Homework 1

Valentina Giannotti

April 28, 2024

1 Question 1

State whether each of following distributions are involutive or not, and briefly justify your answer. If possible, find the annihilator for each distribution.

$$\textbf{a.}\left\{\begin{bmatrix} 3\cdot x_1 \\ 0 \\ -1 \end{bmatrix}\right\} \quad \textbf{b.}\left\{\begin{bmatrix} 1 \\ 0 \\ x_2 \end{bmatrix}\begin{bmatrix} 0 \\ -\alpha \\ x_1 \end{bmatrix}\right\} \quad \textbf{c.}\left\{\begin{bmatrix} 2\cdot x_3 \\ -1 \\ 0 \end{bmatrix}\begin{bmatrix} x_2 \\ x_1 \\ 1 \end{bmatrix}\right\}$$

1a

By definition, **any 1-dimensional distribution is involutive** since $[f(x), f(x)] = 0$, and the zero vector belongs to any distribution by default.
If the distribution as spanned by columns, as here, the annihilator is identified by a set of co-vectors such that $\omega^* F(x) = 0$

$$\begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix}\begin{pmatrix} 3x_1 \\ 0 \\ -1 \end{pmatrix} = 0 => 3w_1x_1 - w_3 = 0$$

Remember that $dim(\Delta) + dim(\Delta^{\perp}) = n$. In this case $dim(\Delta) = 1$ and n=3 and $dim(\Delta^{\perp})$ is 2. So we have to find two co-vectors.

$$\omega_1^* = \begin{bmatrix} \alpha & \beta & 3\alpha x_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 3x_1 \end{bmatrix}$$

$\alpha = 1$, $\beta = 0$

$$\omega_2^* = \begin{bmatrix} \alpha & \beta & 3\alpha x_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$\alpha = 0$, $\beta = 1$

1b

$$\textbf{b.}\left\{\begin{matrix} 1 & 0 \\ 0 & -9 \\ x_2 & x_1 \end{matrix}\right\}$$

First It is necessary to define the distribution: $\Delta = span\{f_1(x), f_2(x)\}$

$$f_1(x) = \begin{pmatrix} 1 \\ 0 \\ x_2 \end{pmatrix} \; f_2(x) = \begin{pmatrix} 0 \\ -9 \\ x_1 \end{pmatrix}$$

$$F(x) = \begin{pmatrix} 1 & 0 \\ 0 & -9 \\ x_2 & x_1 \end{pmatrix} \; rank(F) = 2 \forall x \in U \Rightarrow dim(\Delta(x)) = 2 \forall x \in U;$$

To verify if the distribution is involutive, we need to understand if it is singular. Recall that a distribution is non-singular if $\dim(\Delta(x)) = d$, $\forall x \in U$. In this case the rank is maximum so **we can conclude that it is non-singular**. At this point, we can compute Lie Bracket (in this case, because $d = 2$ there's only one) and verify if all the vector obtained are a linear combination of the other.

$$[f_1(x), f_2(x)] = \frac{\partial f_2(x)}{\partial x} f_1(x) - \frac{\partial f_1(x)}{\partial x} f_2(x) =$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ x_2 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ -9 \\ x_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 10 \end{pmatrix}$$

$$F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -9 & 0 \\ x_2 & x_1 & 10 \end{pmatrix} \Rightarrow det(F) = -90$$

The rank of matrix F is 3 so the Lie bracket of the two vector isn't a combination of the other two, so **the distribution is not involutive**.
To figure out the distribution's **annihilator**, we must determine the vectors whose inner product with the vector is zero:

$$\begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} * F = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -9 \\ x_2 & x_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} w_1 + w_3 x_2 = 0 \\ -9w_2 + w_3 x_1 = 0 \end{cases}$$

$$\Rightarrow \begin{cases} w_1 = -w_3 x_2 \\ w_2 = w_3 x_1/9 = 0 \end{cases}$$

The annihilator is in the form:

$$\omega^* = \begin{bmatrix} -\alpha x_2 & \alpha x_1/9 & \alpha \end{bmatrix}$$

α is chosen like this: $\alpha = 9$, so it becomes:

$\Delta^{\perp}(x) = span\{w^*\} \; dim(\Delta^{\perp}(x)) = 1 \Rightarrow \begin{bmatrix} -9x_2 & x_1 & 9 \end{bmatrix}$

