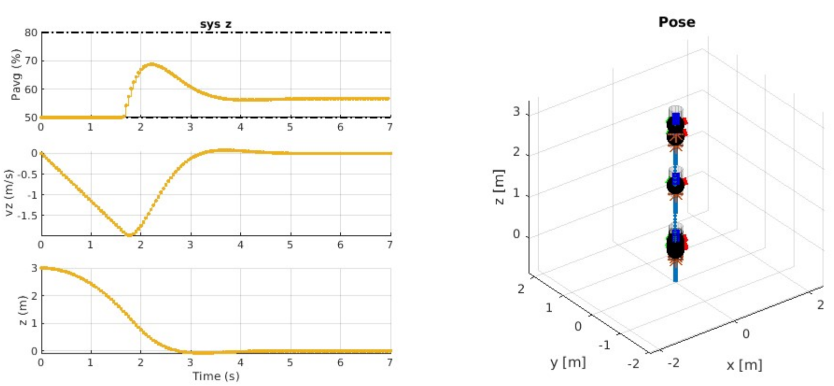

Figure 8: Closed loop simulation on controller Z

3.1.4 Controller Γ

3.1.4.1 Constraints

The controller Γ have 4 states : $x_\Gamma = [\omega_z, \gamma]$ and the input relative to the system is $u_\Gamma = P_{diff}$ We will first look at the physical constraints related to this subsystem. For the states, there is no constraints on it. For the input : we have $P_{diff} \in [-20, 20]$

We can then rewrite in a matricial form the set of constraints that the system is subjected to :

$Mu < m$ with $M = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and $m = \begin{bmatrix} 20 \\ 20 \end{bmatrix}$

3.1.4.2 Cost Matrices

We will use higher weight on the angle gamma to have a fast regulation and also decrease R to 0.1 . to smoother the action related to P_diffà :

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix} \text{ and } R = 0.1$$

3.1.4.3 Terminal Set

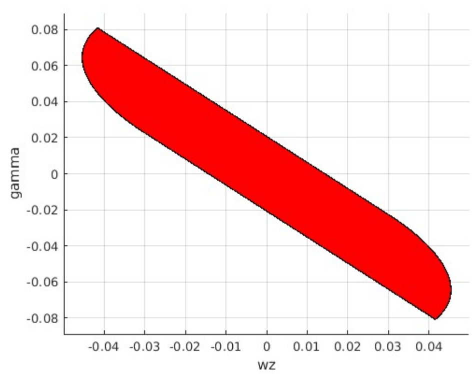

Figure 9: Terminal Set on controller Γ

3.1.4.4 Simulation

For the Γ Controller, the rocket start at an initial position $x = [0, 30]$ relative to the subsystem or at $x_0 = [0, 0, 0]$ with an orentation of 30 degrees in the 3D plane.

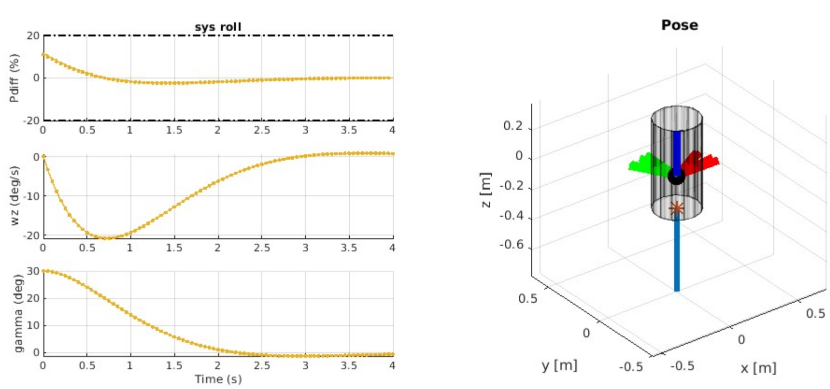

Figure 10: Open loop simulation on controller Γ

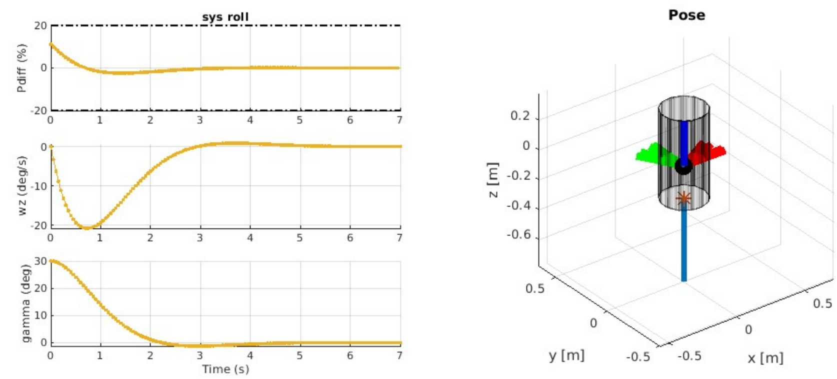

Figure 11: Closed loop simulation on controller Γ

Deliverable 3.2

In this part, we want to track a reference, in the previous part we expected that our state return to the steady state, no we need to change our model to track a certain value x_{ref} of the different state. We will therefore consider the following optimization problem :

$$J^*(x) = \min \sum_{i=1}^{N-1} \Delta x_i^T Q \Delta x_i + \Delta u_i^T R \Delta u_i + Vf(\Delta X_N)$$

$$\text{s.t.} \quad \Delta x_0 = \Delta x$$

$$\Delta x_{i+1} = A\Delta x_i + B\Delta u_i$$

$$H_x \Delta X_i \leq k_x - H_x x_s$$

$$H_u \Delta u_i \leq k_u - H_u u_s$$

$$\Delta x = x - x_{ref}$$

$$\Delta u = u - u_s$$

$$\Delta x_n \in X_f$$

We will then change in our code, constraints and the objective of the controller as following :

$$con = [M * us <= m, F * xs <= f, xs == mpc.A * xs + mpc.B * us, ref == mpc.C * xs + mpc.D];$$

$$obj = us^2;$$

We will keep the same cost matrices for the X,Y,Γ controller and the horizon time large enough to ensure feasibility, but for the controller Z we change the value of the matrice Q to have a more aggressive correction on potential mistracking .

