1. Derive the kinematic model of the mobile robot obtained by connecting N trailers to a rear-drive tricycle. Each trailer is a rigid body with an axle carrying two fixed wheels that can be assimilated to a single wheel located at the midpoint of the axle, and it is hinged to the midpoint of the preceding axis through a revolute joint. Denote by l the distance between the front wheel and the rear wheel axle of the tricycle, and by l_i the joint-by-joint length of the i-th trailer.

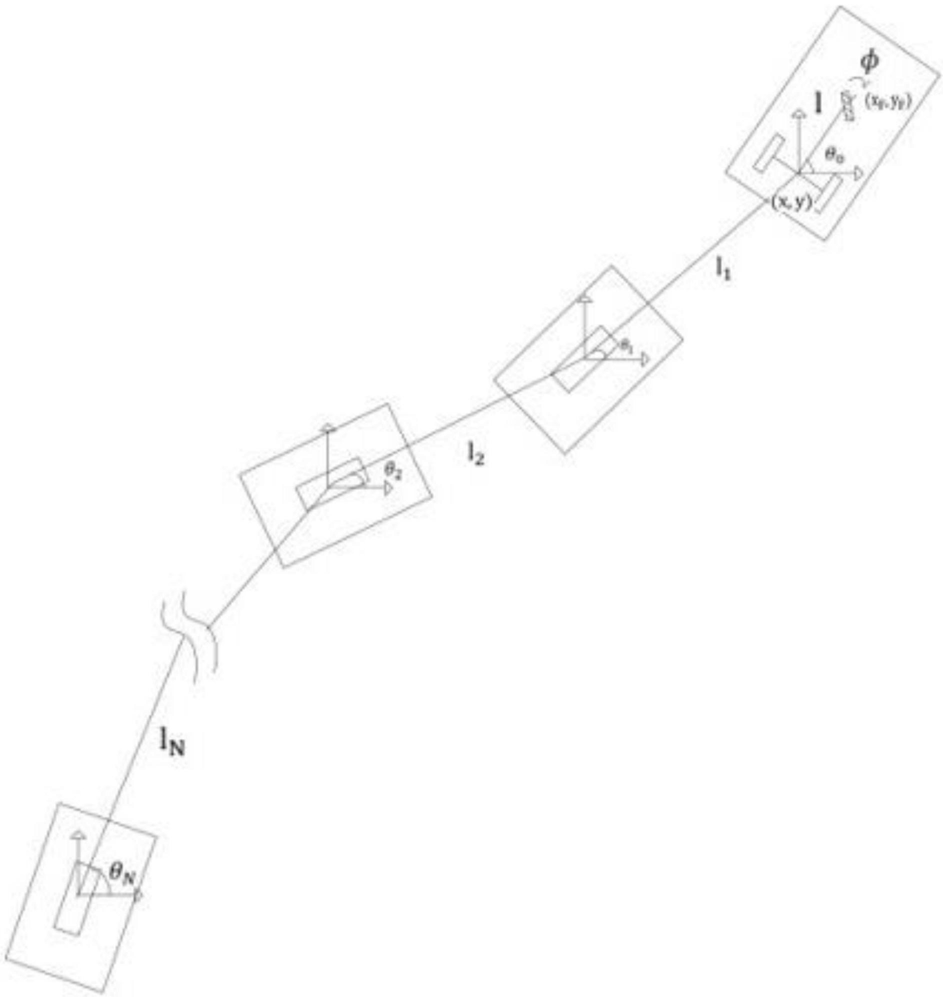


Figure 1:tricycle with trailers

The set of generalized coordinates are:

$$q = [x \quad y \quad \phi \quad \theta_0 \quad \theta_1 \ldots \theta_N]^T$$

(x, y) represent the coordinates of the midpoint between the two rear wheels of the tricycle; (x_F , y_F) are the coordinates of the steerable wheel ; (x_i , y_i) are the coordinates of the i-th trailer wheel axle midpoint, θ_0 indicates the orientation of the 2 fixed wheels of the tricycle compared to the x-axis , while ϕ indicates the orientation of the steerable wheel compared to the tricycle; in the end, $\theta_1 \ldots \theta_N$ are the orientation of each trailer with respect to the x-axis.

The system has the N + 2 following constraints.

$$\dot{x}_f sin(\theta_0 + \phi) - \dot{y}_f cos(\theta_0 + \phi) = 0$$

$$\dot{x} sin\theta_0 - \dot{y} cos\theta_0 = 0$$

$$\dot{x}_i sin\theta_i - \dot{y}_i cos\theta_i = 0 \qquad i = 1, \ldots., N$$

The coordinates of the steerable point can be rewritten respect the position of (x,y)

$$x_f = x + l cos\theta_0$$

$$y_f = y + l sin\theta_0$$

Considering that l_i is the hinge-to-hinge length of the i-th trailer, it's easy to note that:

$$x_i = x - \sum_{j==1}^{i} l_j \; cos\theta_j$$

$$y_i = y - \sum_{j==1}^{i} l_j \; sin\theta_j$$

So the kinematic constraints become

$$(x + l cos\theta_0) sin(\theta_0 + \phi) - (y + l sin\theta_0) cos(\theta_0 + \phi) =$$

$$= \dot{x} sin(\theta_0 + \phi) - \dot{y} cos(\theta_0 + \phi) - \dot{\theta}_0 l cos\phi = 0$$

$$\dot{x} sin\theta_0 - \dot{y} cos\theta_0 = 0$$

$$\left(x - \sum_{j==1}^{i} l_j \, cos\theta_j \right) sin\theta_i - \left(y - \sum_{j==1}^{i} l_j \, sin\theta_j \right) cos\theta_i =$$

$$= \dot{x} sin\, \theta_i - \dot{y} cos\, \theta_i + \sum_{j==1}^{i} \dot{\theta}_j l_j \; cos(\theta_i - \theta_j) = 0 \quad i = 1, \ldots., N$$

These constraints can be written in the form:

$$A^T(q)\dot{q} = 0$$

$A^T(q)$ is a matrix of dimension (N+2)x(N+4), because it has N+2 constraints and N+4 coordinates