1c

$$\textbf{c.}\left\{\begin{array}{cc} 2*x_3 & x_2 \\ -1 & x_1 \\ 0 & 1 \end{array}\right\}$$

Define the distribution: $\Delta = span\{f_1(x), f_2(x)\}$

$$f_1(x) = \begin{pmatrix} 2*x_3 \\ -1 \\ 0 \end{pmatrix} \quad f_2(x) = \begin{pmatrix} x_2 \\ x_1 \\ 1 \end{pmatrix}$$

$$F(x) = \begin{pmatrix} 2x_3 & x_2 \\ -1 & x_1 \\ 0 & 1 \end{pmatrix} \quad rank(F) = 2 \Rightarrow dim(\Delta(x)) = 2;$$

It isn't singular $\forall x \neq 0$

Lie brackets between the two vectors:

$$[f_1(x), f_2(x)] = \frac{\partial f_2(x)}{\partial x} f_1(x) - \frac{\partial f_1(x)}{\partial x} f_2(x) =$$

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2x_3 \\ -1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_2 \\ x_1 \\ 1 \end{pmatrix} = \begin{pmatrix} -3 \\ 2x_3 \\ 0 \end{pmatrix}$$

$$F(x) = \begin{pmatrix} 2x_3 & x_2 & -3 \\ -1 & x_1 & 2x_3 \\ 0 & 1 & 0 \end{pmatrix}$$

$\Rightarrow det(F) = 3 - 4x_3^2$ is equal to 0 when $x_3 = \sqrt{\frac{3}{4}}$

At this point we can distinguish two cases:

- $x_3 \neq \sqrt{\frac{3}{4}}$ and the rank=3 $\Rightarrow$ non-involutive
- $x_3 = \sqrt{\frac{3}{4}}$ and the rank=2 $\Rightarrow$ involutive

To compute the annihilator of the distribution we must compute the base of the vectors whose product to the vector of the distribution produce the null vector:

$$\begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} \begin{pmatrix} 2x_3 & x_2 \\ -1 & x_1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow \begin{cases} 2w_1x_3 - w_2 = 0 \\ w_1x_2 + w_2x_1 + w_3 = 0 \end{cases}$$

The annihilator is in the form: $\left[\begin{array}{ccc} \alpha & 2x_3\alpha & -\alpha(2x_1x_3+x_2) \end{array}\right]$

It can be particolarized by the choice $\alpha = 1$.

$\omega^* = \left[\begin{array}{ccc} 1 & 2x_3 & -2x_1x_3 - x_2 \end{array}\right]$

2 Question 2

Consider an omnidirectional mobile robot having 3 Mecanum wheels placed at the vertices of an equilateral triangle, each oriented in the direction orthogonal to the bisectrix of its angle. Let x and y be the Cartesian coordinates of the center of the robot, θ the vehicle orientation and α, β, γ represent the angle of rotation of each wheel around its axis. Also, denote by r the radius of the wheels and by l the distance between the centre of the robot and the centre of each wheel. This mechanical system is subject to the following Pfaffian constraints, where $q = [x, y, \theta, \alpha, \beta, \gamma\ ]$.
Compute a kinematic model of such an omnidirectional robot and show whether this system is holonomic or not.

$$A^T = \begin{pmatrix} \frac{\sqrt{3}\,\cos(\theta)}{2} - \frac{\sin(\theta)}{2} & \sin(\theta) & -\frac{\sin(\theta)}{2} - \frac{\sqrt{3}\,\cos(\theta)}{2} \\ \frac{\cos(\theta)}{2} + \frac{\sqrt{3}\,\sin(\theta)}{2} & -\cos(\theta) & \frac{\cos(\theta)}{2} - \frac{\sqrt{3}\,\sin(\theta)}{2} \\ l & l & l \\ r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{pmatrix}$$

