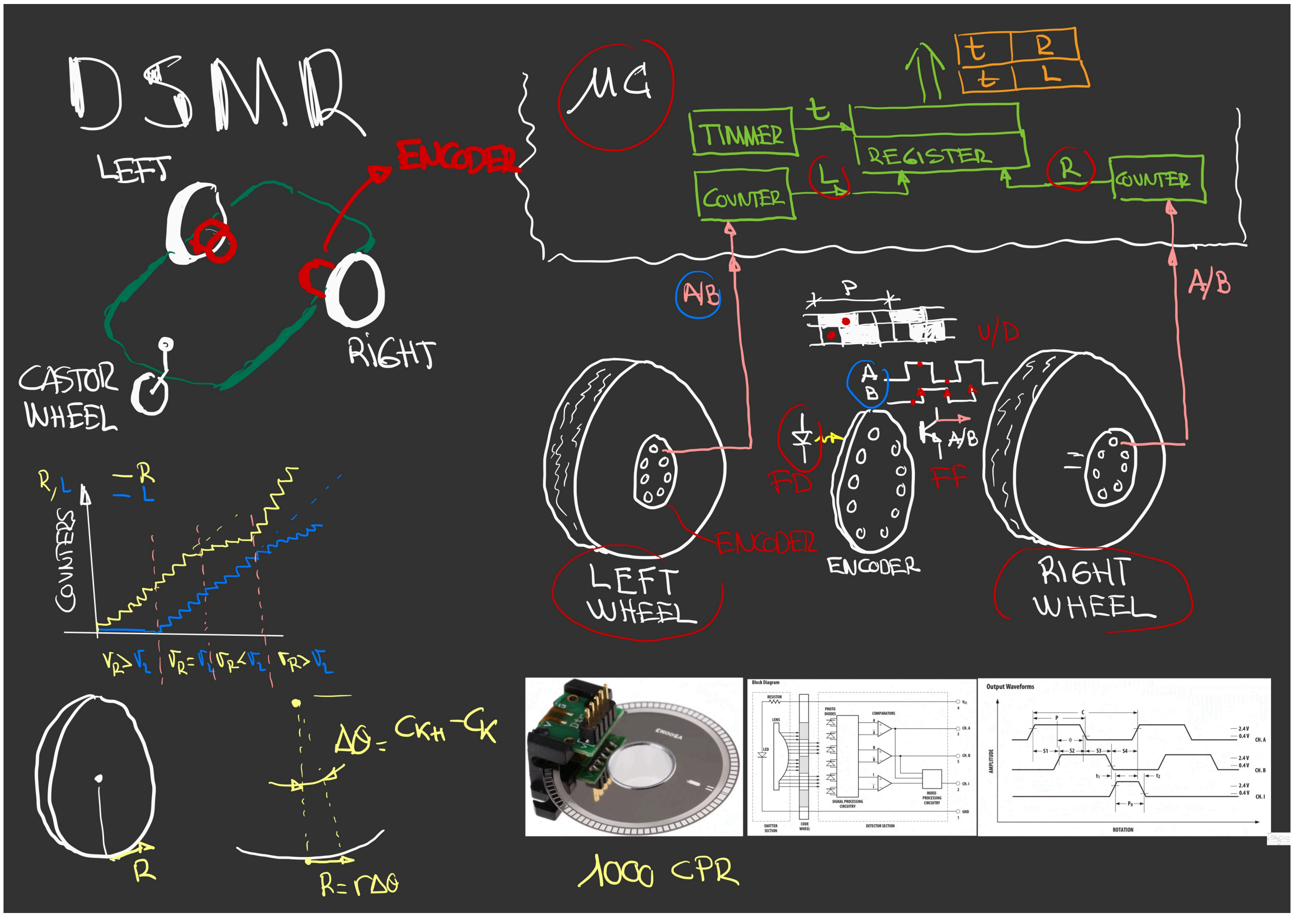
Familiarization with odometry data



Visit: https://en.wikipedia.org/wiki/Rotary_encoder

Table of Contents

Load and visualize data

R_acu & L_acu: The microcontroller counter report; firts column time stamp, second column the total displacement of right and left wheel in meters.

```
R_acu = 3004×2
         0         0
    0.0200         0
    0.0400         0
    0.0600         0
    0.0800         0
    0.1000         0
    0.1200         0
    0.1400         0
    0.1600         0
    0.1800         0
      ⋮

L_acu = 3004×2
         0         0
    0.0200         0
    0.0400         0
    0.0600         0
    0.0800         0
    0.1000         0
    0.1200         0
    0.1400         0
    0.1600         0
    0.1800         0
      ⋮
```

Ts & Tf : Sample time and total experiment time (Tf=Numbers of rows *Ts) in [s]

```
Ts = 0.0200

Tf = 60.0800

Tf = 60.0800

ts = 3003×1
    0.0200
    0.0200
    0.0200
    0.0200
    0.0200
    0.0200
    0.0200
    0.0200
    0.0200
    0.0200
      ⋮
```

r_w & W: Mobile Robot paramenters; wheel radius and distance between wheels in [m]

```
r_w = 0.0947
```

```
W = 0.5200
```

```
W = 0.5200
```