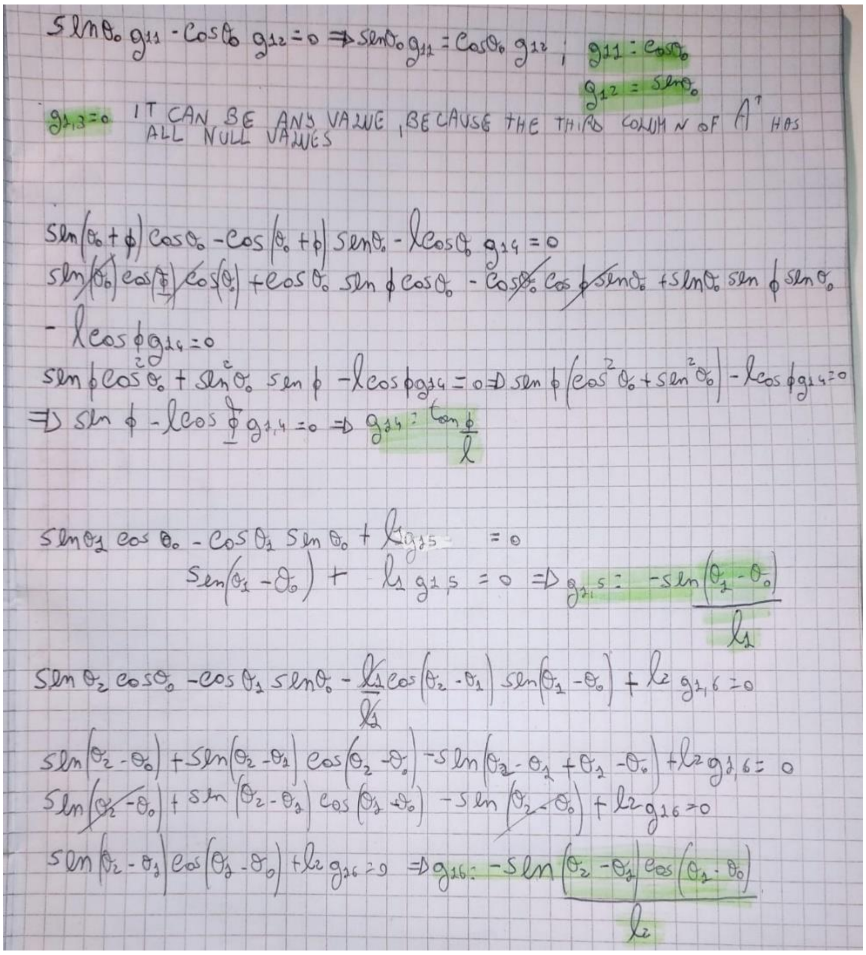
$$\begin{bmatrix} sin\theta_o & -cos\theta_o & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ sin(\theta_o + \phi) & -cos(\theta_o + \phi) & 0 & -l cos\phi & 0 & \cdots & \cdots & \cdots & 0 \\ sin\theta_1 & -cos\theta_1 & 0 & 0 & l_1 & 0 & \cdots & \cdots & 0 \\ sin\theta_2 & -cos\theta_2 & 0 & 0 & l_1 cos(\theta_2 - \theta_1) & l_2 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ sin\theta_N & -cos\theta_N & 0 & 0 & l_1 cos(\theta_N - \theta_1) & l_2 cos(\theta_N - \theta_2) & \cdots & \cdots & l_N \end{bmatrix}$$

The null space of the constraint matrix is spanned by the columns $g_1(q) g_2(q)$. To obtain the values of $g_1(q)$ *and* $g_2(q)$ we should resolve the equation:

$$A^T(q) g = 0$$

The matrix G(q) $= [g_1(q) g_2(q)]$has m-n=2 columns and N+4 rows

Below I will show the first calculations of this procedure



$sen\theta_0 \, g_{11} - cos\theta_0 \, g_{12} = 0 \Rightarrow sen\theta_0 \, g_{11} = cos\theta_0 \, g_{12}$; $g_{11} = cos\theta_0$

$g_{12} = sen\theta_0$

$g_{1,3} = 0$ IT CAN BE ANY VALUE, BECAUSE THE THIRD COLUMN OF A^T HAS ALL NULL VALUES

$sen(\theta_0+\phi)cos\theta_0 - cos(\theta_0+\phi)sen\theta_0 - l cos\phi \, g_{14} = 0$

$\cancel{sen(\theta_0)cos(\phi)cos(\theta_0)} + cos\theta_0 \, sen\phi \, cos\theta_0 - \cancel{cos\theta_0 \, cos\phi \, sen\theta_0} + sen\theta_0 \, sen\phi \, sen\theta_0$

$- l cos\phi \, g_{14} = 0$

$sen\phi \, cos^2\theta_0 + sen^2\theta_0 \, sen\phi - l cos\phi \, g_{14} = 0 \Rightarrow sen\phi(cos^2\theta_0 + sen^2\theta_0) - l cos\phi \, g_{14} = 0$

$\Rightarrow sen\phi - l cos\phi \, g_{1,4} = 0 \Rightarrow g_{14} = \frac{tan\phi}{l}$

$sen\theta_1 \, cos\theta_0 - cos\theta_1 \, sen\theta_0 + l_1 g_{15} = 0$

$sen(\theta_1-\theta_0) + l_1 g_{15} = 0 \Rightarrow g_{15} = \frac{-sen(\theta_1-\theta_0)}{l_1}$

$sen\theta_2 \, cos\theta_0 - cos\theta_2 \, sen\theta_0 - \frac{l_1}{l_1} cos(\theta_2-\theta_1) sen(\theta_1-\theta_0) + l_2 g_{1,6} = 0$

$sen(\theta_2-\theta_0) + sen(\theta_2-\theta_1)cos(\theta_1-\theta_0) - sen(\theta_2-\theta_1+\theta_1-\theta_0) + l_2 g_{1,6} = 0$

$\cancel{sen(\theta_2-\theta_0)} + sen(\theta_2-\theta_1)cos(\theta_1-\theta_0) - \cancel{sen(\theta_2-\theta_0)} + l_2 g_{16} = 0$

$sen(\theta_2-\theta_1)cos(\theta_1-\theta_0) + l_2 g_{16} = 0 \Rightarrow g_{16} = \frac{-sen(\theta_2-\theta_1)cos(\theta_1-\theta_0)}{l_2}$

By iterating the procedure up, we get $g_{1,N+4} = \frac{1}{l_N}(\prod_{j=1}^{N-1} cos(\theta_j - \theta_{j-1}))sin(\theta_N - \theta_{N-1})$.

To satisfy the equation $A^T(q)g_2 = 0$, g_2 has been chosen as a vector of all null values, except for the third element that I chose equal to 1(the third column of $A^T(q)$ has all null elements).

The final result is:

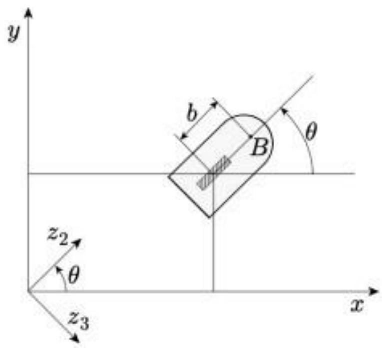
$$G(q) = \begin{bmatrix} cos\theta_0 & 0 \\ sin\,\theta_0 & 0 \\ 0 & 1 \\ \frac{1}{l}tan\phi & 0 \\ -\frac{1}{l_1}sin(\theta_1-\theta_0) & 0 \\ -\frac{1}{l_2}cos(\theta_1-\theta_0)sin(\theta_2-\theta_1) & 0 \\ \vdots & \vdots \\ \frac{1}{l_N}(\prod_{j=1}^{N-1} cos(\theta_j-\theta_{j-1}))sin(\theta_N-\theta_{N-1}) & 0 \end{bmatrix} = [g_1(q)g_2(q)]$$

It's possible for us to write the kinematic control system:

$$\dot{q} = g_1(q)v + g_2(q)\omega$$

where v is the driving velocity of the rear wheels and ω the steering velocity of the tricycle.