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 500 \end{bmatrix}$$

3.2.1 Simulation Controller X

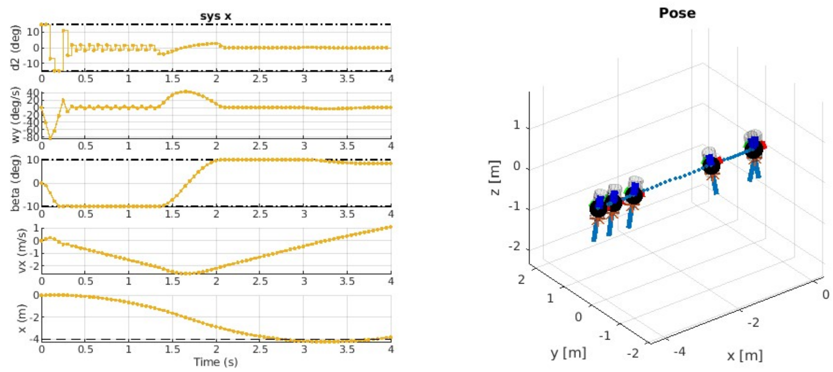

Figure 12: Open loop simulation on controller X with tracking

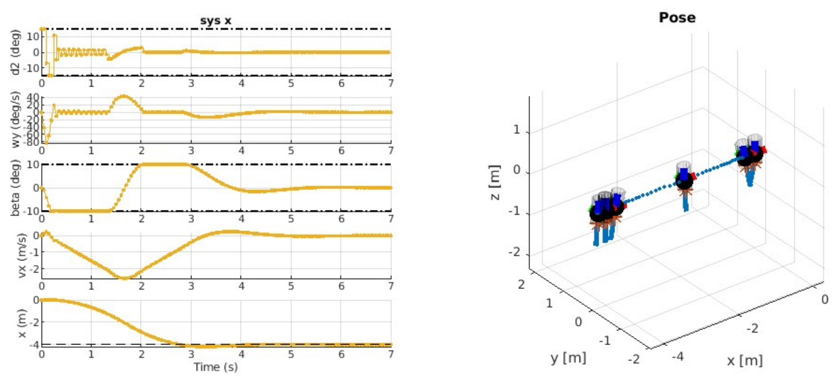

Figure 13: Closed loop simulation on controller X with tracking

3.2.2 Simulation Controller Y

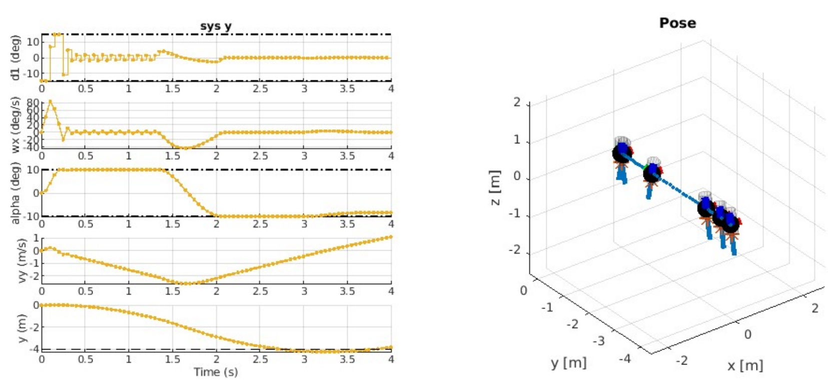

Figure 14: Open loop simulation on controller Y with tracking

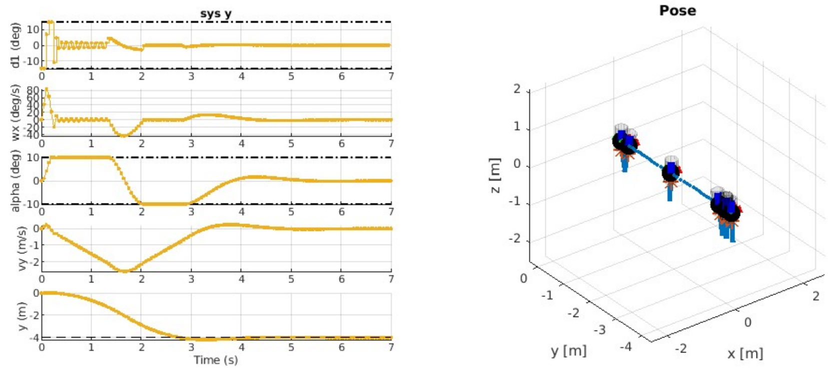

Figure 15: Closed loop simulation on controller Y with tracking

3.2.3 Simulation Controller Z

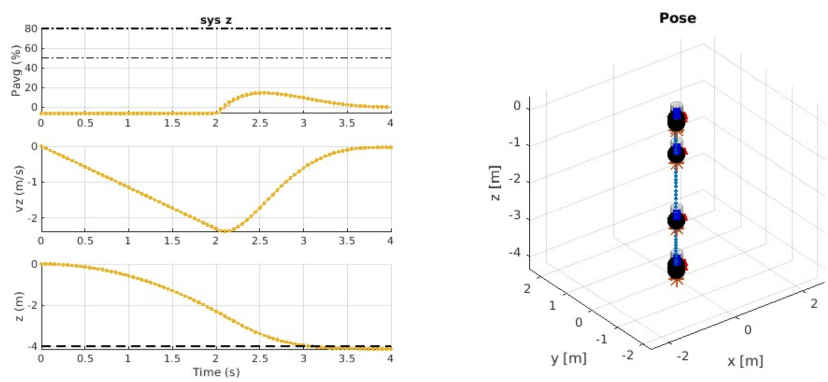

Figure 16: Open loop simulation on controller Z with tracking