In order to define whether the system subject to the following Pfaffian constrain is holonomic or not, we must know that there is at least one non-holonomic constraint that makes such a system non-holonomic. To understand if the Pfaffian constraint $A^T(q)\dot{q} = 0$ is integrable (holonomic) we have to compute a controllability analysis of the Kinematic model of the omnidirectional robot. $\dot{q} = G(q)u$.
The matrix G is obtained by finding the matrix of co-vectors that make the matrix A^T null. It was calculated with the help of the software MATLAB with the function $null(A^T)$:

$$G = \begin{pmatrix} -\frac{r(3\cos(\theta)-3\sqrt{3}\sin(\theta))}{3(3\sqrt{3}\cos(\theta)^2+3\sqrt{3}\sin(\theta)^2)} & -\frac{2r\sin(\theta)}{3(\cos(\theta)^2+\sin(\theta)^2)} & \frac{r(3\cos(\theta)+3\sqrt{3}\sin(\theta))}{3(3\sqrt{3}\cos(\theta)^2+3\sqrt{3}\sin(\theta)^2)} \\ -\frac{r(3\sin(\theta)+3\sqrt{3}\cos(\theta))}{3(3\sqrt{3}\cos(\theta)^2+3\sqrt{3}\sin(\theta)^2)} & \frac{2r\cos(\theta)}{3(\cos(\theta)^2+\sin(\theta)^2)} & \frac{r(3\sin(\theta)-3\sqrt{3}\cos(\theta))}{3(3\sqrt{3}\cos(\theta)^2+3\sqrt{3}\sin(\theta)^2)} \\ -\frac{r}{3l} & -\frac{r}{3l} & -\frac{r}{3l} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

In Order to prove the controllability of this system, we have to compute the accessibility distribution Δ_A of the matrix G. Indeed, the system is controllable only if the accesibility rank condition is satisfy $dim(\Delta_A) = n$. It's important to remember that accessibility distribution is the distribution generated by the vector fields and all the Lie bracket that can be generated by these vector fields. For this reason the accessibility distribution matrix was built recursively, calculating all the Liebracket of each vector and adding to the matrix only the Liebracket that increased the rank of the matrix.
Computation of the accessibility distribution Δ_A of the matrix G:

$$\Delta_A = [g_1, g_2, g_3, [g_1, g_2], [g_2, g_3], [g_1, g_3], ...]$$

$$\Delta_A = \begin{pmatrix} -\frac{2r\cos(\theta+\frac{\pi}{6})}{3} & -\frac{2r\sin(\theta)}{3} & \frac{2r\sin(\theta+\frac{\pi}{3})}{3} & \frac{2\sqrt{3}r^2\sin(\theta+\frac{\pi}{3})}{9l} & \frac{2\sqrt{3}r^2\sin(\theta)}{9l} \\ -\frac{2r\cos(\theta-\frac{\pi}{3})}{3} & \frac{2r\cos(\theta)}{3} & -\frac{2r\cos(\theta+\frac{\pi}{3})}{3} & -\frac{2\sqrt{3}r^2\cos(\theta+\frac{\pi}{3})}{9l} & -\frac{2\sqrt{3}r^2\cos(\theta)}{9l} \\ -\frac{r}{3l} & -\frac{r}{3l} & -\frac{r}{3l} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

The rank of this matrix is $rank = 5$ so the system is not controllable, some model-configuration are prevented and the system has some holonomic constraints but not all. Therefore, it can be concluded that the system is **non-holonomic**.

3 Question 3

Implement via software the path planning algorithm for a unicycle based on a cubic Cartesian polynomial. Plan a path leading the robot from the configuration $q_i = [x_i, y_i, \theta_i]^T = [0, 0, 0]^T$, to a random configuration $qf = [x_f, y_f, \theta_f]^T$ generated automatically by the code such that $||q_f - q_i|| = 1$. Then, determine a timing law over the path to satisfy the following velocity bounds $|v(t)| \leq 2m/s$ and $|w(t)| \leq 1rad/s$. [Hint: For the final configuration, use the command rand(1,3): save the result in a vector and divide it by its norm.]
Using the cubic Cartesian polynomial formulas, it is possible to derive the complete path from q_i to q_f.