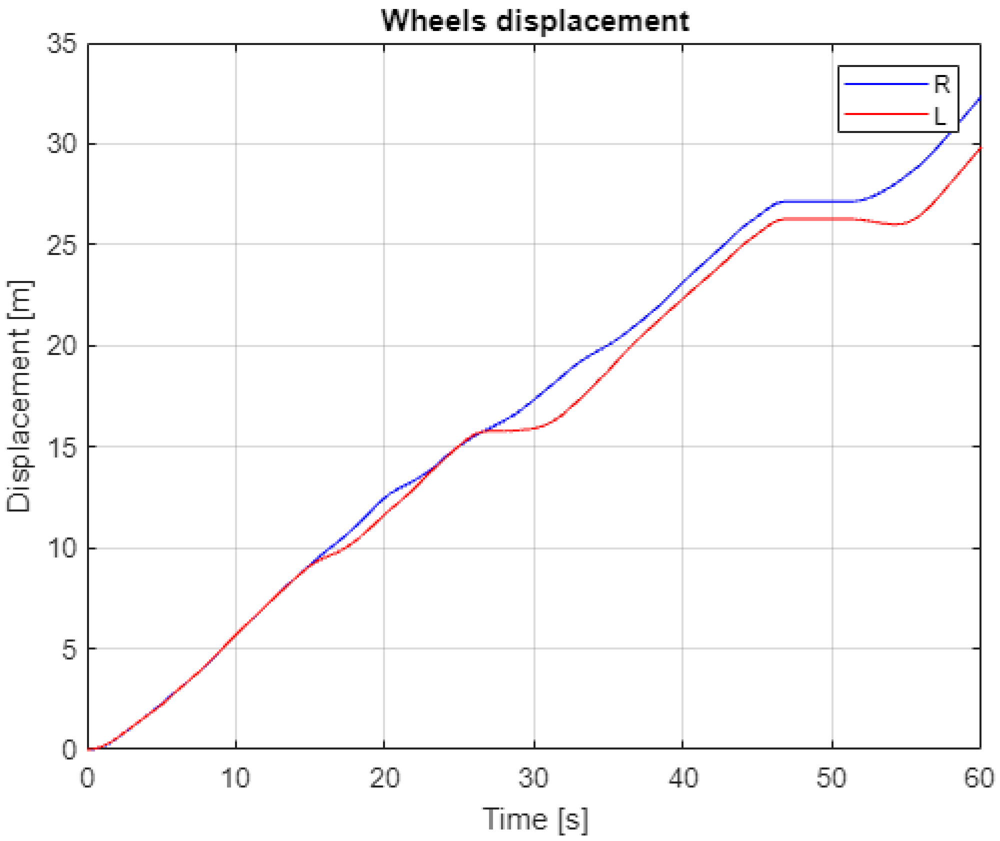
Plotting encoders data with respect time

```
t = 3004×1
         0
    0.0200
    0.0400
    0.0600
    0.0800
    0.1000
    0.1200
    0.1400
    0.1600
    0.1800
      ⋮
```

```
t = 3004×1
         0
    0.0200
    0.0400
    0.0600
    0.0800
    0.1000
    0.1200
    0.1400
    0.1600
    0.1800
      ⋮
```

```
t = 3004×1
         0
    0.0200
    0.0400
    0.0600
    0.0800
    0.1000
    0.1200
    0.1400
    0.1601
    0.1801
      ⋮
```

Total wheel displacements profile



Wheel incremental displacements

It tell us how much displacement did the wheel during a sample time.

$$R_{\mathrm{inc}} = R_{k+1} - R_k$$

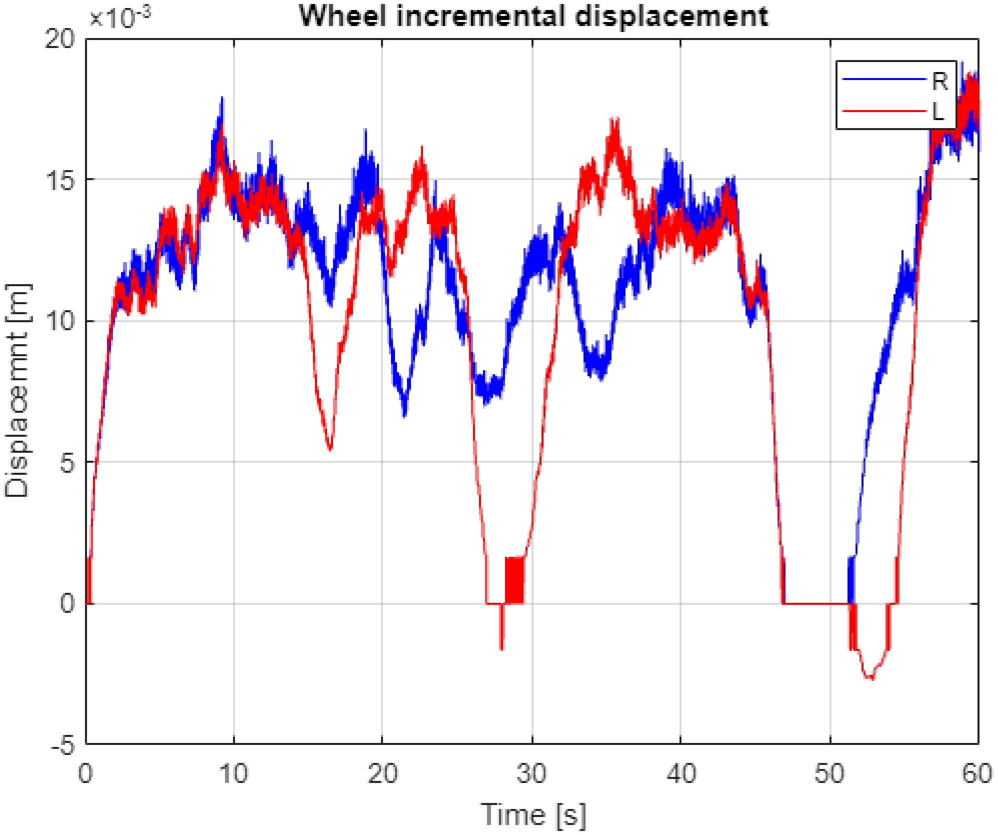
$$L_{\mathrm{inc}} = L_{k+1} - L_k$$

```
R_inc = 3003×1
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         ⋮
```

```
L_inc = 3003×1
         0
         0
         0
         0
         0
         0
         0
```

