Giada Silvestrini

(10659711 – 222787)

Control of Mobile Robots

Project 2023 – 2024

Index:

1. The problem

The objective of the assigned project is the simulation of a car-like robot, which can be described either by a single-track kinematic or dynamic model, with linear or Fiala tier-ground interaction.

During the simulation, the robot should track a prescribed eight-shaped trajectory, exploiting the presence of a complex controller, that is composed of a feedback linearization law and a PI trajectory tracker with velocity feed-forward.

2. Important information

For this project, I have been assigned the following parameters:

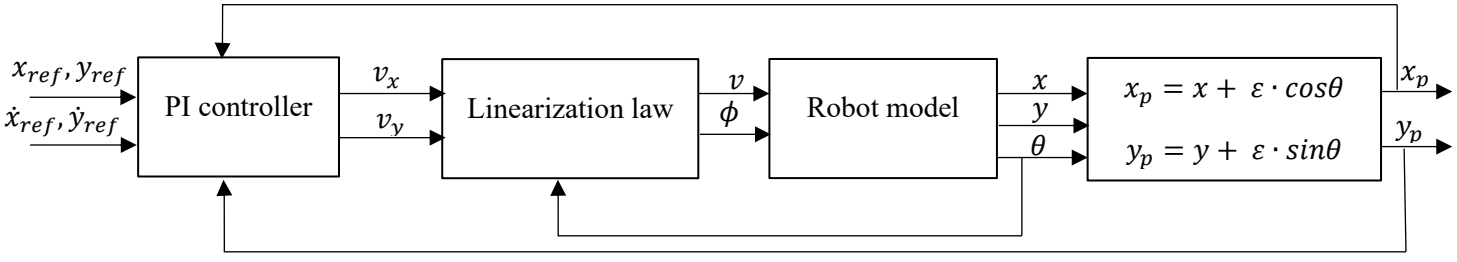
- $m = 1.8$
- $I_z = 0.023$
- $T = 2.1$

To verify what is written underneath, it is possible to run the following files present in the folder:

- car_trajectory_control.launch (to launch the simulation with the dynamic single-track model: in the file car_simulator.yaml set tyre_model=0 to select the linear model, or tyre_model=1 for the fiala model)
- bicycle_trajectory_control.launch (to launch the simulation with the kinematic bicycle model)
- plot_results.py (to plot the data relative to a bag recorded during the simulation)

3. The PI trajectory tracking controller and its tuning procedure

The overall control architecture that I implemented can be schematise with the following block architecture:



In this context, the PI controller is

$$\begin{cases} v_x = \dot{x}_{ref} + K_{px} \cdot (x_{ref} - x_p) + \frac{K_{ix}}{T_s} \cdot T_{ix} \cdot [(x_{ref} - x_p) - (x_{ref\_old} - x_{p\_old})] \\ v_y = \dot{y}_{ref} + K_{py} \cdot (y_{ref} - y_p) + \frac{K_{iy}}{T_s} \cdot T_{iy} \cdot [(y_{ref} - y_p) - (y_{ref\_old} - y_{p\_old})] \end{cases},$$

and the linearization law is

$$\begin{cases} v = v_x \cdot \cos\theta + v_y \cdot \sin\theta \\ \phi = \tan^{-1}\left(\frac{L}{\varepsilon} \cdot \frac{v_y \cdot \cos\theta - v_x \cdot \sin\theta}{v_x \cdot \cos\theta + v_y \cdot \sin\theta}\right) \end{cases}.$$

Once implemented the general design of the controller, I proceeded with the tuning of all the parameters in such a way that the trajectory performed by the simulated robot is as close as possible to the reference one.

The tuning procedure has been done considering, for the robot, a dynamic bicycle model with a linear tier-road interaction. Starting with nominal parameters of K_{px} and K_{py} and momentarily neglecting the integral action ($K_{ix} = K_{iy} = 0$), I applied a trial-and-error approach to discover the values of the two proportional gains that minimize the longitudinal and lateral errors: I both incremented and decremented the two values, also experimenting with asymmetric corrections ($K_{px} \neq K_{py}$) to better understand the behaviour of the robot with respect to different control actions. Using the exact same strategy, I then added the integral action that, as expected, significantly reduced the long-term drift of both the longitudinal and lateral position of the robot.