2. 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 $q_i = [x_i \quad y_i \quad \theta_i]^T = [0 \quad 0 \quad 0]^T$, to the configuration $q_f = [x_f \quad y_f \quad \theta_f]^T = [\alpha \quad 1 \quad \pi/2]^T$, with α the last digit of your matriculation number. Then, determine a timing law over the path to satisfy the following velocity bounds $|v(t)| \leq 1$ m/s and $|\omega(t)| \leq 1$ rad/s.



The aim of this exercise is to plan a path of a unicycle from the point q_i=[0 0 0] to the point q_f=[4 1 $\frac{\pi}{2}$] (because my matriculation number is P38000094), through the cubic Cartesian polynomial technique.

The unicycle is a differentially flat system with x and y as flat outputs. Its geometric model is

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tilde{\omega} + \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \tilde{v}$$

With

$$\theta(s) = \operatorname{atan2}\big(y'(s), x'(s)\big) + k\pi \,, k = \{0,1\}$$
$$\tilde{v}(s) = \pm\sqrt{x'(s)^2 + y'(s)^2}$$
$$\tilde{\omega}(s) = \frac{y''(s)x'(s) - x''(s)y'(s)}{x'(s)^2 + y'(s)^2}$$

I chose k=0, so as to force the robot to move forward

The cubic Cartesian polynomial is defined by equations:

$$x(s) = s^3 x_f - (s-1)^3 x_i + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2$$
$$y(s) = s^3 y_f - (s-1)^3 y_i + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2$$

where

$$x(s_i) = x(0) = x_i$$
$$x(s_f) = x(1) = x_f$$
$$y(s_i) = y(0) = y_i$$
$$y(s_f) = y(1) = y_f$$

The initial and the final orientation, are useful to impose some boundary conditions to retrieve the coefficients α_x, α_y, β_x, β_y. We impose $k_i = k_f = k = 2$(k>0 because the robot is moving forward)

$$
\begin{aligned}
x'(s_i) &= x'(0) = k_i \cos\theta_i \\
y'(s_i) &= y'(0) = k_i \sin\theta_i \\
x'(s_f) &= x'(1) = k_f \cos\theta_f \\
y'(s_f) &= y'(1) = k_f \sin\theta_f
\end{aligned}
\qquad
\begin{aligned}
\alpha_x &= k\cos\theta_f - 3x_f \\
\alpha_y &= k\sin\theta_f - 3y_f \\
\beta_x &= k\cos\theta_i + 3x_i \\
\beta_y &= k\sin\theta_i + 3y_i
\end{aligned}
$$

The exercise imposes that |v(t)|≤1 and $|\omega(t)|$ ≤1, to satisfy this condition, it is sufficient to impose a t_f sufficient big, I chose t_f equal to 10s.

The code is present in the file HW2_2.m, the results are:

The plots of the results are shown below:

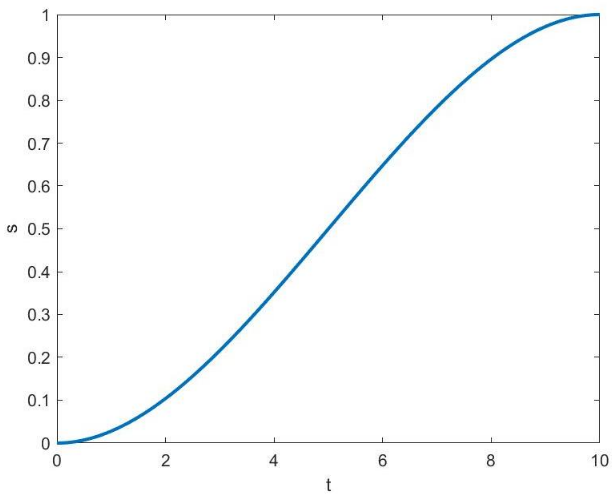


Figure 2:s with respect to time

The value of $\omega(t)$ and v(t) have been obtained: $\omega(t)$=$\tilde{\omega}$(s)$\dot{s}(t)$ and $v(t)$=$\tilde{v}$(s)$\dot{s}(t)$

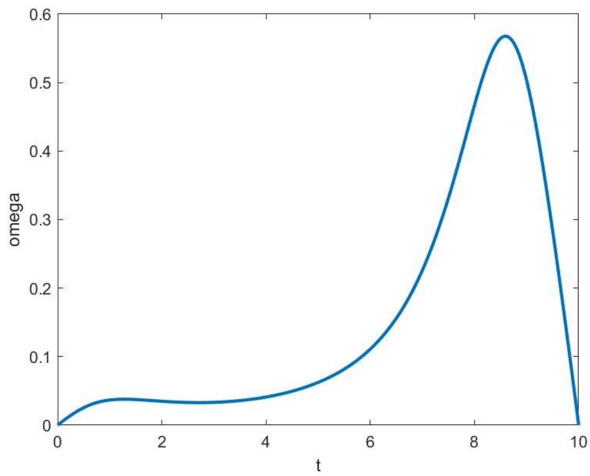


Figure 3:angular velocity with respect to time

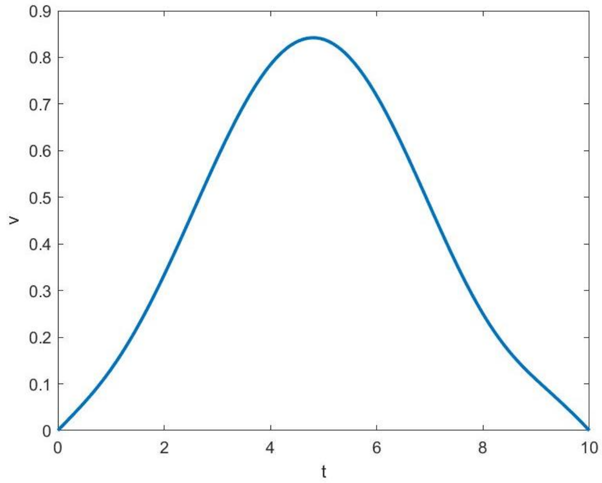


Figure 4:the heading velocity with respect to time

It is easy to see that v and ω do not exceed the imposed constraints, therefore the scaling is not necessary