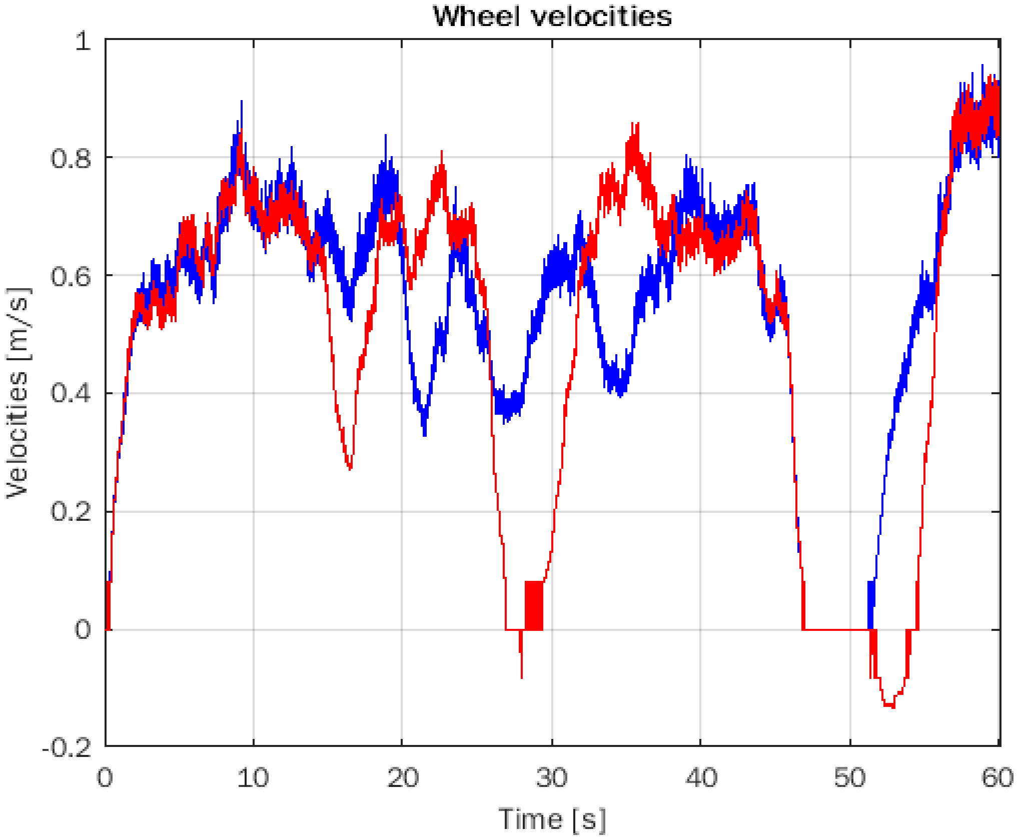
```
0
0
0
⋮
```

Incremental displacements profile



Wheel velocities profile

$$\text{Velocity} = \frac{\Delta\text{displacemts}}{\Delta\text{time}} = \frac{\Delta e}{\Delta t} = \frac{\Delta R}{T_s} = \frac{R_{\text{inc}}}{T_s}$$



Equivalence of Encoder Data

Some time the microcontroller gives wheels increment displacement. To recover total wheel displacement

