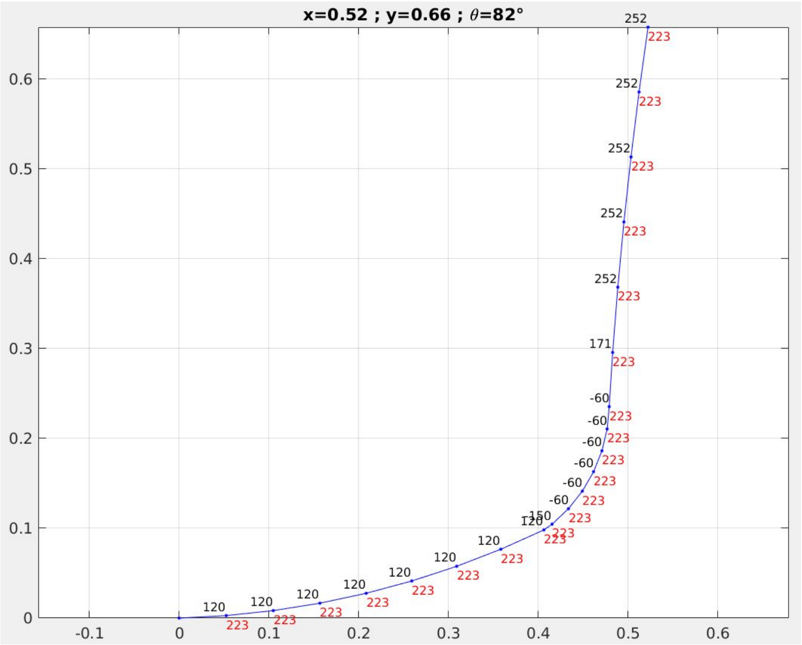
Robótica Móvel

Robot Localization - Part 1

1- Odometry calculation for differential drive

- Consider a differential drive robot
    - Wheel diameter=20 cm
    - Wheel separation=60 cm
    - Pulses per full wheel turn = 2048
- What is the approximate final position and orientation of the robot if the encoders counters (**R**ight and **L**eft) showed the following values during the sampling times?
    - Consider the robot to start at (0,0,0)



Left and right wheels encoder increments

```
R: 0,223,446,669,892,1115,1338,1561,1784,2007,2230,2453,2676,2899,3122,3345,3568,3791,4014,4237,4460,4683
L: 0,120,240,360,480, 600, 720, 840, 960, 810, 750, 690, 630, 570, 510, 450, 621, 873,1125,1377,1629,1881
```

Expressions to use for differential drive odometry

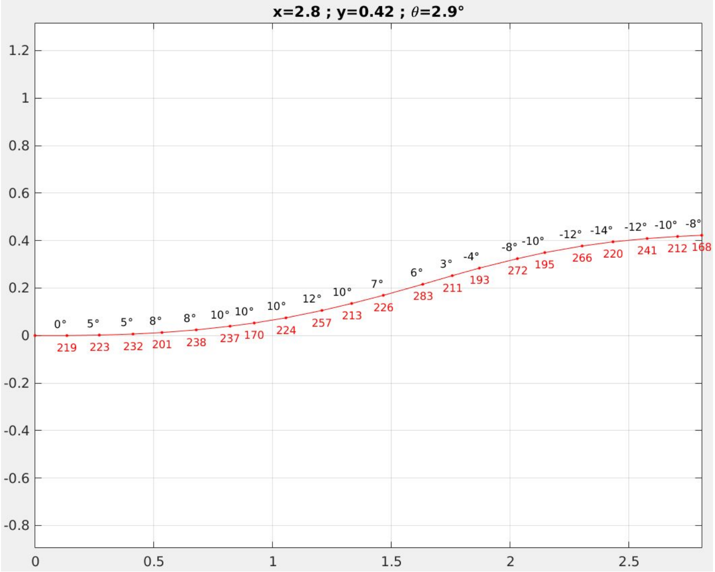
- $k = \frac{\pi d}{n C_e}$
  - d - wheel diameter
  - n - gear ratio (motor/wheel)
  - C_e - number of pulses per turn (encoder)