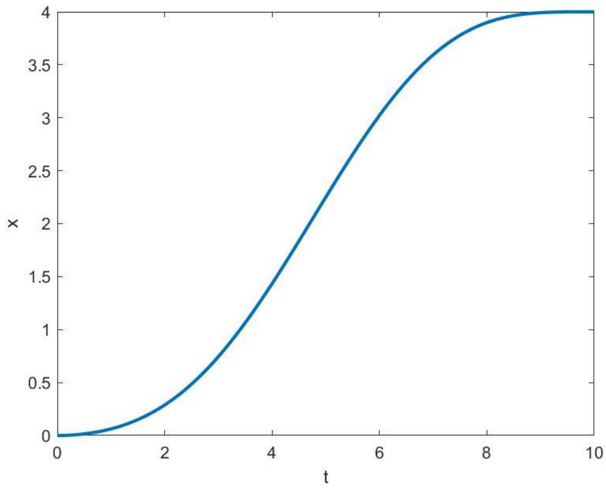


Figure 5:x with respect to time

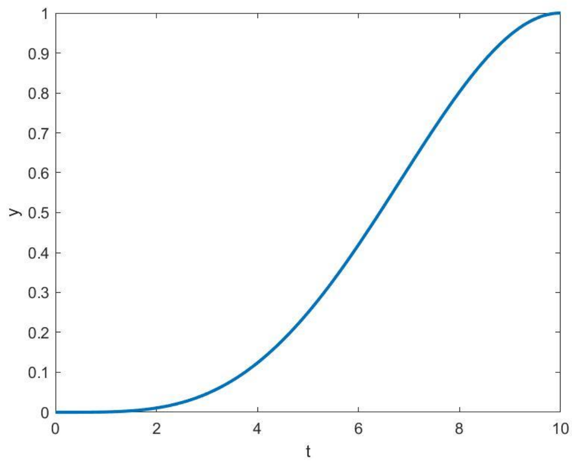


Figure 6:y with respect to time

The x and y reach the origin in the predetermined time.

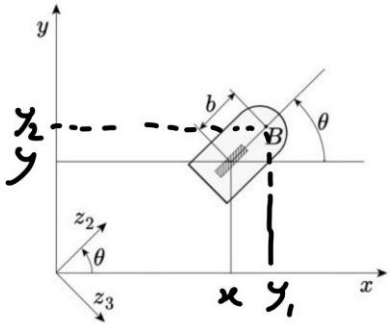
3. Given the trajectory in the previous point, implement via software an input/output linearization control approach to control the unicycle's position. Adjust the trajectory accordingly to fit the desired coordinates of the reference point B along the sagittal axis, whose distance to the wheel's center it is up to you.

Feedback linearization is a strategy employed to control nonlinear systems. Feedback linearization techniques may be applied to nonlinear control systems of the form

$$\dot{x} = f(x) + g(x)u$$
$$y = h(x)$$

Where x(t)∈ $\mathbb{R}^n$ is the state vector, y∈ $\mathbb{R}$ is the output vector and u∈ $\mathbb{R}$ is the output vector.

The approach involves transforming a nonlinear control system into an equivalent linear control system through a change of variables and a suitable control input. In our case:



B is a point with distance b=0.04 from the center of the wheel along the sagittal axis.

(x,y) are the coordinates of the center of the wheel.

(y_1, y_2) are the coordinates of the point **B.**

B has coordinates:

$$y_1 = x + b\cos\theta$$
$$y_2 = y + b\sin\theta$$

The time derivatives of these outputs are (substituting the kinematic model of the unicycle). Notice that det(T(θ))$\neq 0 \leftrightarrow b \neq 0$.

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -b\sin\theta \\ \sin\theta & b\cos\theta \end{bmatrix}}_{T(\theta)} \begin{bmatrix} v \\ \omega \end{bmatrix} = T(\theta)\begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

It is possible to design

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = T(\theta)^{-1}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (2)$$

u_1, u_2 are the 2 virtual control input.

Substituting (2) into (1), we get

$$\begin{cases} \dot{y}_1 = u_1 \\ \dot{y}_2 = u_2 \end{cases}$$

The following simple controller can be designed(with k_1, k_2>0). This controller guarantees exponential convergence to the desired $y_{1,d}$ and $y_{2,d}$. For this implementation I chose $k_1 = k_2$ =2

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

Unfortunately, this approach controls the position of the point B only, leaving the orientation uncontrolled.

The following implementation is based on the trajectory calculated in the previous exercise, with the appropriate modifications to fit the desired coordinates of the reference point B along the sagittal axis.