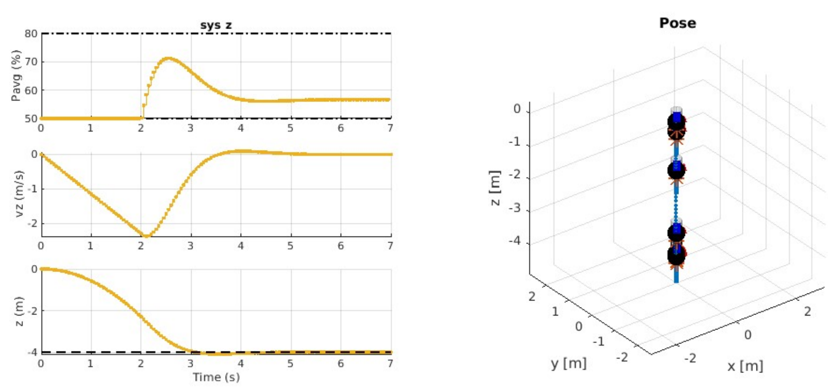

Figure 17: Closed loop simulation on controller Z with tracking

3.2.4 Simulation Controller Γ

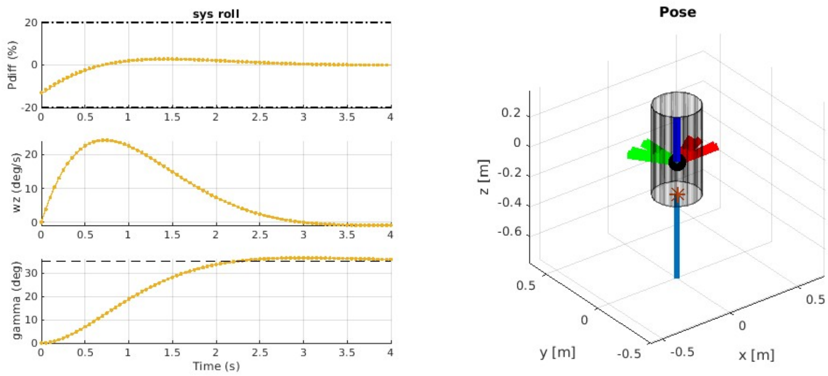

Figure 18: Open loop simulation on controller Γ with tracking

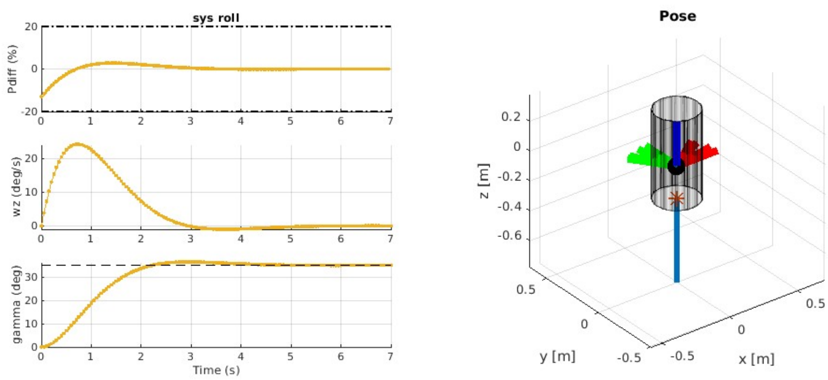

Figure 19: Closed loop simulation on controller Γ with tracking

Part 4 Simulation with Nonlinear Rocket

Once, we merged the four controllers and did some tries, we observe that the tracking of controller X and Y were pretty good, but controller on Z and Δ could be improve. So I decrease for the Z controller the value R to 0.1 and the weight on Q to 100 to be less aggressive on correction z and increase the Q value of Γ relative to γ to 50.