- $\Delta l_L = k N_L$
- $\Delta l_R = k N_R$

- $\Delta \theta_i = \frac{\Delta l_{R_i} - \Delta l_{L_i}}{D}$
- $\Delta l_i = \frac{\Delta l_{R_i} + \Delta l_{L_i}}{2}$
- $\theta_i = \theta_{i-1} + \Delta \theta_i$
- $x_i = x_{i-1} + \Delta l_i \cos \theta_i$
- $y_i = y_{i-1} + \Delta l_i \sin \theta_i$

2- Odometry calculation for a tricycle

- Consider a tricycle with steering and traction wheel in the front
  - Wheel diameter=20 cm
  - Distance to rear wheels =80 cm
  - Pulses per full wheel turn = 1024
  - Steering resolution = 1º
- What is the approximate final position and orientation of the robot if the encoder counter and the steering angle (º) showed the following values during the sampling times?
  - Consider the robot to start at (0,0,0)



Steering angles and wheel encoder increments

S: 0,219,442,674,875,1113,1350,1520,1744,2001,2214,2440,2723,2934,3127,3399,3594,3860,4080,4321,4533,4701

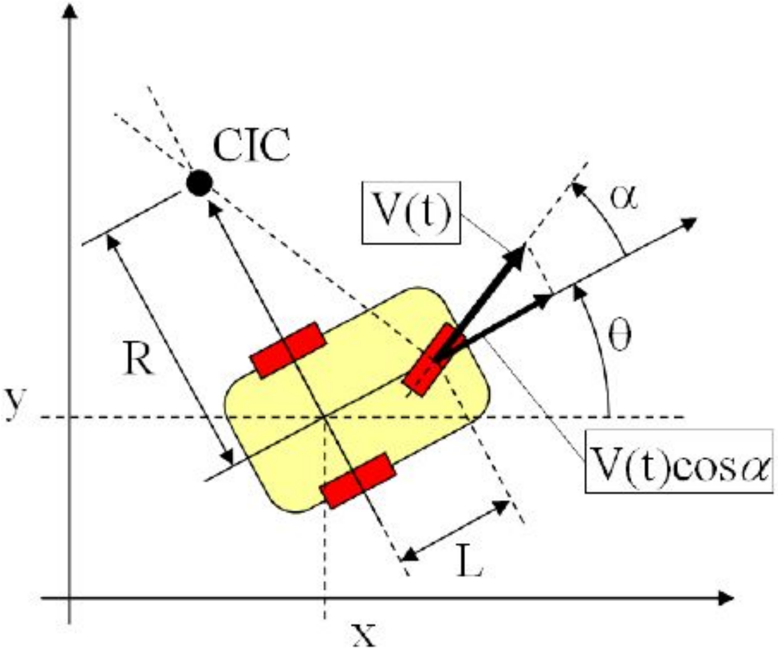
α: 0, 0, 5, 5, 8, 8, 10, 10, 10, 12, 10, 7, 6, 3, -4, -8, -10, -12, -14, -12, -10, -8

Expressions to use for tricycle odometry

- $$\begin{cases} \theta_i = \theta_{i-1} + \frac{V_i}{L} \sin \alpha_i \Delta t \\ x_i = x_{i-1} + V_i \cos \alpha_i \cos \theta_i \Delta t \\ y_i = y_{i-1} + V_i \cos \alpha_i \sin \theta_i \Delta t \end{cases}$$

$$V_i \Delta t = \Delta l_i$$

- Notice that the displacement of the steering wheel is measured with pulses from the encoder