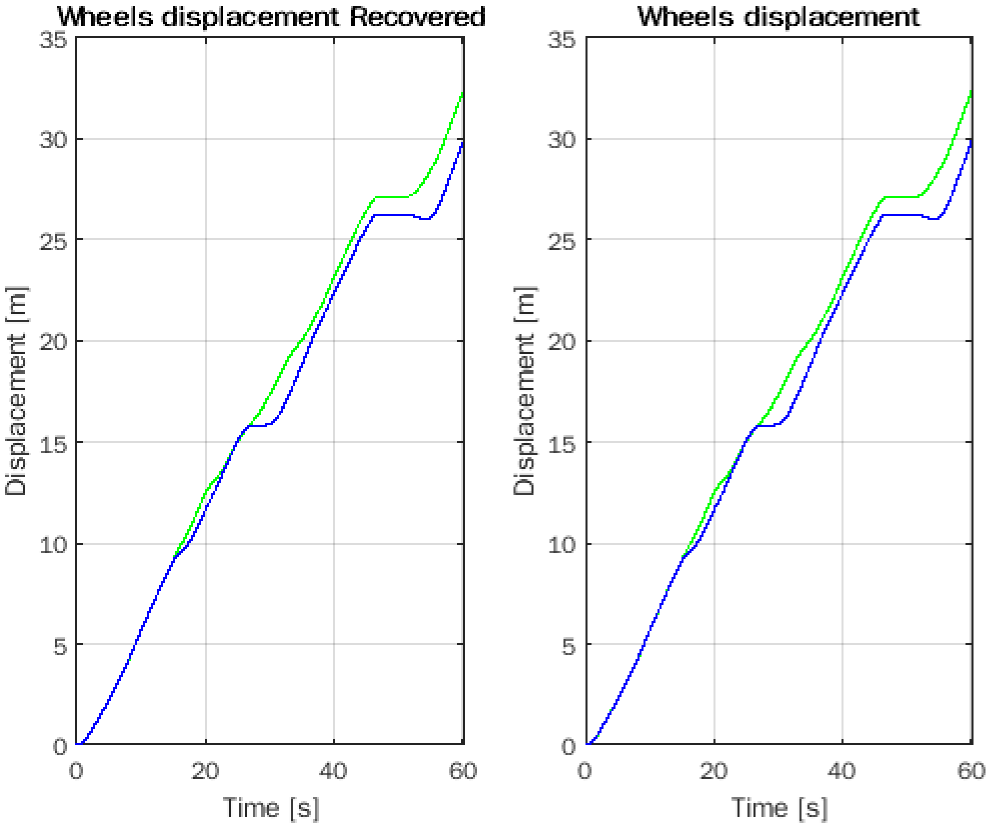
$$R_{\mathrm{acu}_k} = \int_0^{t_k} R_{\mathrm{inc}}(t)\mathrm{dt} \equiv \sum_i^k R_{\mathrm{inc}_i}$$

```
R_ac = 3003×1
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         ⋮

L_ac = 3003×1
         0
         0
         0
         0
         0
         0
         0
         0
         0
```