The resulting parameters of the described tuning procedure are:

- $K_{px} = 0.5$
- $K_{py} = 0.5$
- $K_{ix} = 1.5$
- $K_{iy} = 1.5$
- $T_{ix} = 1.0$
- $T_{iy} = 1.0$
- $T_s = 1.0$

4. The Bicycle Kinematic Model

As first test for the control system, I decided to model the robot with a single-track kinematic model, which can be written as

$$\begin{cases} \dot{x} = v \cdot \cos\theta \\ \dot{y} = v \cdot \sin\theta \\ \dot{\theta} = \frac{v}{L} \cdot \tan\phi \end{cases},$$

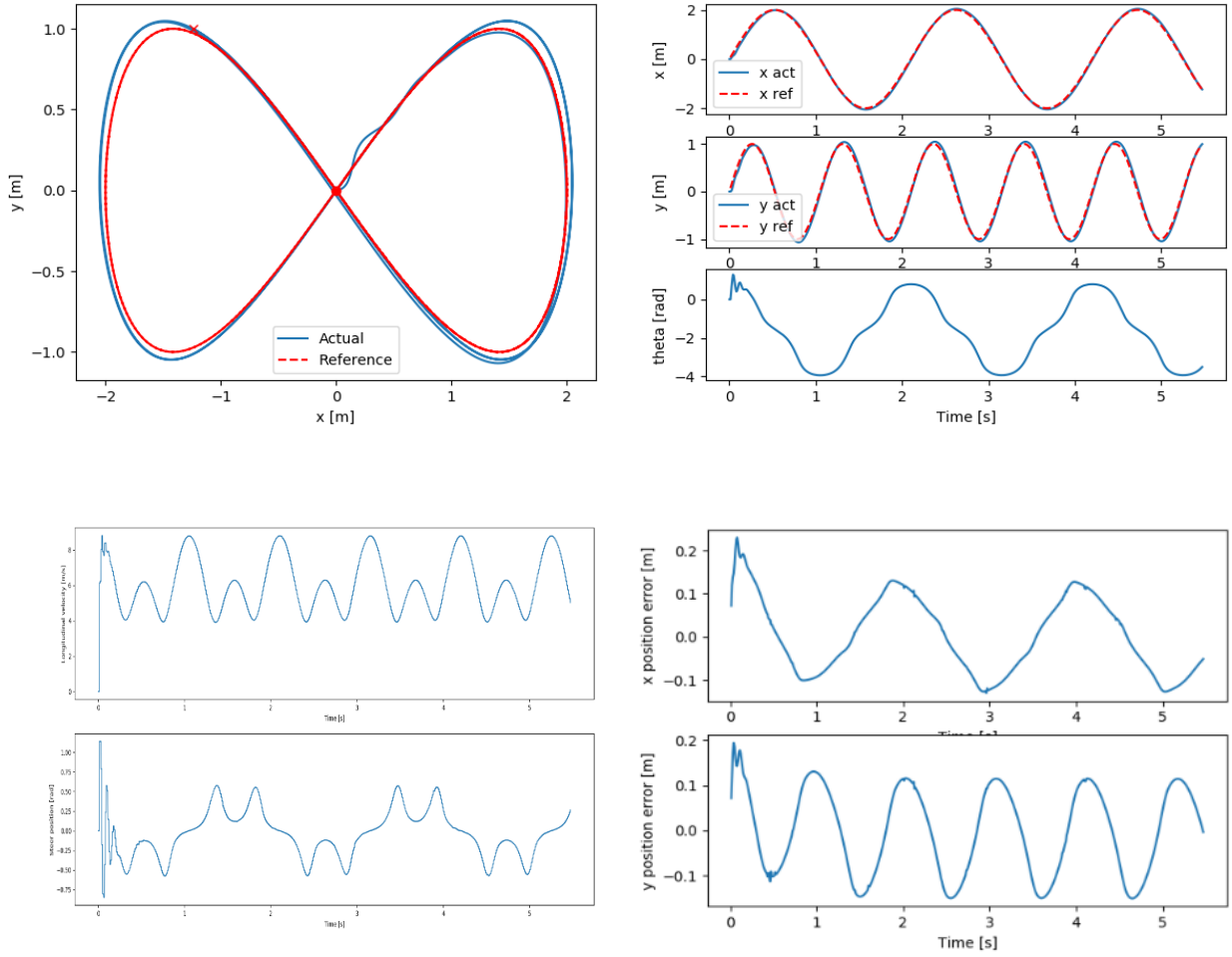
and I set the parameter $L = 0.26m$.