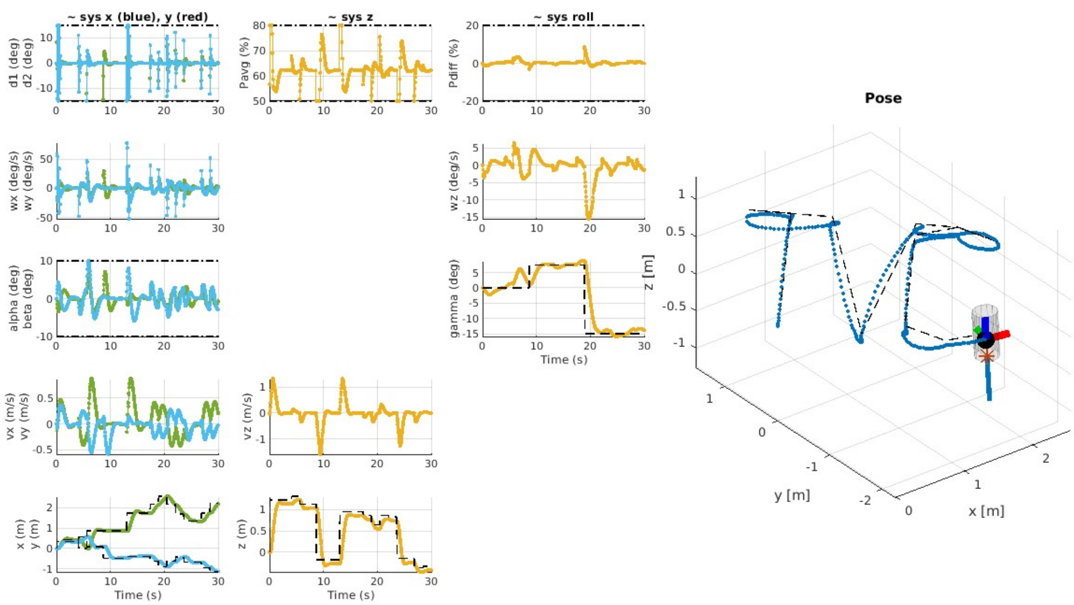

Figure 20: Simulation with Non linear rocket

From the simulation we can observe a little undertracking for the z position due to the reduction on our weight on Q.

Part 5 Offset-Free Tracking

Deliverable 5.1

In this part, we need to add the "disturbance" related to the mass of the rocket, we then obtain a new model with :

$$x_k + 1 = Ax_k + B_u k + B_d d_k$$

$$d_k + 1 = d_k$$

$$y_k = Cx_k + C_d d_k$$

As seen during Week 6, for the offset tracking we need to compute the matrix L relative to the observer using the matrices A,B,C and chosen poles values. We will choose the poles of the system = [0.1, 0.2, 0.3] to converge fast while keeping the error stable enough. We implement the disturbance model with the computation in the controller Z as follow : $A_{bar} = \begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix}$ $B_{bar} = \begin{bmatrix} B \\ 0 \end{bmatrix}$ $C_{bar} = \begin{bmatrix} C \\ 0 \end{bmatrix}$

$$L = -place(A_{bar}^T, C_{bar}^T, [0.1, 0.2, 0.3])^T$$

5.1.1 Simulation with and without mass tracking

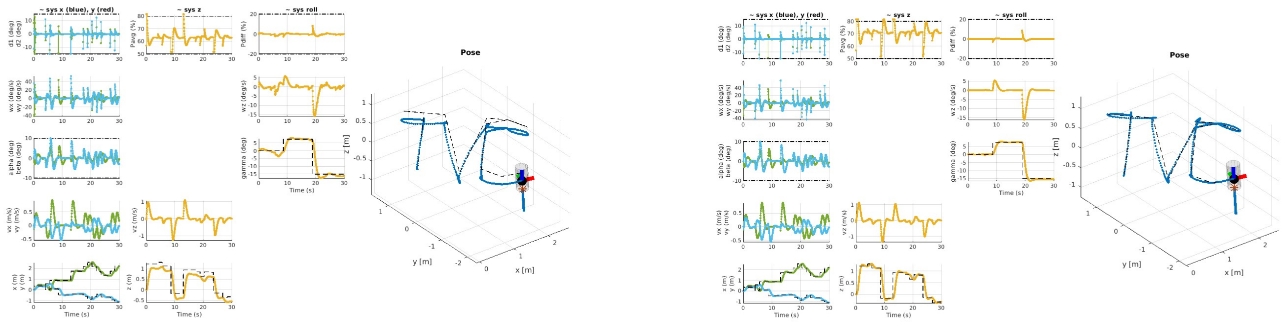

Figure 21: TVC Simulation without and with mass estimation

5.1.2 Simulation of position z

From the plot we can observe the small decrease at the start, related to the mass decrease, then the system respond with an increase of the average throttle to maintain the system at an altitude of 3 meters.