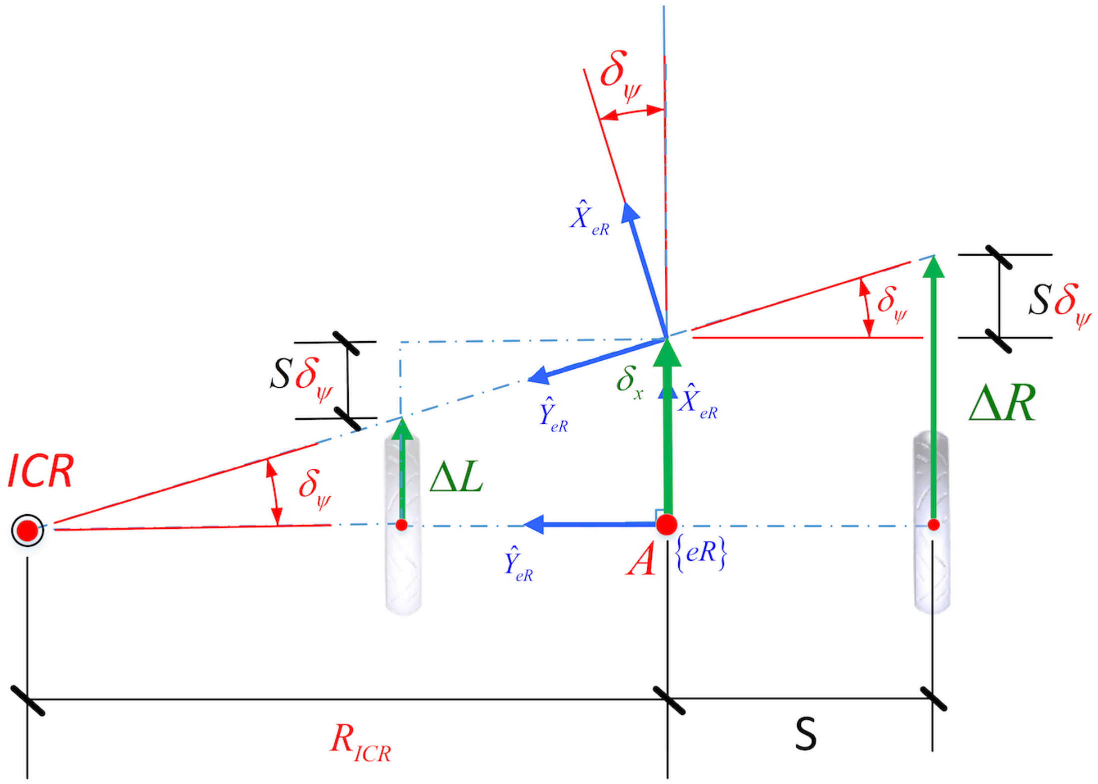
0

⋮

Recovered wheel displacements profile



Odometry

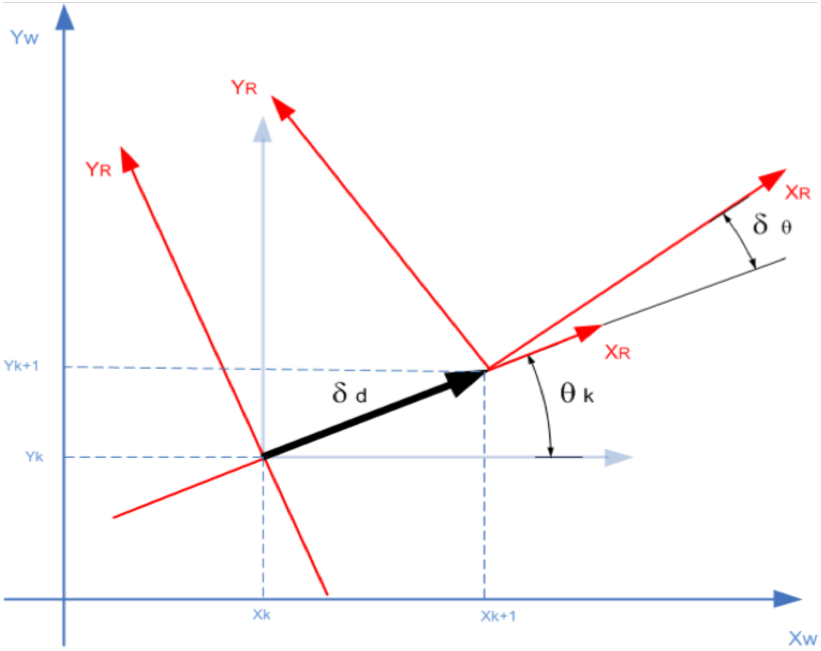


$$\delta_x = \frac{R_{inc} + L_{inc}}{2}$$

$$\delta_\Psi = \frac{R_{inc} - L_{inc}}{2S}$$

```
delta_d = 3003×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮

delta_t = 3003×1
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     ⋮
```

Pose integration



Pose as a Frame description; $\xi_k = \begin{pmatrix} c\theta_k & -s\theta_k & x_k \\ s\theta_k & c\theta_k & y_k \\ 0 & 0 & 1 \end{pmatrix}$

Using post multiplication

Next pose; $\xi_{k+1} = \xi_k \tan sl_x(\delta_d) Rot_Z(\delta_\theta)$

```
  Initial_pose = 4×4
           0    1.0000         0    8.6500
     -1.0000         0         0   17.2000
           0         0    1.0000         0
           0         0         0    1.0000
```