As reference global plan for my control system (structured as explained in the section above), I generated an eight-shaped trajectory that is described by the following equations:

$$\begin{cases} x_{ref} = a \cdot \sin\left(\frac{2\pi}{T}t\right) \\ \dot{x}_{ref} = \frac{2\pi}{T} \cdot a \cdot \cos\left(\frac{2\pi}{T}t\right) \\ y_{ref} = a \cdot \sin\left(\frac{2\pi}{T}t\right) \cdot \cos\left(\frac{2\pi}{T}t\right) \\ \dot{y}_{ref} = \frac{2\pi}{T} \cdot a \cdot \left(\cos\left(\frac{2\pi}{T}t\right)^2 - \sin\left(\frac{2\pi}{T}t\right)^2\right) \end{cases}$$

where $a = 2m$ and $T = 2.1s$.

Executing the test, I obtained the following results:



Analysing these figures, I can see that the reference trajectory is tracked by the controller in a very precise manner: after a short initial phase in which the heading of the robot makes small oscillations, and consequently both the lateral and longitudinal errors reach their maximum values of respectively 20 and 22 centimetres, the controller manages to asymptotically stabilize the closed loop system.

5. The Single-Track Dynamic Model

For the following tests, instead of modelling the robot as a kinematic bicycle I implemented its dynamic version, that is:

$$\begin{cases} \dot{r} = \dfrac{a \cdot F_{yf} - b \cdot F_{yr}}{I_z} \\ \dot{\beta} = \dfrac{(F_{yf} + F_{yr}) \cdot \cos(\beta)}{m \cdot v} - r \\ \dot{x} = v \cdot \cos\theta \\ \dot{y} = v \cdot \sin\theta \\ \dot{\theta} = \dfrac{v}{L} \cdot \tan\phi \end{cases}$$

where the parameters are:

- $I_z = 0.023$
- $a = 0.14$
- $b = 0.12$
- $m = 1.8$

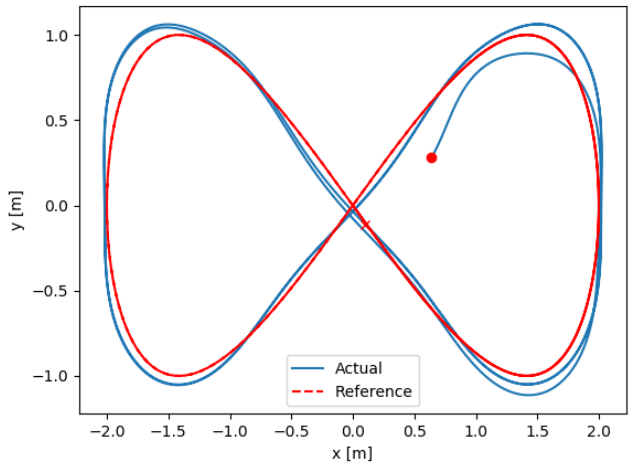
The lateral forces F_{yf} and F_{yr} depend on the chosen road-tire interaction model: examples of the possible models are the Linear (with or without saturation) and the Fiala (with or without saturation) ones.