The scheme Simulink for this implementation is saved as HW2_3.slx

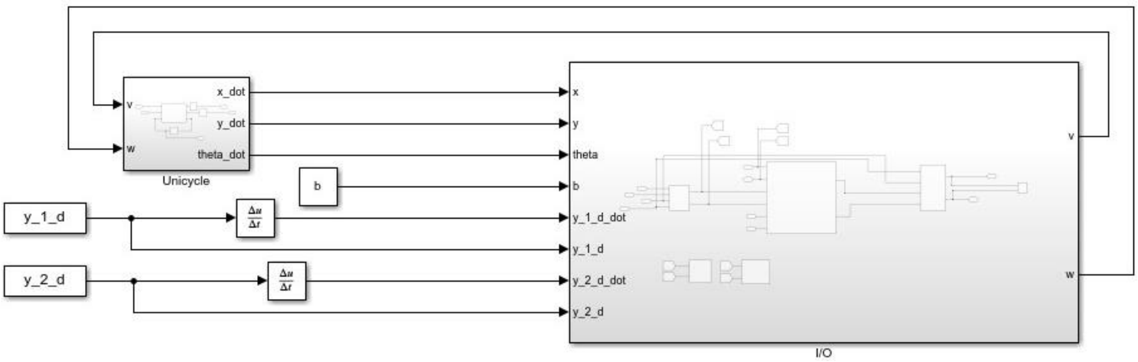


Figure 7:Scheme of the third exercise

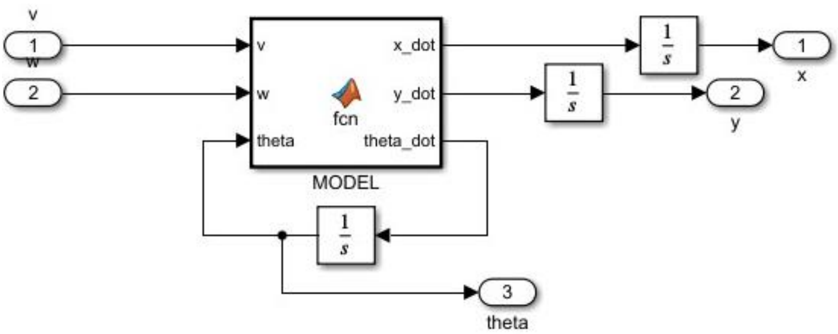


Figure 8:unyicicle

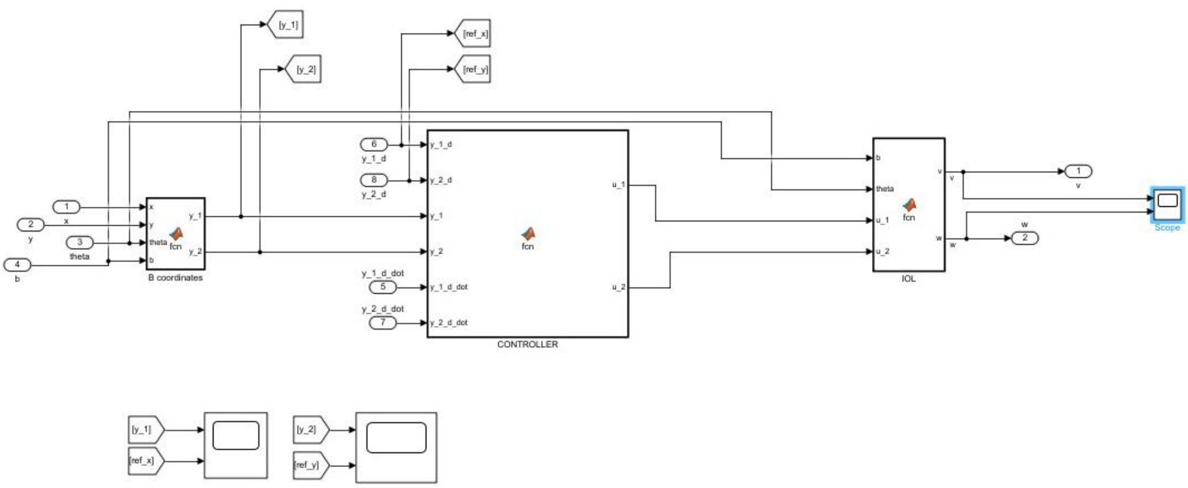


Figure 9:Subsystem third exercise

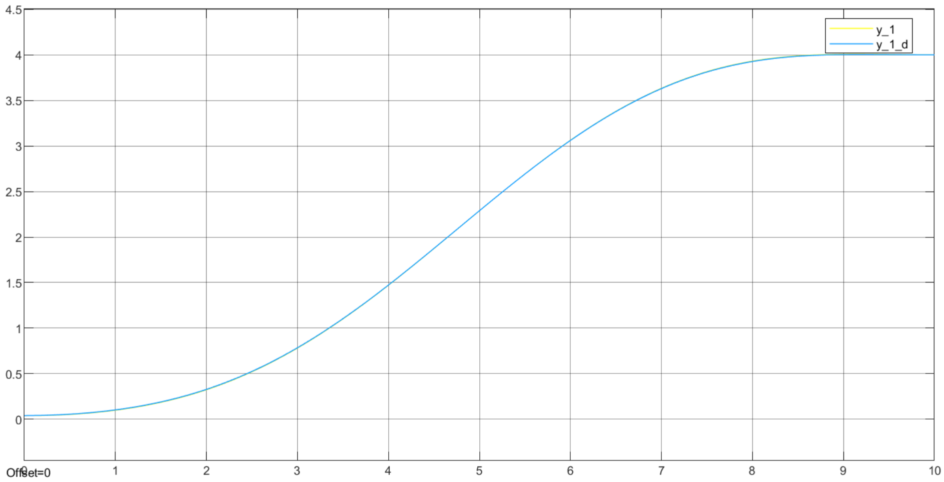


Figure 10:y_1_desired and the real y_1

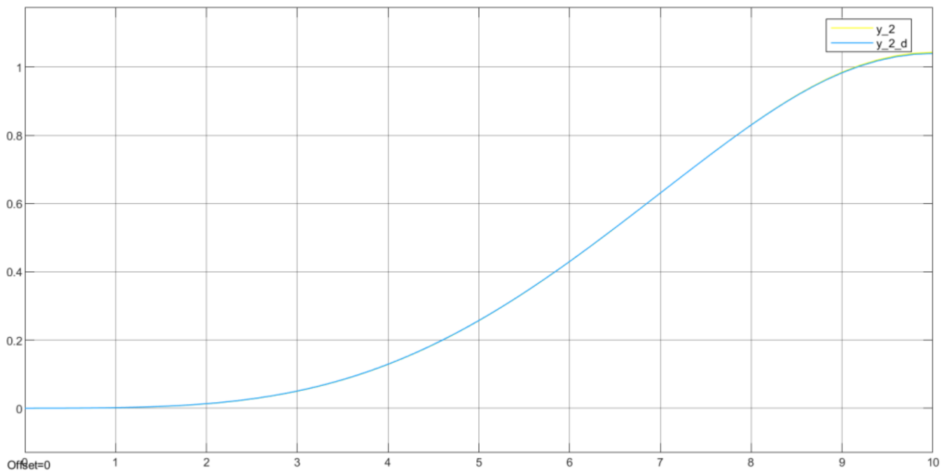


Figure 11:y_2_desired and the real y_2

As it is possible to note, the real and desired trajectories are very similar. The point B does not converge perfectly in the point (4,1), because the reference point remains the center of the unicycle, and therefore B maintains a distance b from the center.