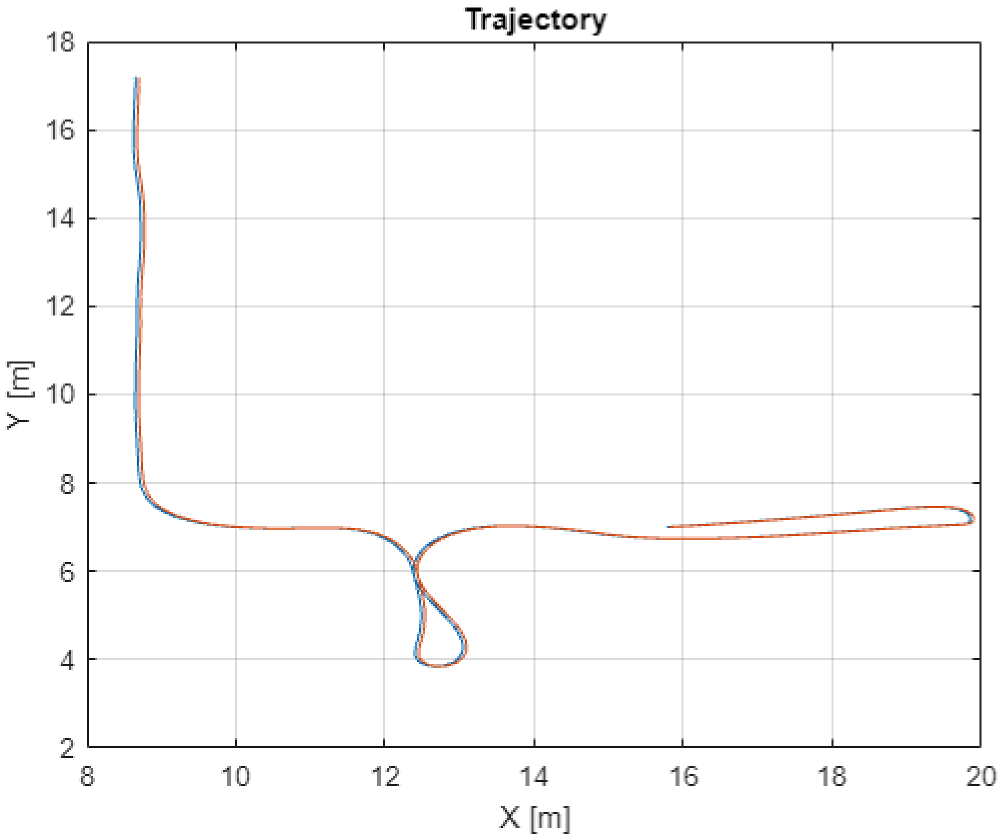
or using

$$\xi_{k+1} = \begin{pmatrix} p_{k+1} \\ \theta_{k+1} \end{pmatrix} = \begin{pmatrix} x_k + \delta_d c\theta_k \\ y_k + \delta_d s\theta_k \\ \theta_k + \delta_\theta \end{pmatrix}$$

```
  Initial_position = 3×1
      8.6500
     17.2000
           0

  Initial_orientation = -1.5708
```

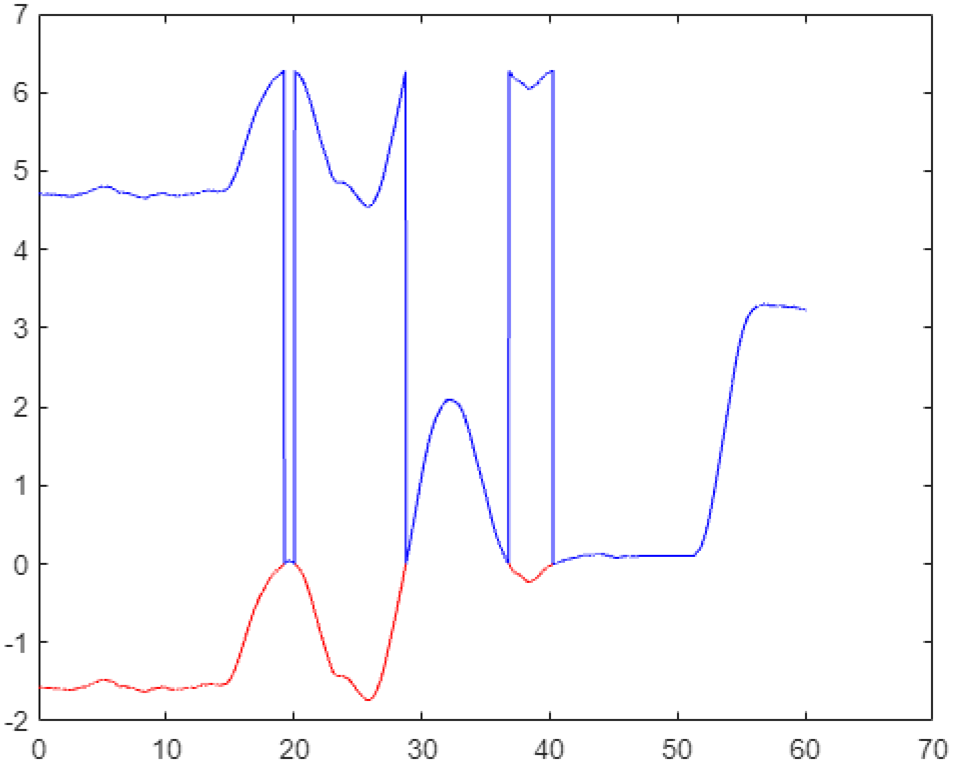
```
x = 8.7000

y = 17.2000

o = -1.5708
```