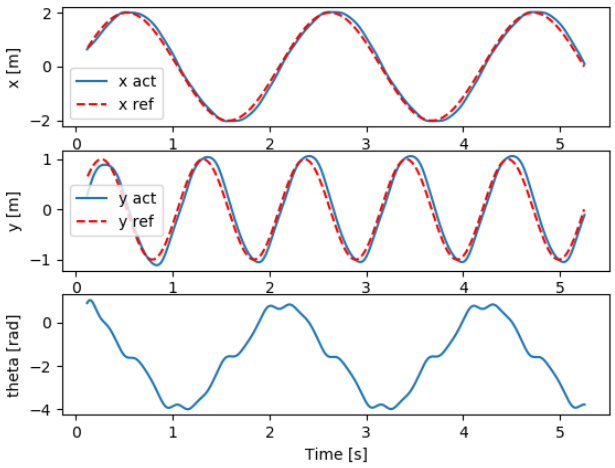
4.1 Linear Road-Tire model

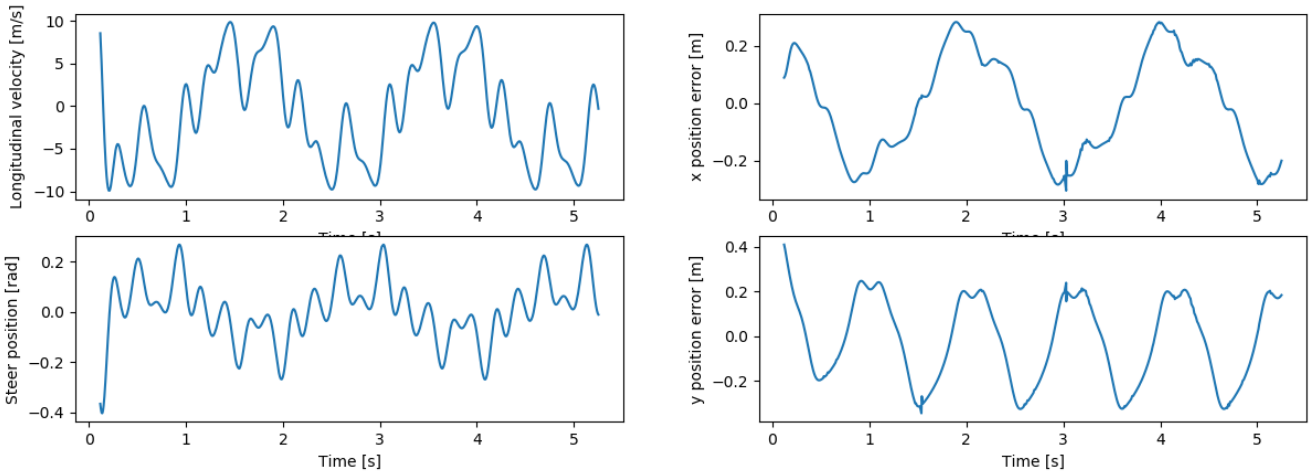
In this case, the two forces are $\begin{cases} F_{yf} = -C_f \cdot \alpha_f \\ F_{yr} = -C_r \cdot \alpha_r \end{cases}$, with:

- $C_f = 50$
- $C_r = 120$
- $\alpha_r = \beta - b \cdot \frac{r}{v}$
- $\alpha_f = \beta + a \cdot \frac{r}{v} - \delta$

Running the experiment, I get the following results:







From these images it is possible to observe that the controller still performs a good tracking of the eight-shaped trajectory described above, even though the results are less accurate with respect to the ones obtained with the kinematic model. In particular, we immediately note a regular but "shaky" evolution of the longitudinal and lateral errors (the maximum values are also larger with respect to the previous case: almost 30 and 40 centimetres respectively), and, moreover, the lateral dynamics seems to suffer a slight delay (it can be noted by comparing the reference and the actual $y(t)$): it may be caused by the high frequency oscillation of both the control signals (the steering position and the longitudinal velocity are not smooth at all).

Despite these issues, the controller is still able to perform a closed loop asymptotic stabilization of the system, and results tracking the reference trajectory sufficiently well.

4.2 Fiala Road-Tire Model

According to the Fiala model, the two lateral forces are described by:

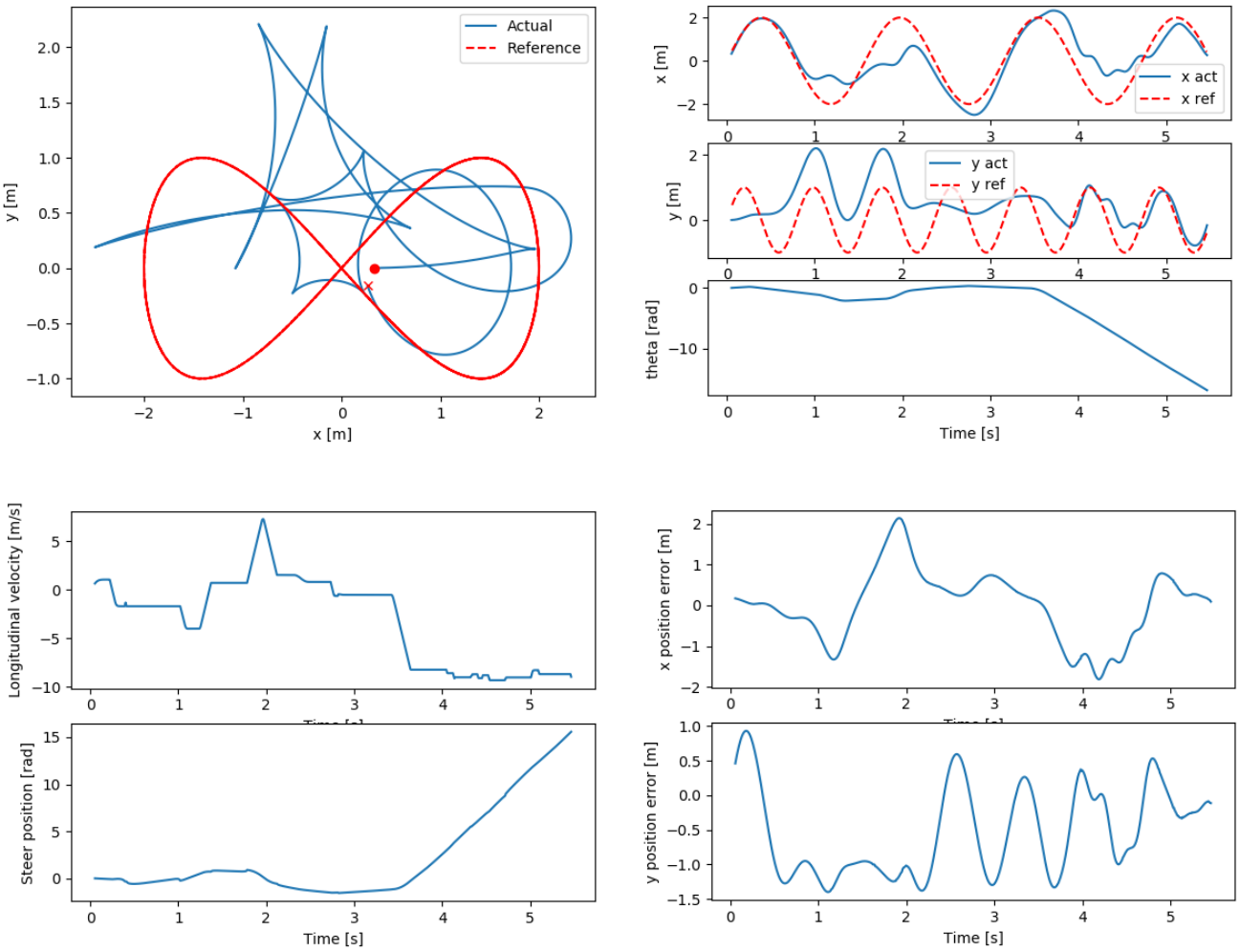
$$\begin{cases} F_{yf} = \begin{cases} C_f \cdot z_f \cdot (-1 + \frac{|z_f|}{z_{f\_sl}} - \frac{{z_f}^2}{3 \cdot {z_{f\_sl}}^2}), & |z_f| < z_{f\_sl} \\ -\mu \cdot F_{z\_f} \cdot sign(\alpha_f), & |z_f| \geq z_{f\_sl} \end{cases} \\ F_{yr} = \begin{cases} C_r \cdot z_r \cdot (-1 + \frac{|z_r|}{z_{r\_sl}} - \frac{{z_r}^2}{3 \cdot {z_{r\_sl}}^2}), & |z_r| < z_{r\_sl} \\ -\mu \cdot F_{z\_r} \cdot sign(\alpha_r), & |z_r| \geq z_{r\_sl} \end{cases} \end{cases}$$

with the additional (with respect to the ones already listed) parameters:

- $F_{z\_f} = m \cdot g \cdot \frac{b}{L}$
- $F_{z\_r} = m \cdot g \cdot \frac{a}{L}$
- $\mu = 0.385$
- $z_f = \tan(\alpha_f)$
- $z_r = \tan(\alpha_r)$
- $z_{f\_sl} = 3 \cdot \mu \cdot \frac{F_{z\_f}}{C_f}$

