Robótica Móvel

Exercises on Inverse Kinematics
in Wheeled Robots

Main purpose

- Simulate wheeled robot locomotion using inverse kinematics
- These exercises complement the approach of the previous class
- Velocities of robot parts are calculated for some specified trajectories
- Then check those trajectories by applying the direct kinematics
- Robot to be used: differential drive

1-Function for inverse kinematics of differential drive

```
function [VR,VL]=invkinDD(X,Y,th,L,t)
% X - target position in x (meters)
% Y - target position in y (meters)
% th- target orientation in th (radians)
% L - wheel separation (meters)
% t - time to accomplish the trajectory (seconds)
% Notice: X, Y and th can not be set all at the same time
% one of them must be NaN for the function to return valid results.
```

- Cases are to be managed
    - X and th
    - Y and th
- X and Y are managed in a different exercise.

Hints to solve Exercise 1

- Cases are based on constant velocities V_R and V_L
- The equations involved are:

$$\theta(t) = \int_0^t \omega \mathrm{d}\tau = \omega \int_0^t \mathrm{d}\tau = \omega t$$
$$x(t) = V \int_0^t \cos(\omega\tau)\mathrm{d}\tau = \frac{V}{\omega}\sin(\omega t)$$
$$y(t) = V \int_0^t \sin(\omega\tau)\mathrm{d}\tau = -\frac{V}{\omega}\left[\cos(\omega\tau)\right]_0^t = \frac{V}{\omega}(1-\cos(\omega t))$$

$$\omega = \frac{V_R - V_L}{L} \qquad V = \frac{V_L + V_R}{2}$$

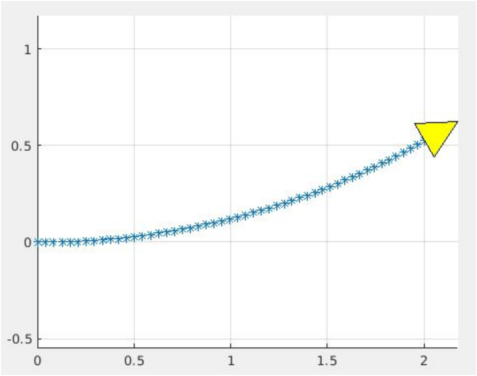
- Rearrange expressions to have V_R and V_L in function of V and ω

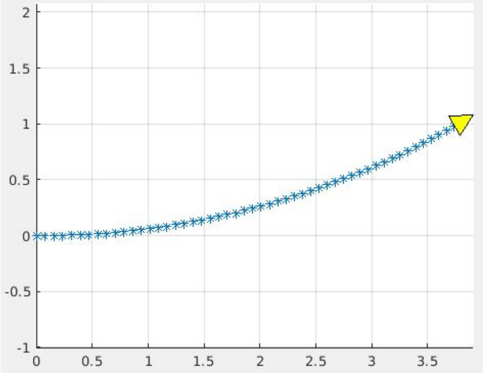
2-Test function invkinDD()

- Test function invkinDD() with examples from TP class
  - L = 50 cm
  - starting at (0,0,0)
  - arrives in t = 5 s,
  - **2 m** in **x**
  - with a final orientation (θ) of +30◦ (π/6)
- Verify with Direct Kinematics from previous class

Notice that now the number of simulation steps is defined by the time (5 s in this case) and the resolution of the simulation step Dt (Δt). In the image Dt=0.1 (but finer resolution gives greater accuracy in the point reached)

- Repeat for the second example given:
  - Destination is **1 m** in **y** instead of 2 m in x
  - All the rest remains





Limitations of invkinDD() and alternatives

- invkinDD() does not allow to specify (x,y) destination pairs easily
- A simple alternative is to perform linear trajectories and pure rotations
- One solution is to create a function that can calculate the velocities for those two situations
- In general, the robot should orient towards next sub-goal (steady rotation) and then move to that goal
- Repeat procedure for all points in the path

3-Create invkinDDxy()

```
Function [VR,VL]=invkinDDxy(X,Y,L,t,Xn,Yn)
% X - target position in x (meters)
% Y - target position in y (meters)
% L - wheel separation (meters)
% t - time to accomplish the trajectory (seconds)
% Xn - next target position in x (meters)
% Yn - next target position in y (meters)
```

If X==0 and Y==0 generate velocities to orient the robot to next target (Xn,Yn)

If X~=0 or Y~=0 generate velocities to move to (X,Y) and ignore (Xn,Yn)

4-Calculate resulting path with several via points

- Create the trajectories to cover the following points with the indicated times:
    - **P0** → 3s → **P1** → 4s → **P2** → 5s → **P3** → 4s → **P0**
        - P0=(0,0)
        - P1=(3,2)
        - P2=(5,1)
        - P3=(2,-2)
- At each point there must be first a rotation to orient robot to the next point
    - That rotation should take 1 second
- Calculate the velocities using relative increments ($\mathbf{P_{i+1}} - \mathbf{P_i}$) passed to the function
- Generate the trajectories and accumulate them (points and orientations)
- Animate a triangular robot to demonstrate that trajectories were correctly calculated