Displaying trajectory



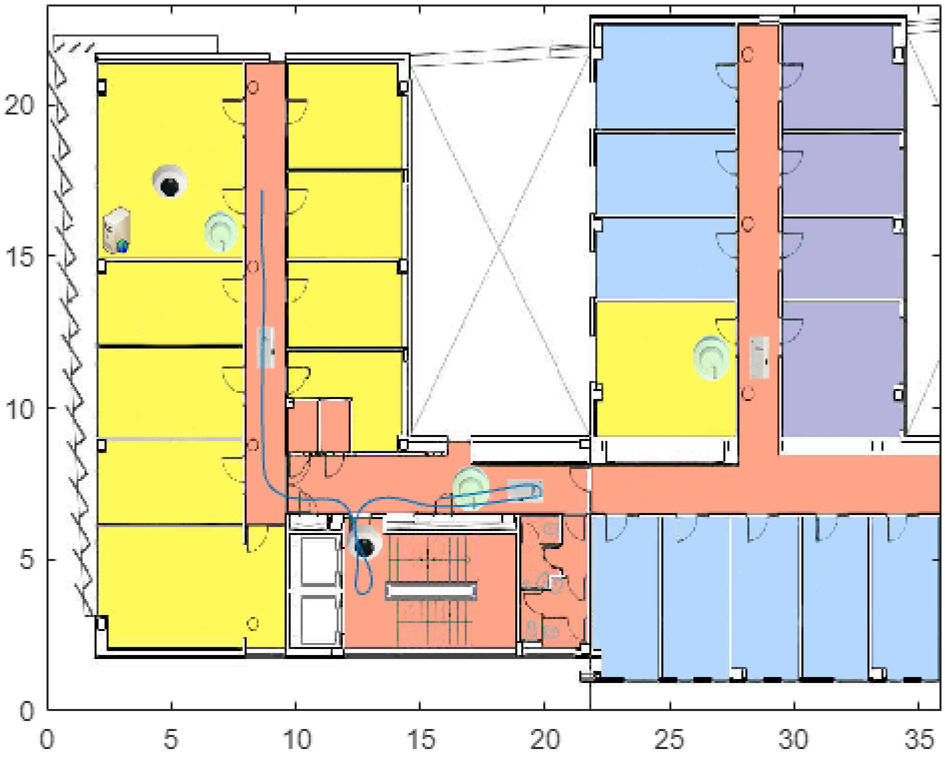
Understanding Orientation

Think about $\sin\left(-\frac{\pi}{2}\right) = \sin\left(\frac{3}{2}\pi\right) = \sin\left(2\pi + \frac{3}{2}\pi\right)$



Plotting the enviroment and trajectory

```
figure1 = 
  Figure (7) with properties:

      Number: 7
        Name: ''
       Color: [1 1 1]
    Position: [671 661 577 433]
       Units: 'pixels'

  Show all properties
```



Uncertanty: Adding noise

$$\delta_d = \frac{R+L}{2} + \nu_d$$

$$\delta_\theta = \frac{R-L}{2S} + \nu_\theta$$

Noise in the odometry displacement

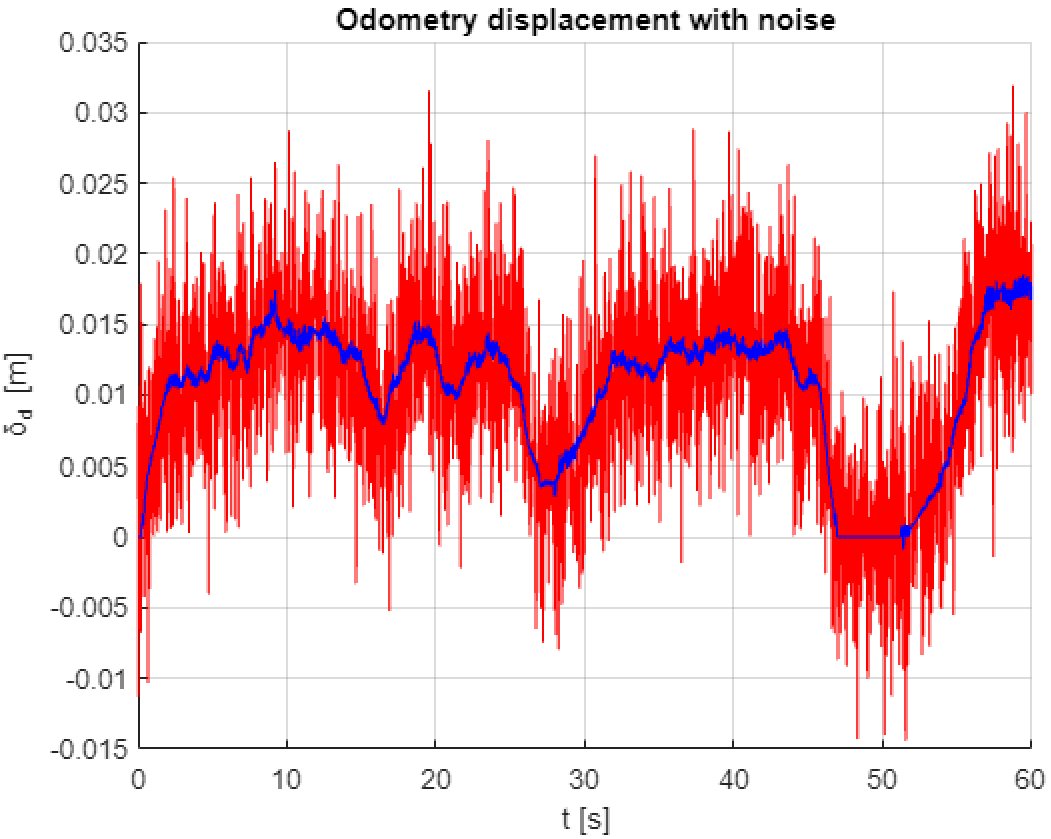
```
delta_d_n = 3003×1
    0.0027
    0.0092
   -0.0113
    0.0043
    0.0016
   -0.0065
   -0.0022
    0.0017
    0.0179
    0.0138
      ⋮
```