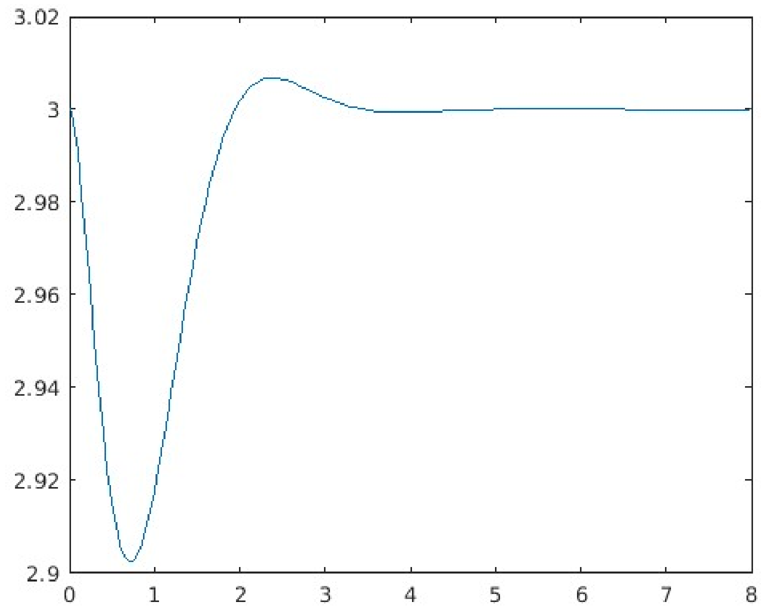

Figure 22: Plot of the position for 8 seconds

Deliverable 5.2

5.2.1 Simulation of position z with mas rate consideration

From the plot below with the drop of mass due to fuel we observe that as before from 0 to 2 seconds, the position z decrease then increase due to the increase of the average throttle but the system then stabilize not around 3 but around 3.02 meters. This little increase of height is due to the mass decreasing continuously during the time. The system compute at each step for a certain mass but at each step the mass decrease. So if the average throttle between two mass is the same, the mass that is lighter will be higher that's why we can observe this offset of 0.02m. A solution is maybe to add a parameter relative to mass, in fact if we put some relations between the average throttle (P_{avg}) and the mass, we can quickly respond to this change of mass by increase rapidly the average throttle.

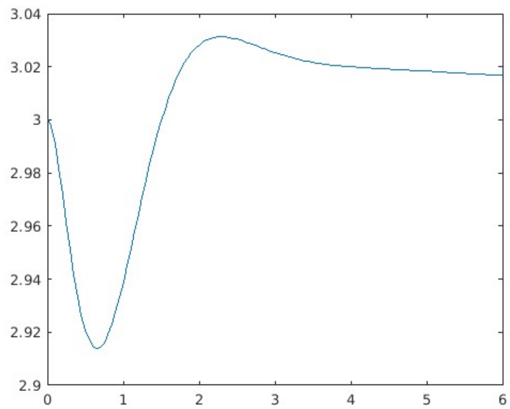

Figure 23: Plot of the position for 8 seconds with mass rate consideration

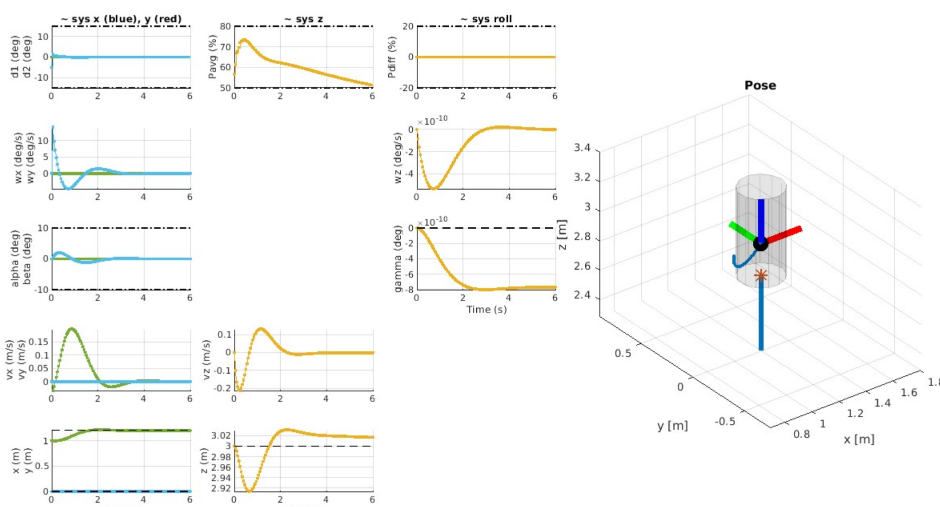

Figure 24: Simulation of the position for 8 seconds with mass rate consideration

5.2.2 Simulation with mass rate consideration

If we consider the mass rate into our simulation for a run time of 30 seconds for example, we can observe that after approximately 6 seconds our simulation stopped because of the limit of the average throttle. Indeed considering a mass drop during the simulation will have as consequence on our system, a drop of the average throttle below the 50% required. We can consider to potentially decrease the limit imposed by the downward acceleration to potentially continue our simulation.

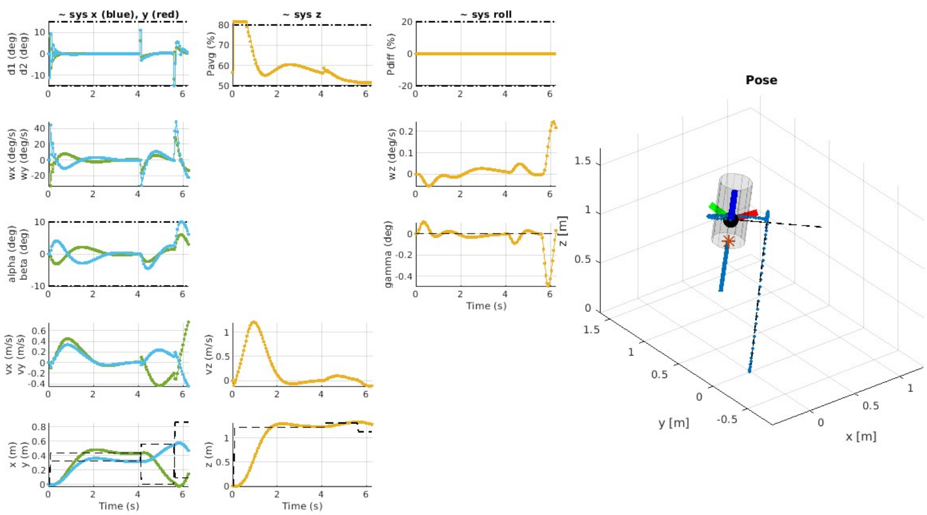

Figure 25: TVC Simulation with mass estimation and mass rate consideration

Part 6 Nonlinear MPC

In this part we will develop a non linear controller for the rocket. The model corresponding to this is the following :

$$u^*(x0) = argmin \sum_{i=1}^{N-1} l(x_i, u_i) + V_f(x_N)$$

$$\text{s.t } x_i + 1 = f(xi, ui)$$

$$g(xi, ui) \leq 0$$

$$h(x_N) \leq 0$$

We will need a discrete function to evaluate our function at the next point. To do that we will use the Runge-Kutta function (RK4.m in folder Deliverable 6.1) develop during exercise 7 to the function f in the class Rocket (rocketfdiscrete = @(x,u) RK4(x,u,DeltaT,rocket)).