The chosen motion law is characterized by a cubic polynomial whose trajectory goes from $s_i = 0$ to $s_f = 1$, with $t_i = 0$ and $t_f = 6$ seconds.
The cubic polynomial is computed like this:

$$\begin{aligned} s(t) &= a_3 t^3 + a_2 t^2 \\ \dot{s}(t) &= 3a_3 t^2 + 2a_2 t \\ \ddot{s}(t) &= 6a_3 t + 2a_2 \end{aligned}$$

Below is shown the path planning algorithm for a unicycle based on a cubic Cartesian polynomial:

$$\begin{aligned} x &= s^3 * xf - (s-1)^3 * xi + \alpha_x * s^2 * (s-1) + \beta_x * s * (s-1)^2; \\ y &= s^3 * yf - (s-1)^3 * yi + \alpha_y * s^2 * (s-1) + \beta_y * s * (s-1)^2; \\ x' &= 3 * s^2 * xf - 3 * (s-1)^2 * xi + \alpha_x * s * (3 * s - 2) + \beta_x * (3 * s - 1) * (s-1); \\ y' &= 3 * s^2 * yf - 3 * (s-1)^2 * yi + \alpha_y * s * (3 * s - 2) + \beta_y * (3 * s - 1) * (s-1); \\ x'' &= 6 * s * xf - 6 * (s-1) * xi + \alpha_x * (6 * s - 2) + \beta_x * (6 * s - 4); \\ y'' &= 6 * s * xf - 6 * (s-1) * xi + \alpha_y * (6 * s - 2) + \beta_y * (6 * s - 4); \end{aligned}$$

It is important to note that q_i was provided by the trace while q_f was randomly extracted. Below are the results with $q_f = [0.8190, 0.5670, 0.0874]$
With:

$$\begin{aligned} \alpha_x &= k * \cos\theta_f - 3 * xf \\ \alpha_y &= k * \sin\theta_f - 3 * yf \\ \beta_x &= k * \cos\theta_i + 3 * xi \\ \beta_y &= k * \sin\theta_i + 3 * yi \\ k &= 3; \end{aligned}$$

3a Result

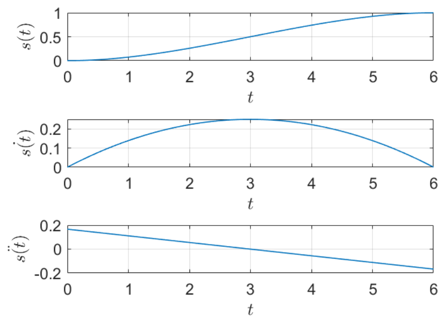

Figure 1: Time law before time scaling

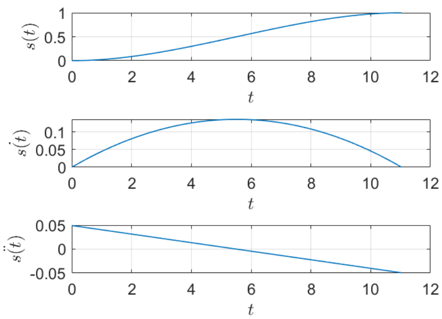

Figure 2: Time law after time scaling

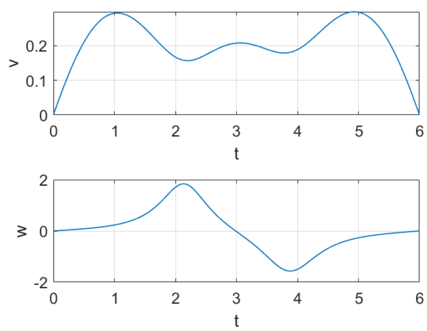

Figure 3: Value of v and w

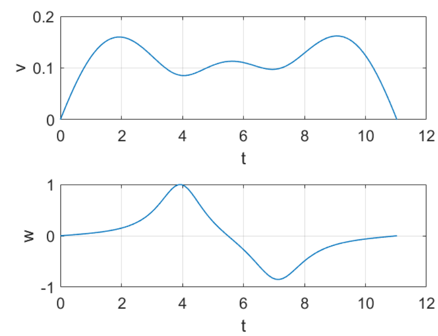

Figure 4: Value of v and w after time scaling

In order satisfies the speed limits on v(t) heading velocity and w(t) angular velocity it necessary scaling a timing law. A proportional motion scale has been implemented to ensure compliance with predetermined speed limits. Thanks to this approach, the system can be dynamically adjusted to changes in movement speed while meeting safety requirements.

When the maximum allowed speed is exceeded, the trajectory is scaled. The comparison between the highest speeds reached and the highest allowed speeds has been determined. The trajectory will be reduced guaranteeing that the constraints for both speeds are satisfied and taking into account the worst case or the greater violation of the speed between the two. Subsequently, the time increases proportionally . This means that the higher the ratio, that is, the more the speed exceeds the limit, the more time increases, ensuring an adequate slowdown of the trajectory.