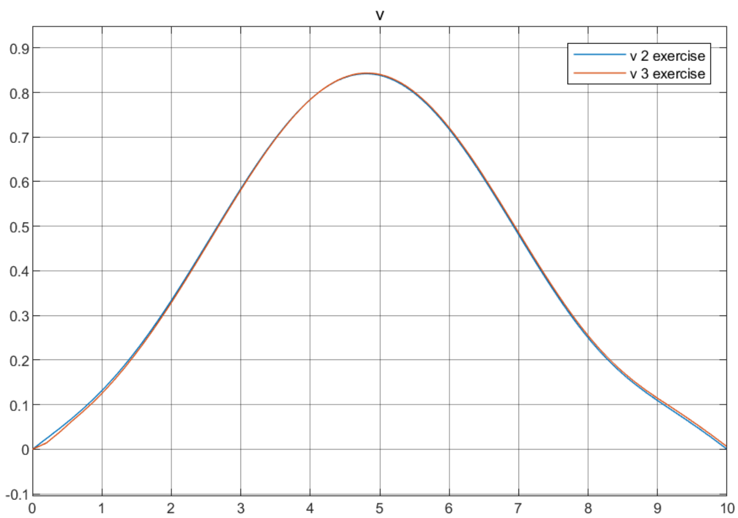


Figure 12:values of v in second and third exercise

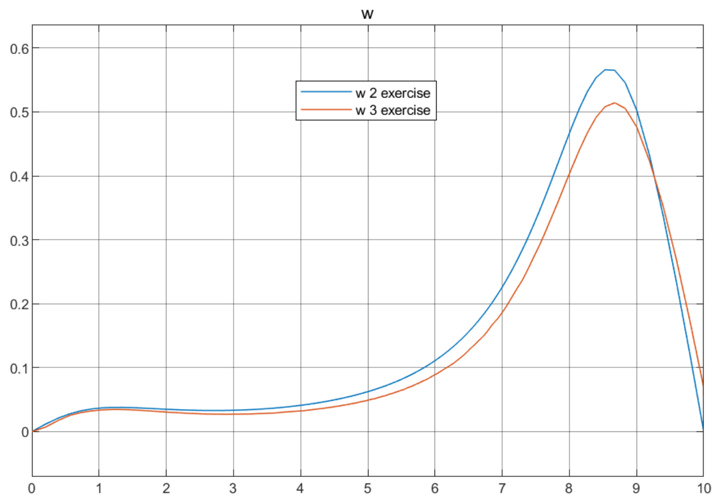


Figure 13:value of w in second and third exercise

The trends of v between the second and third exercise are very similar, while as regards ω it can be seen that in the second exercise the function takes on a greater value. So we can say that v and ω stay under the limit of 1 in module.

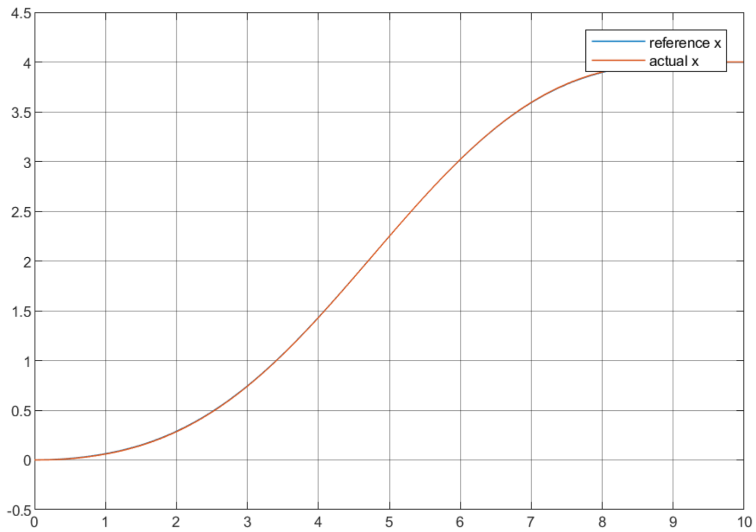


Figure 14:the actual signal of x and the reference signal x form the previous exercise

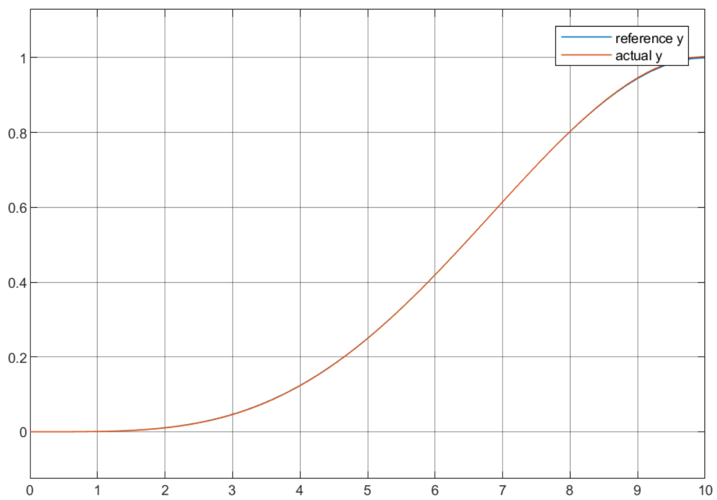


Figure 15:the actual signal of x and the reference signal x form the previous exercise

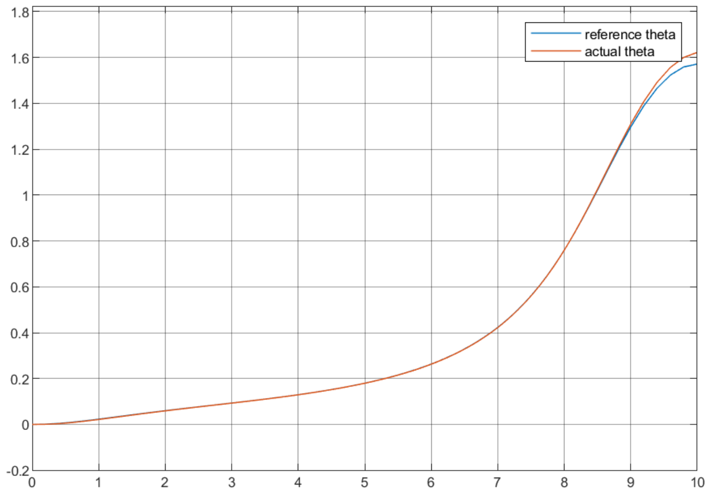


Figure 16:the actual signal of theta and the reference signal theta form the previous exercise

We can see that the signals x y θ are well tracked, and even if θ is uncontrollable, it follows the reference well due to the path and the chosen earnings.

4. Implement via software the Cartesian regulator to bring the unicycle from the point $q_i = [x_i \quad y_i \quad \theta_i]^T = [\alpha + 1 \quad 2 \quad {}^{\pi}/_{2}]^T$, with α the last digit of your matriculation number, to the origin.

The regulation problem takes care of bringing the robot to a given configuration. The aim of this exercise is to bring the unicycle from the point q_i =[$x_i \quad y_i \quad \theta_i$]=[5 2 $\frac{\pi}{2}$] (because my matriculation number is P38000094) to the origin , through the Cartesian Regulation.

The position error is

$$e_p = [-x \quad -y]^T$$

Recall the kinematic model of the unicycle, the following regulation controller is designed

$$\begin{cases} v = -k_1(x\cos\theta + y\sin\theta) \\ \omega = k_2(atan2(y,x) + \pi - \theta) \end{cases}$$

With $k_1, k_2 > 0$

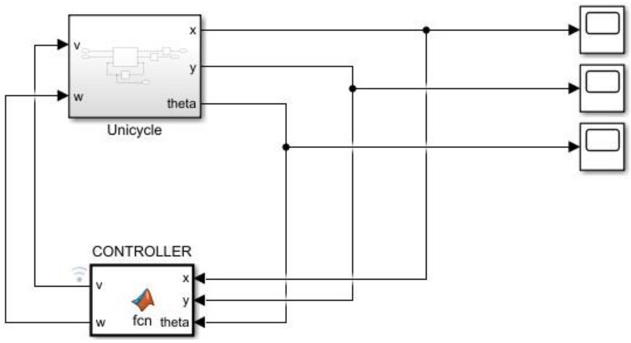


Figure 17:scheme of the fourth exercise

I chose $\mathrm{k}_1 = 1$ and $\mathrm{k}_2 = 6$

The implementation is present in the file HW2_4.slx, the results are:

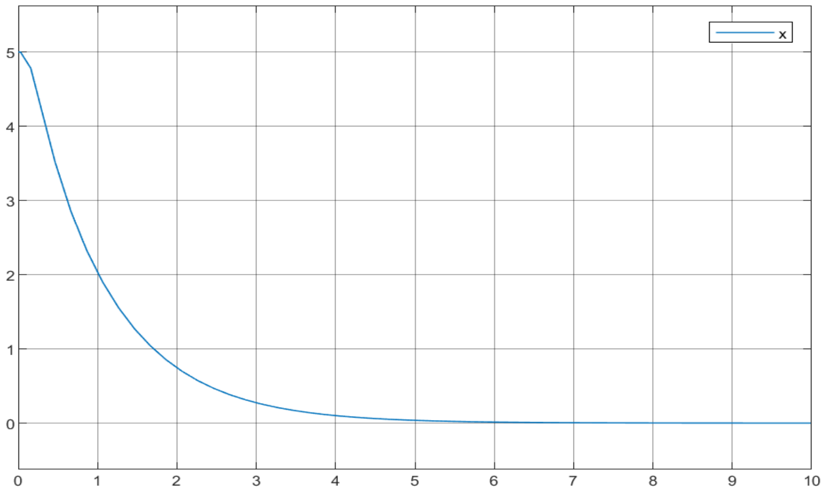


Figure 18:path of the x

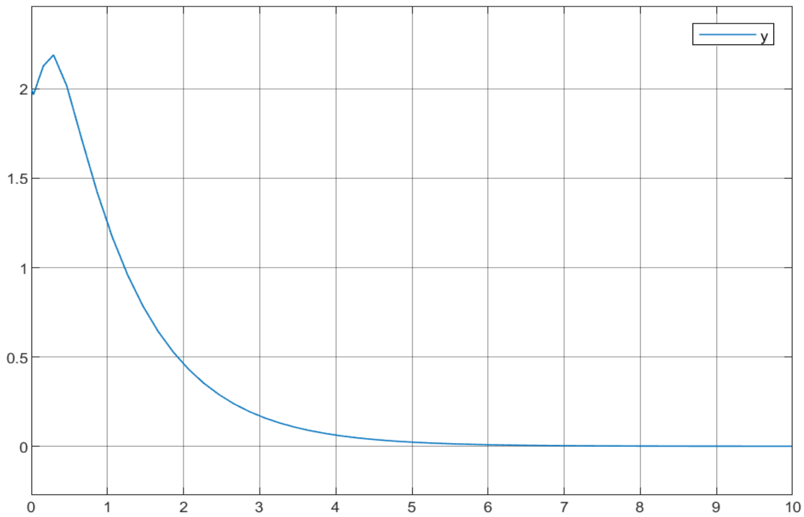


Figure 19:path of the y

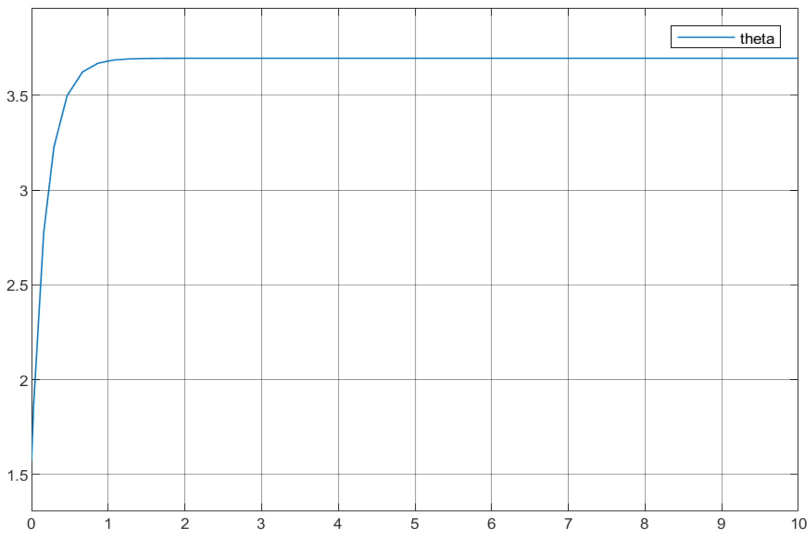


Figure 20:path of theta

How it is easy to see the x and the y signals can reach the origin, while θ assumes a constant value of approximately 3.7.

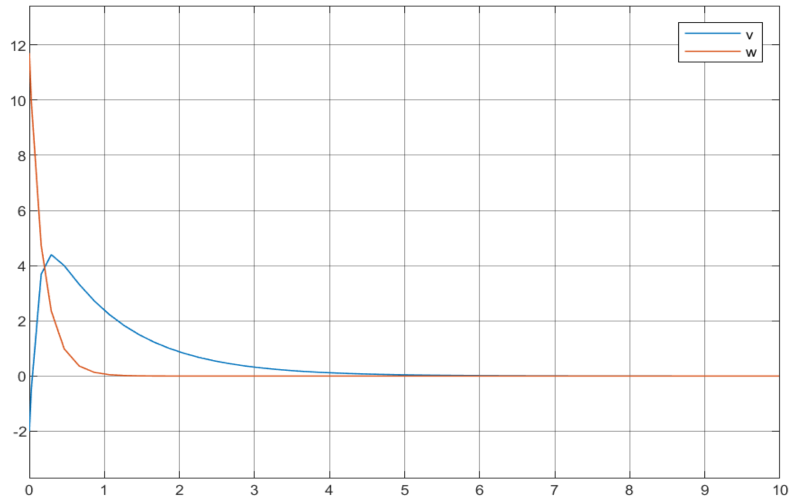


Figure 21: heading and angular velocity

the v starts from a negative value, this means that it initially moves backward, while the w takes on a value approximately equal to 12

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 as in the previous exercise.