Deliverable 6.1

6.1.1 Constraints

We will keep the same constraints on the input ($\delta_1, \delta_2, P_avg, P_diff$) but we will adapt the constraints on the state, we will remove the one α and increase the one on β to 75 degrees. We will also the equality constraints to the state; first with the initialisation and then define the next state as the evaluation of the previous with Runge-Kutta.

6.1.2 Cost matrices and Tuning parameters

To be consistent we all the controllers that we have implemented before, we will keep the same weight on each of them, we will adjust the one on the angle α and β. We also decided not to change the weight on the z-position to not affect the performance of the other parameters specially the x and y. In fact we made a trade off between perfect tracking of z position but less performing tracking of x and y and a good performing of the three of them. We eventually adjust the value of Q_γ to 500 for sharper change of angle. This change is justified by the design that we want to track (step changes in gamma angle), if we had something smoother without suddenly change to track we will keep Q_γ to 50.

So :

$$Q = \text{diag}([50\ 50\ 50\ 1\ 1\ 500\ 1\ 1\ 1\ 100\ 100\ 100])$$

$$R = \text{diag}([1\ 1\ 0.1\ 0.1])$$

We will also choose to reduce the horizon time H to 1, to reduce computational cost.

6.1.3 Terminal Set

To compute our terminal set we linearize our system around the steady state and then discretize it to compute the different matrice A,B C and D and then with LQR we compute the terminal penalty Qf.