The figures indicate that both speed limits are now being observed throughout the trajectory.

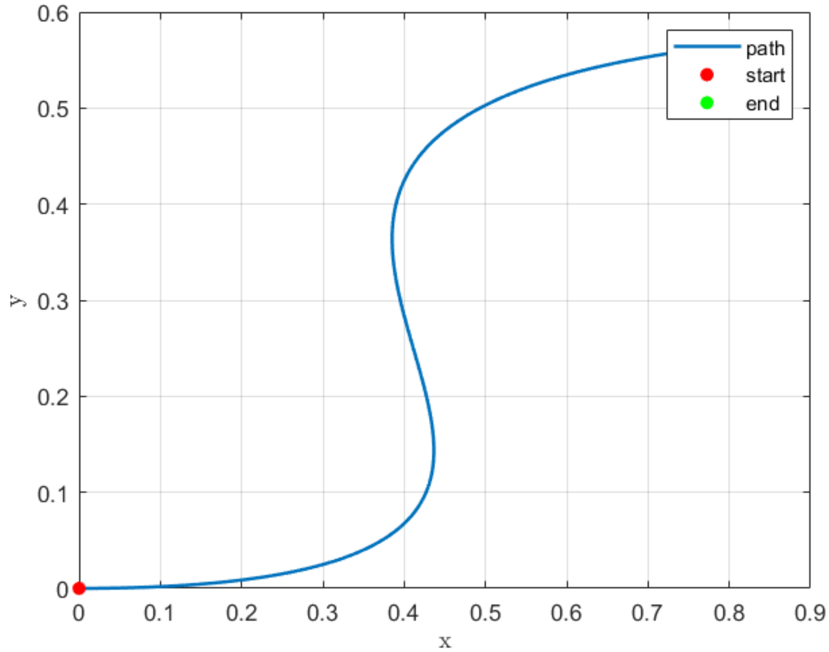

Figure 5: path

4 Question 4

Given the trajectory in the previous point, implement via software an input/output linearization control approach to control the unicycle's position. Adjust the trajectory accordingly to fit the desired coordinates of the reference point B along the sagittal axis, whose distance to the wheel's center it is up to you. A high-level abstraction explanation of the implemented programming logic is given here. Run the matlab program and the sdirectly to view the full operation of the code.

A trajectory tracking technique based on input-output linearization will then be employed. This systematic approach is based on linearization by feedback. In the context of the unicycle, the system outputs, which represent the Cartesian coordinates of a point B (located along the sagittal axis of the unicycle, at a distance $|b|$ from the point of contact of the wheel with the ground) are formulated as follows:

$$\begin{cases} y_1 = x + b\cos\theta \\ y_2 = y + b\sin\theta \end{cases}$$

Where b has been chosen as b =0.05.
Prime derivatives of y_1 and y_2:

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -b\sin\theta \\ \sin\theta & b\cos\theta \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} = T(\theta) \begin{pmatrix} v \\ \omega \end{pmatrix}$$

As indicated, to calculate the desired speeds v and ω, a Matlab function has been inserted that involves the inversion of the matrix T.

In this way, we obtain an input-output linearization through feedback that allows us to use a linear controller of this type to ensure the zero convergence of the tracking error.

$$\begin{cases} u_1 = \dot{y}_{1d} + k_1(y_{1d} - y_1) \\ u_2 = \dot{y}_{2d} + k_2(y_{2d} - y_2) \end{cases}$$

Where $k_1 = 8$ and $k_2 = 8$.