The localization is the procedure of estimating the robot's state. The Runge-Kutta odometric localization method takes in input from the unicycle the constant heading velocity v_k and angular velocity ω_k in a time interval $[t_k, t_k+T_s]$, where the sampling time Ts has been chosen equal to 0.01 and gives back the value

$q(t_{k+1})= q_{k+1}$ through these equations

$$\begin{cases} x_{k+1} = x_k + v_k T_s \cos\left(\theta_k + \frac{1}{2}\omega_k T_s\right) \\ y_{k+1} = y_k + v_k T_s \sin\left(\theta_k + \frac{1}{2}\omega_k T_s\right) \\ \theta_{k+1} = \theta_k + \omega_k T_s \end{cases}$$

To express the problem in polar coordinates:

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

Where:

- $\rho=|e_p|$ is the distance between the unicycle and the origin
- γ is the angle between e_p and the sagittal axis
- $\delta$is the angle between e_p and the x –axis

We design the following controllers

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

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

- $k_1, k_2, k_3 > 0$

The value chosen are $k_1 = 2, k_2 = 7, k_3 = 1$

The implementation is present in the file HW2_5.slx, the results are:

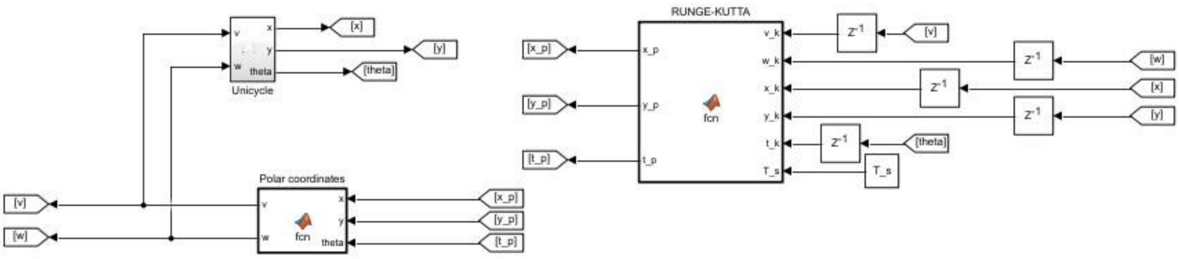


Figure 22:scheme of the fifth exercise

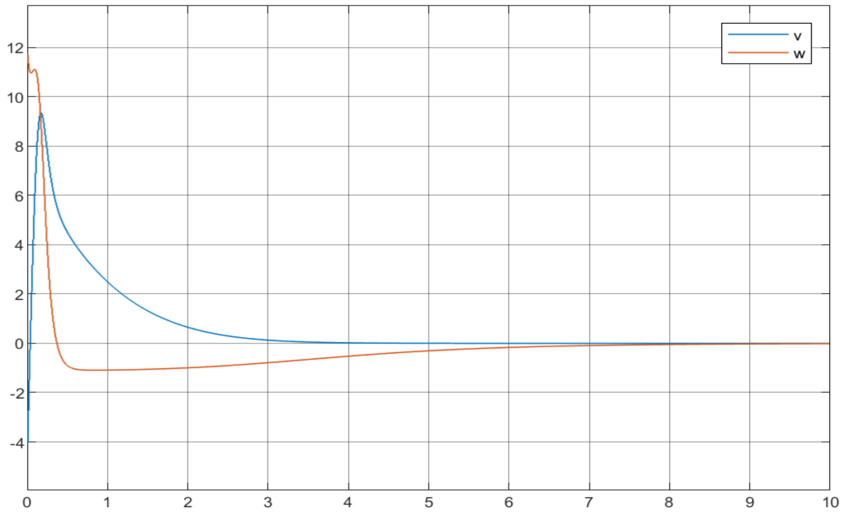


Figure 23:heading and angular velocity

the v starts from a negative value, this means that it moves backwards, while the w takes on a value approximately equal to 12.

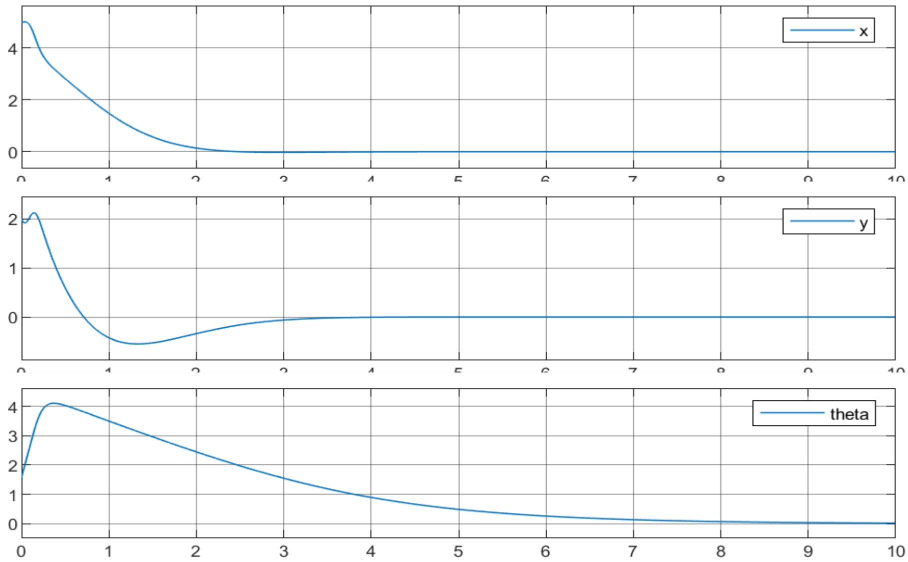


Figure 24:x y theta of the unicycle

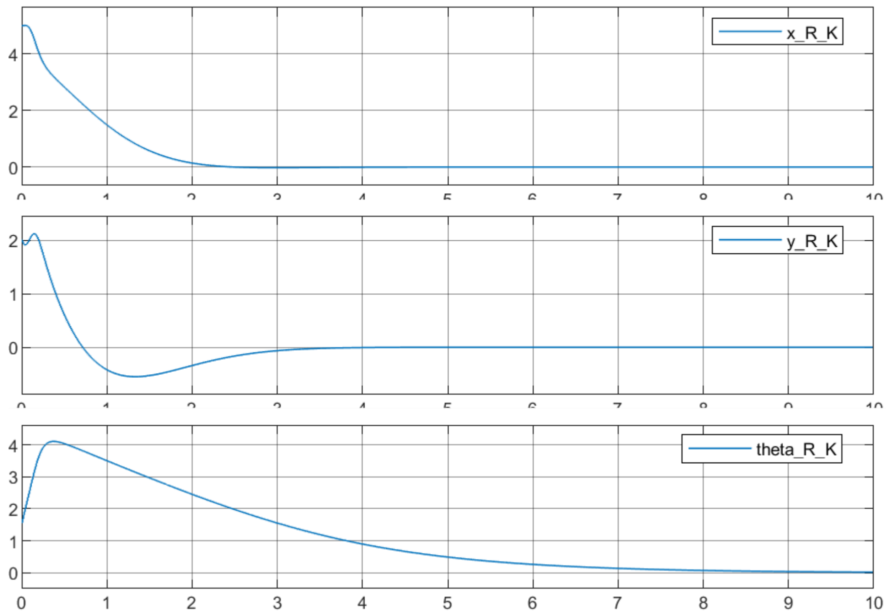


Figure 25:x y theta in output from the RUNGE-KUTTA

Signals x, y and θ tend to 0 asymptotically.