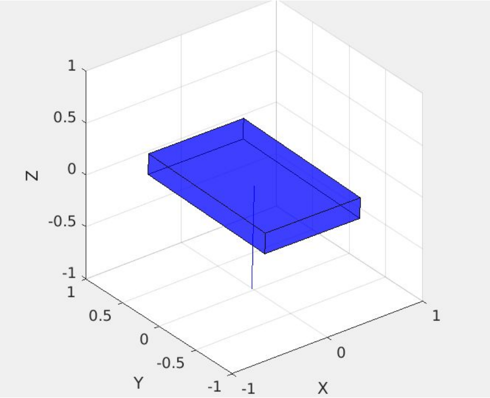


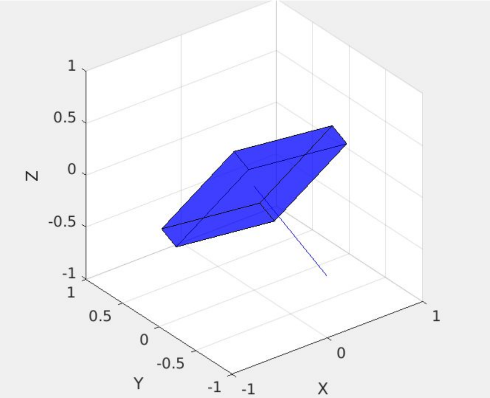
Inertial and positioning logging from smartphone

- Install Matlab Support Package for Android (or IoS) in your computer
- Install Matlab Mobile (Android or IoS) in your smartphone
- Run the application in smartphone and enter your mathworks account
- Select “sensors” in smartphone application
- Activate the sensors you want to monitor (acceleration, etc.)
- In the computer create a mobile device object
  - m=mobiledev
- Check the available data in the fields of **m**
  - Like m.Acceleration, m.AngularVelocity, etc.

3a - Represent continuously the 3D orientation of phone

- Obtain the gravity vector: g=accel/norm(accel)
  - g=[gx gy gz]
- Obtain the orientation angles
  - Pitch: around y → atan2(-gx, sqrt(gy^2+gz^2));
  - Roll: around x → atan2(gy,gz)
- Draw a parallelepiped emulating the smartphone and illustrate its orientation with data from the remote acquisition
  - System may be not very reactive depending on the network conditions
- See next page for drawing suggestions





3a (cont.) - Hints to draw in Matlab

Use a patch with vertices and faces like:

```
% Define the vertices of the smartphone
vertices = [0.5  0.5 -0.5 -0.5  0.5  0.5 -0.5 -0.5   % X
            0.8 -0.8 -0.8  0.8  0.8 -0.8 -0.8  0.8   % Y
            0.1  0.1  0.1  0.1 -0.1 -0.1 -0.1 -0.1]; % Z
% Define the faces of the smartphone
faces = [1 2 6 5   % front
        4 3 7 8   % back
        1 4 8 5   % left
        2 3 7 6   % right
        1 2 3 4   % top
        5 6 7 8];% bottom
% Plot the initial orientation of the model
h=patch('Vertices', vertices', 'Faces', faces,...
        ... 'FaceColor', 'b');
view(3)
```

Inside a while loop, calculate the new vertices and update:

```
% Compute rotated vertices
rotated_vertices= R_roll*R_pitch*
vertices;

% Update plot with rotated vertices
h.Vertices=rotated_vertices';
```

R_roll is a 3D rotation around x
R_pitch is a 3D rotation around y

3b-Integrate data from smartphone for localization

- Create a loop and get angular velocity and acceleration data from smartphone
  - You can obtain some initial data in rest to use as offset to reduce errors during motion
- Integrate angular velocities to try to obtain orientations
- Double integrate accelerations to obtain positioning
- Apply a moving average to signals to reduce noise
  - v(n)=mean(v(n-1), v(n-2), v(n-3) , ..., v(n-N)) . Set N to several different values
- Perform simple trajectories with the smartphone to try to obtain a reasonable estimation of the localization with the inertial system.
- Results may not be as fine as expected!
  - What other tools do you expect to use to improve results?