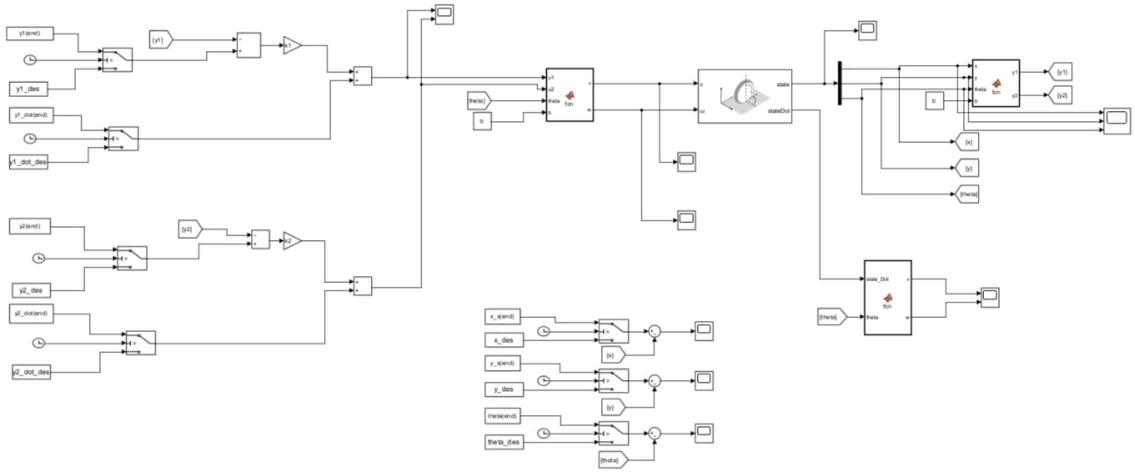

Figure 6: Simulink model

It is important to remember that the values of K1 and K2 were chosen in relation to the trajectory in the example. For more complex trajectories it is necessary to choose lower K1 and K2 values.

A switching system has been implemented to prevent the trajectory from diverging. Since we can only control the trajectory during certain time intervals, we have taken the approach of storing the last controlled value as a constant. This ensures us a correct tracking of the trajectory even outside the periods when we are able to control it directly.

4a Result

Evolution of the states:

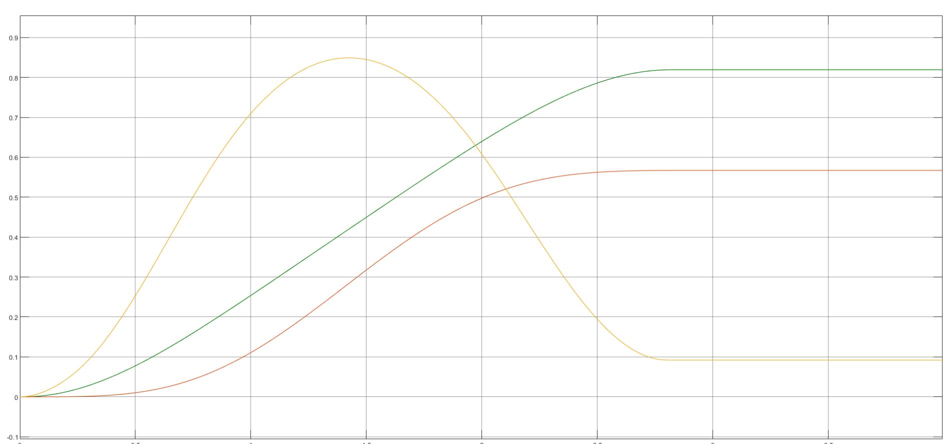

Figure 7: Value of x y and theta

Note that although the system ensures a tracking error of $10^{-4}/10^{-5}$ for the controlled variables while the error for not controlled variable θ is around 10^{-3} .