6.1.4 Simulation of the non linear system

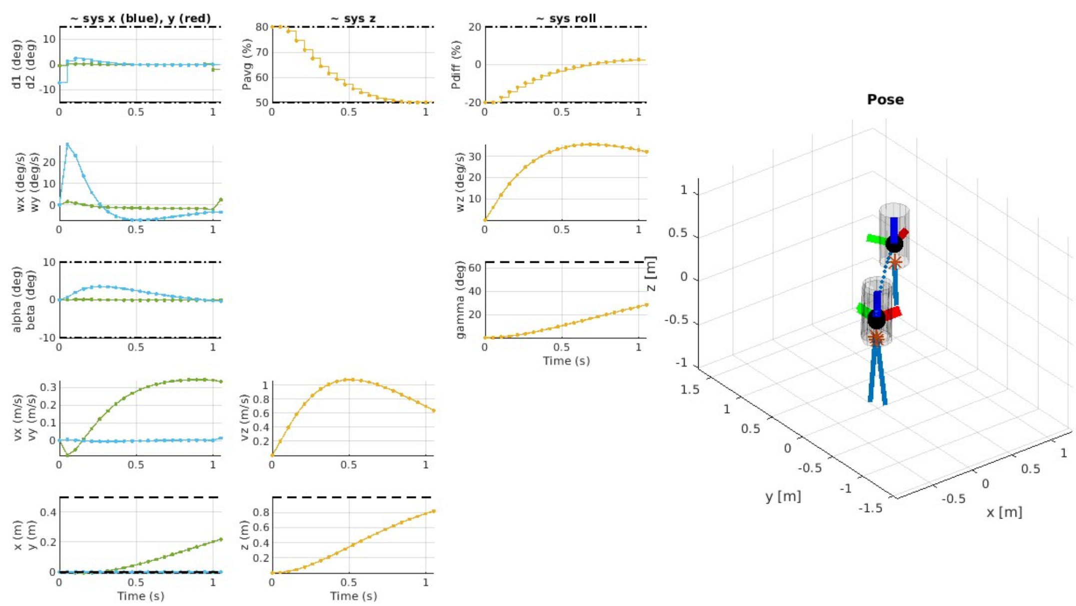

Figure 26: Plot of the open loop for the non linear system

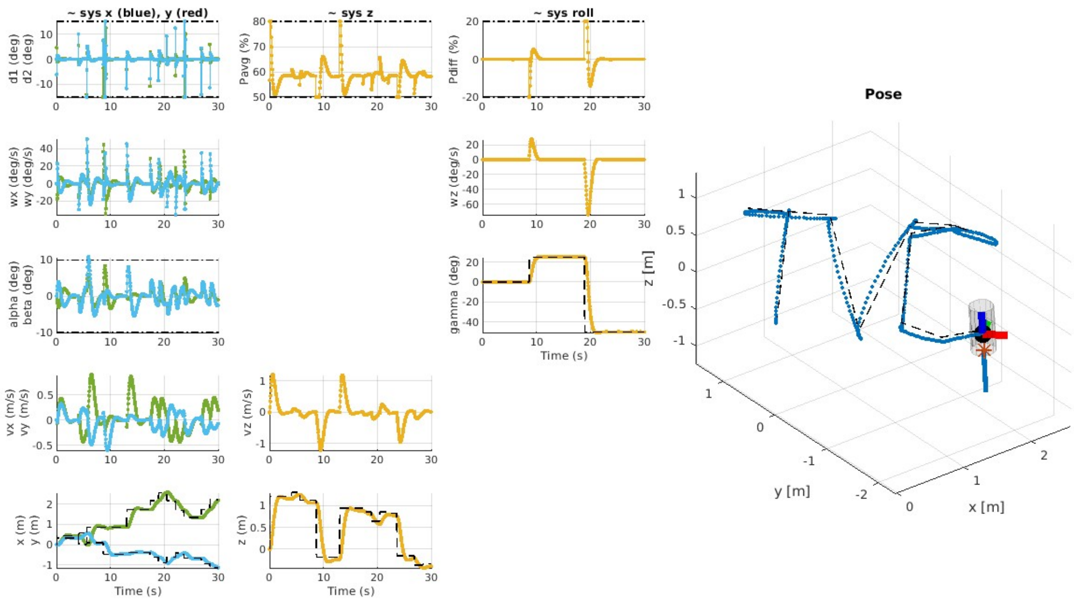

Figure 27: TVC simulation for the non linear system

6.1.5 Pros and cons of non linear system

From a general point of view, most of our system on world is non linear system so the use of non linear system enable a wide range of applications and a more realistic representation. On the other hand, linear system are easiest to develop and use, in our case it also enable to have larger horizon for this model predictive control without exceeding the computational cost. In fact we were not enable to perform an horizon time of 3 second for the non linear system. So if we want have rough idea of the system behavior, the linear system will be enough, if we want more details and predictions on every situations the non linear model is required.

Deliverable 6.2

In this part we want to consider the delay of the system. We will use an Euler approximation (the one developed during the exercise 7 with rocketf-discrete-euler = @(x,u) Euler(x,u,Ts,obj.rocket)) to compute the state considering the delay ($X_{k+delay}$). It will enable us to compute the appropriate input needed at $X_{k+delay}$ and not the one at X_k.

We will also adjust the input by shifting our inputs by the delay (variable: expected delay). We will also use a better initialization with the initial input as the steady state input (us). It micht be help in the convergence of our system and also reduce the computational cost.

6.2.1 Performance of our controller with delay

From our simulation we observe that with a delay of 1 step ($rocket.delay = Ts$) the system become unstable, and with a delay of 4 steps ($rocket.delay = 4 * Ts$) we observe a drop in performance specially on the z position.

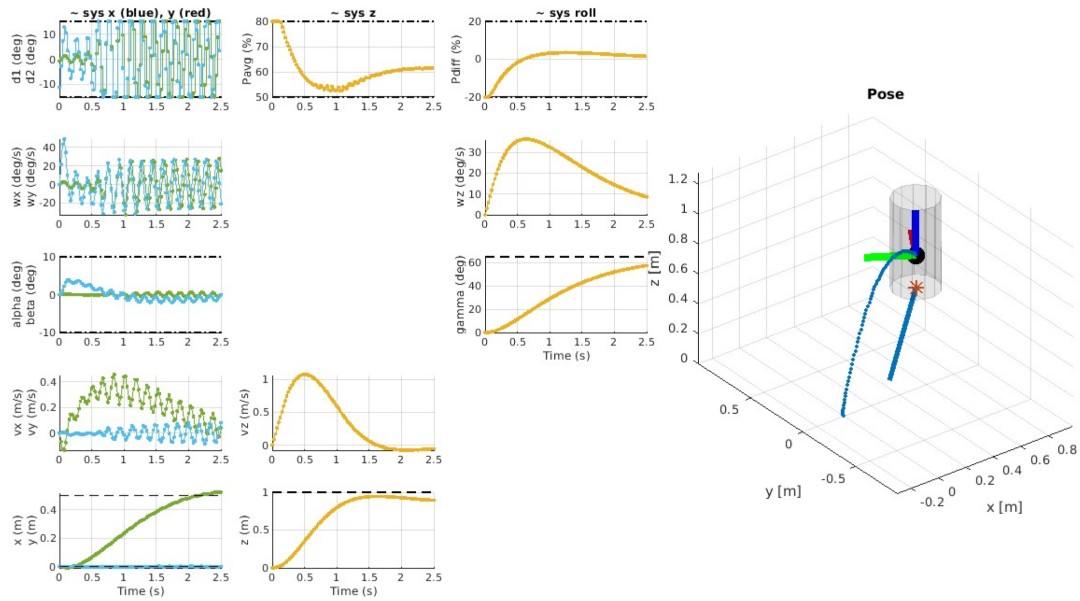

Figure 28: Simulation with delay of 1 step (unstable)