- $z_{r\_sl} = 3 \cdot \mu \cdot \frac{F_{z\_r}}{C_r}$

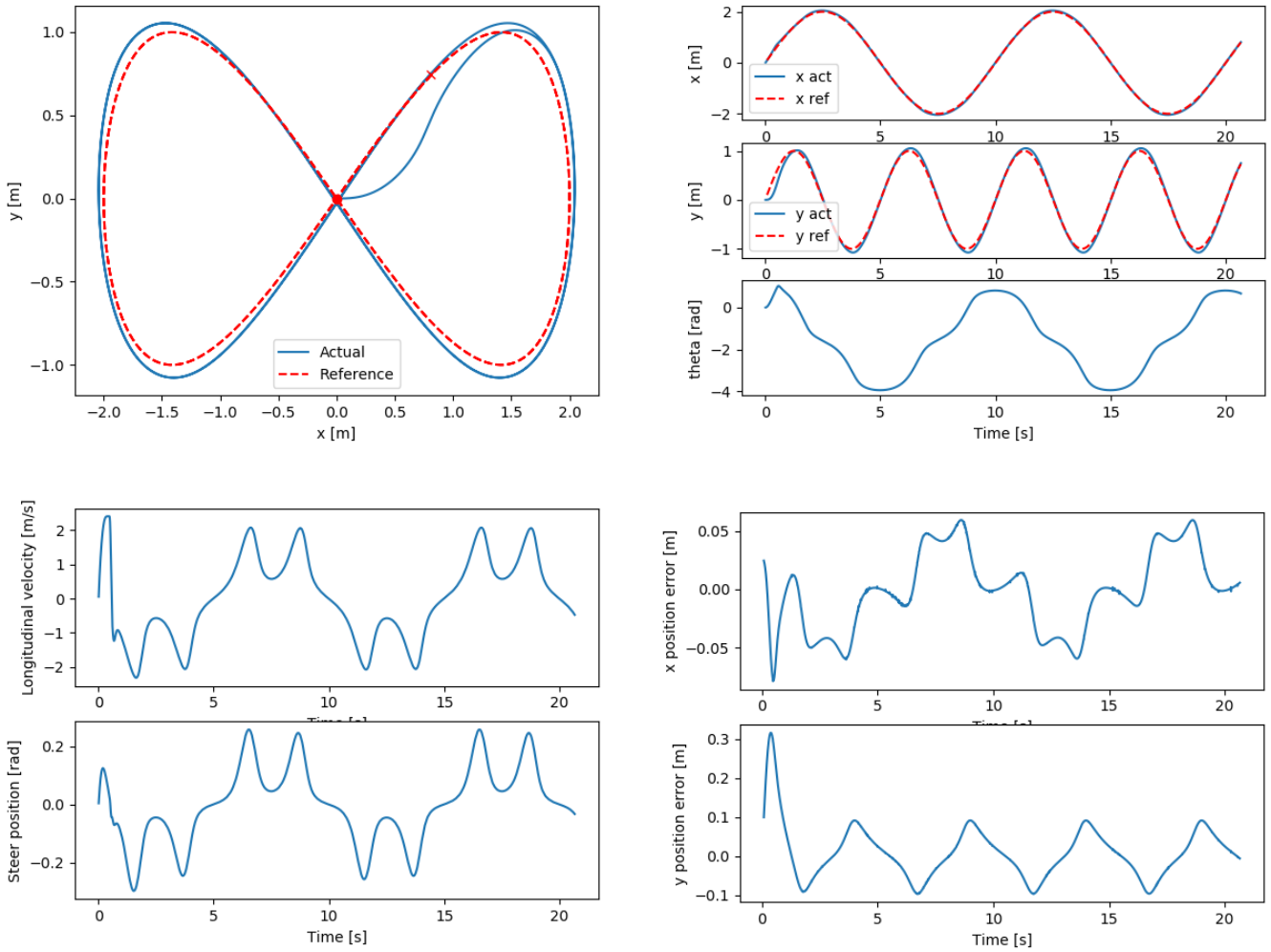
The very same experiment, this time with $T = 0.75 \cdot 2.1 = 1.575s$, and the Fiala Road-Tire model yields the following results:



In this case, it is clear and unmistakable from each of the 4 images that the controller is not able to track the reference trajectory. Despite the huge number of values and combinations that I tried for the controller gains ($K_{px}, K_{py}, K_{ix}, and\ K_{iy}$), no matter how small or how big they were, I did not manage to find a set able to stabilize the system, when the required period is this small.

The problem, in fact, seems to be the working frequency, because I noticed that, setting a higher value of T (the threshold is around 6.5 seconds), and therefore making the robot move in a slower way, the controller (with the values of the gains used up to now) is perfectly able even with this model to track the required trajectory.

In particular, the higher is the value of T, the better is the performance of the controller. For example, if I set $T = 10$, the above results completely change into the following:



6. Conclusions

Through this project, I had the possibility of implementing and deeply understanding the concepts seen during the course, as well as familiarising with a complex tool like ROS. For these reasons, despite representing a proper challenge in some moments (due to my computer science academic background), I enjoyed the experience and found it very useful.