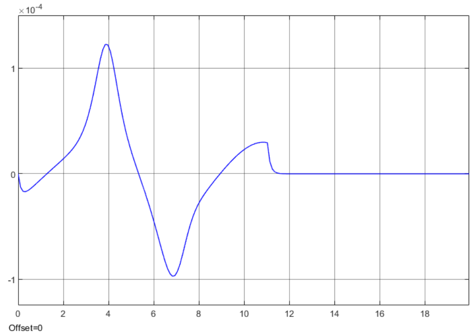

Figure 8: x error

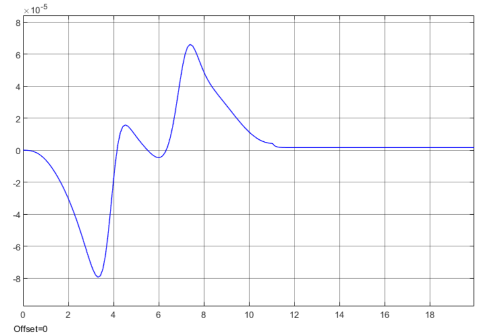

Figure 9: y error

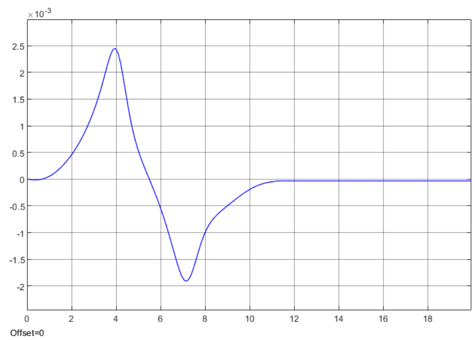

Figure 10: θ error

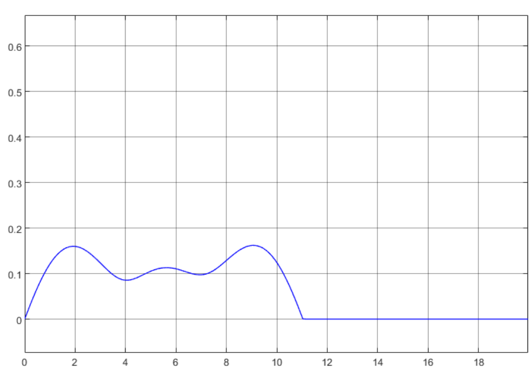

Figure 11: Heading velocity

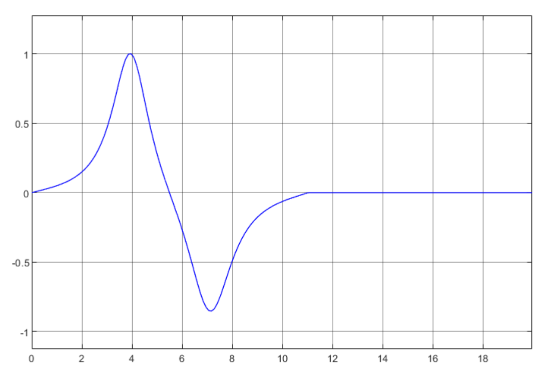

Figure 12: steering velocity

5 Question 5

Implement via software the unicycle posture regulator based on polar coordinates, with the state feedback computed through the Runge-Kutta odometric localization method. Starting and final configurations are $q_i = [x_i, y_i, \theta_i]^T = [\alpha + 1, 1, \pi/4]^T$, with α the last digit of your matriculation number, and $q_f = [x_f, y_f, \theta_f]^T = [0, 0, 0]^T$

Feedback controller in polar coordinate:

$$v = k_1 \rho \cos\gamma,$$

$$\omega = k_2\gamma + k_1 \sin\gamma\cos\gamma\left(1 + k_3\frac{\delta}{\gamma}\right)$$

with $k_1 = 1$, $k_2 = 1$, $k_3 = 0.2$.

$$Ts = 0.01$$

Where ρ, γ e δ are computed as follow:

$$\rho = \frac{p}{x^2 + y^2}$$
$$\gamma = \text{atan2}(y, x) - \theta + \pi$$
$$\delta = \text{atan2}(y, x) + \pi$$

A localization procedure is provided below to estimate the robot's state. To implement a feedback controller, you must always know the robot model configuration. A 2nd order Runge-Kutta approximation algorithm was used to obtain this configuration.

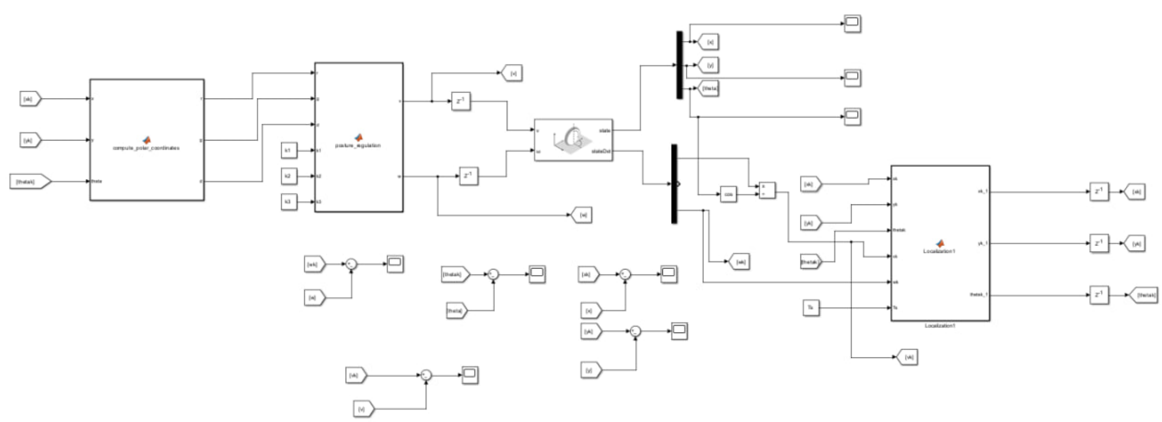

Figure 13: Simulink model

The values of v and w provided by the Simulink block of the unicycle do not coincide with the real values, since in reality they are measured indirectly through encoders that detect the displacement of each wheel:

$$v_k = \frac{2T_s\rho(\Delta\Phi_r + \Delta\Phi_s)}{\rho}; \quad \omega_k = \frac{dT_s\rho(\Delta\Phi_r - \Delta\Phi_s)}{\rho}$$

where d is the distance between the wheels of the differential drive robot and ρ is the radius of the wheels.

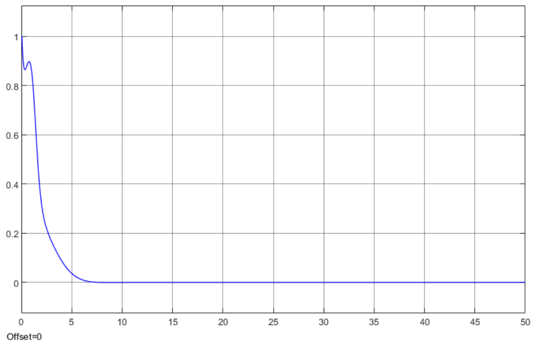

Figure 14: plot of x

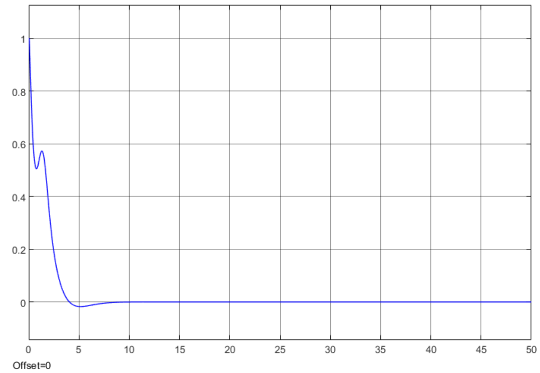

Figure 15: plot of y

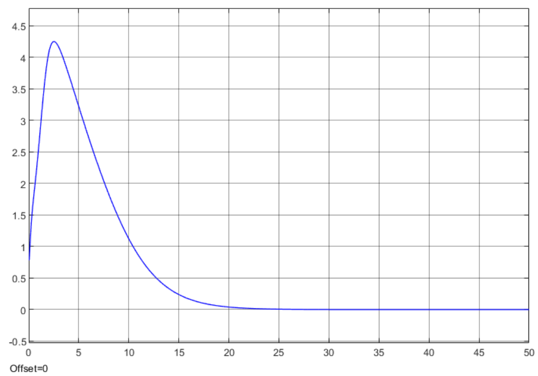

Figure 16: plot of θ

It can be seen from the figure that the odometry localization error is not brought exactly to zero at steady state, because of the slippage, inaccuracy in the kinematic paraters.

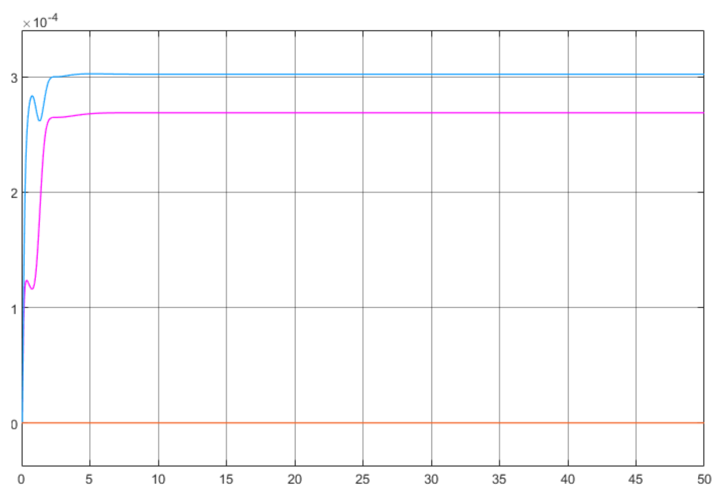

Figure 17: error