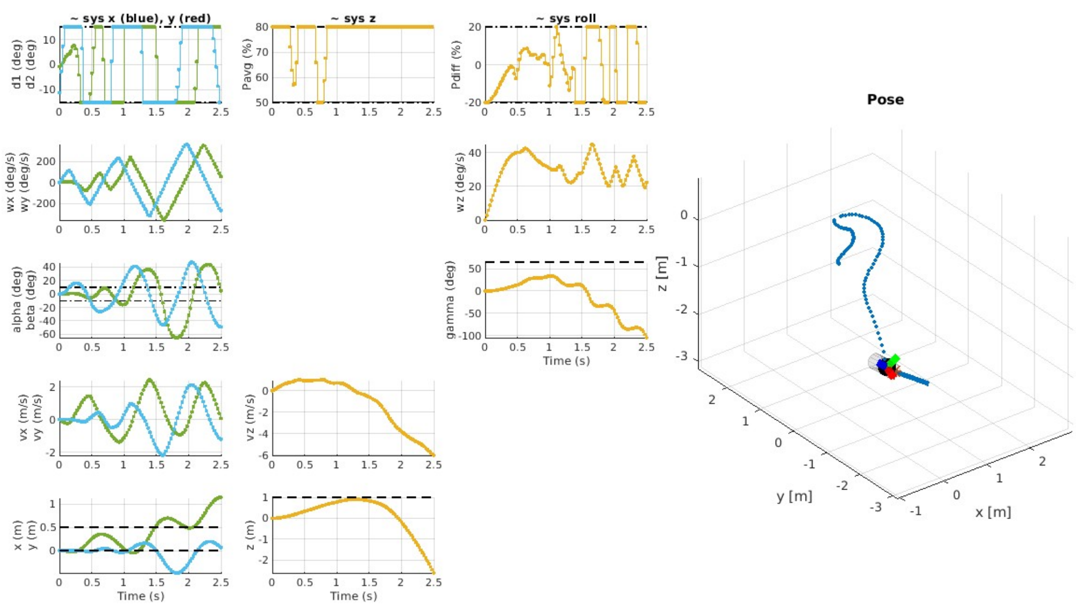

Figure 29: Simulation with delay of 4 steps (drop in performance)

6.2.2 Performance of our controller with delay compensation

We simulate the system with a delay of 8 steps to be sure that the system is unstable and have a drop in performance. We then simulate first with an expected delay of 6 steps (partial delay compensation) and then with a complete delay compensation (expected delay of 8 steps). From our simulation we observe clearly the difference on the performance but also on the stability between partial and complete delay compensation. In fact with a partial delay compensation, the system is unstable and performance are medium, meanwhile with the complete delay compensation our system is stable and the performance is as good as the system with no delay consideration.

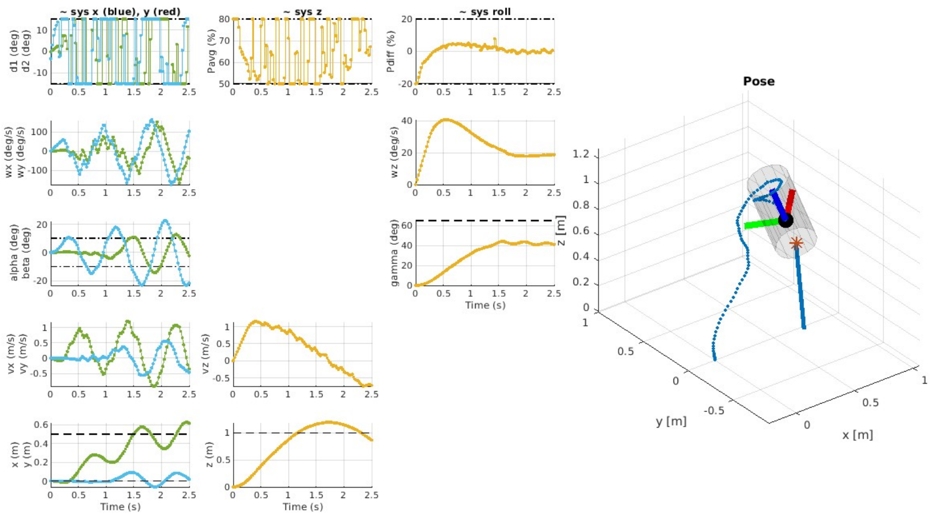

Figure 30: Simulation with partial delay compensation

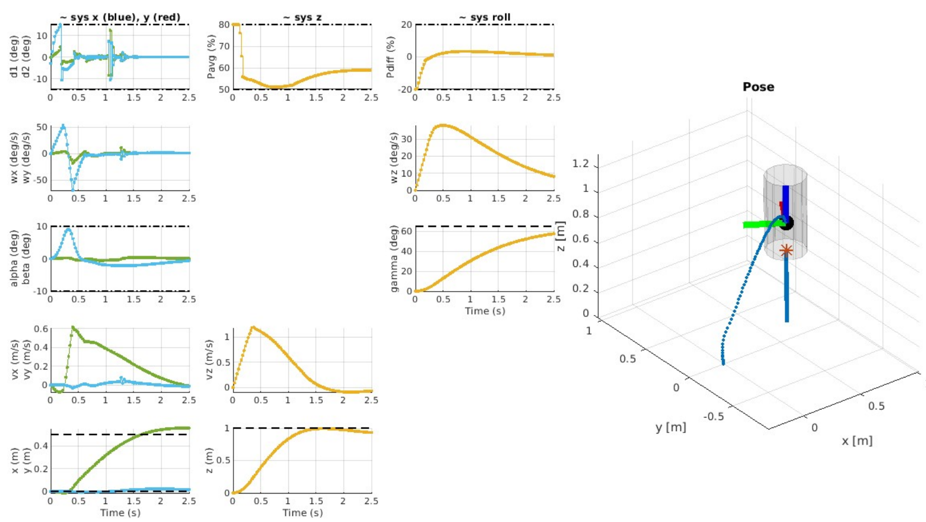

Figure 31: Simulation with complete delay compensation

Conclusion

During this project we develop several system in model predictive control. We start with four linear independent subsystem, we study the horizon time cost matrices and we simulate it with and without tracking of a reference, then we study one linear system taking the different subsystems into consideration and evaluate also the importance of the weight effect on the system. Eventually we consider non linear system and the importance of delay compensation.