Noise in the odometry change of orientation

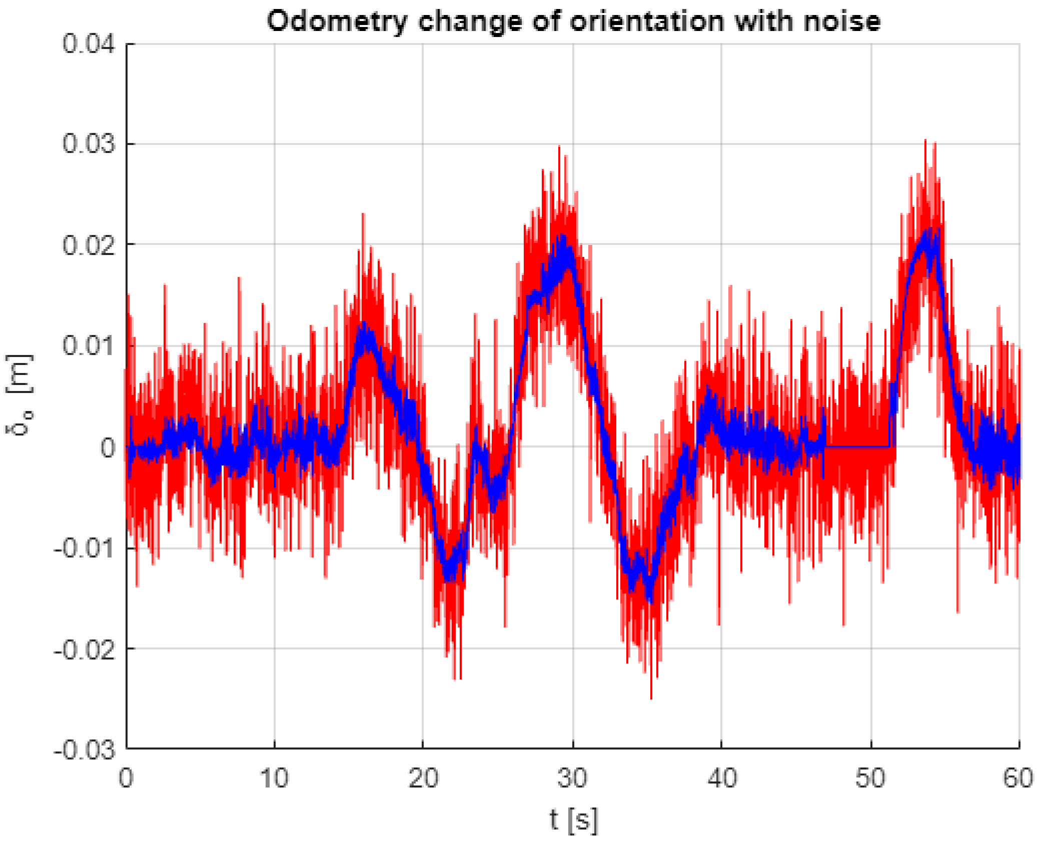
```
delta_t_n = 3003×1
    0.0077
```

```
-0.0031
-0.0035
-0.0071
-0.0005
-0.0005
-0.0016
-0.0042
-0.0083
 0.0151
  ⋮
```

Displacement noise visualization



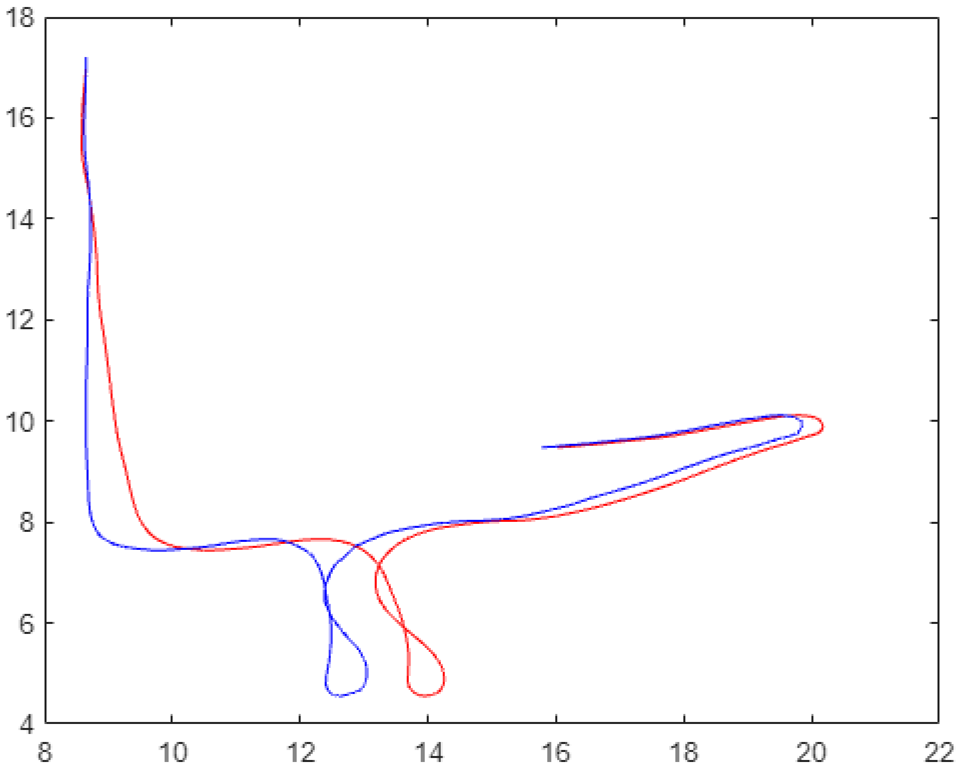
Orientation noise visualization



Pose integration with noise

```
  Initial_pose = 4×4
          0    1.0000         0    8.6500
    -1.0000         0         0   17.2000
          0         0    1.0000         0
          0         0         0    1.0000
```

Comparing trajectories

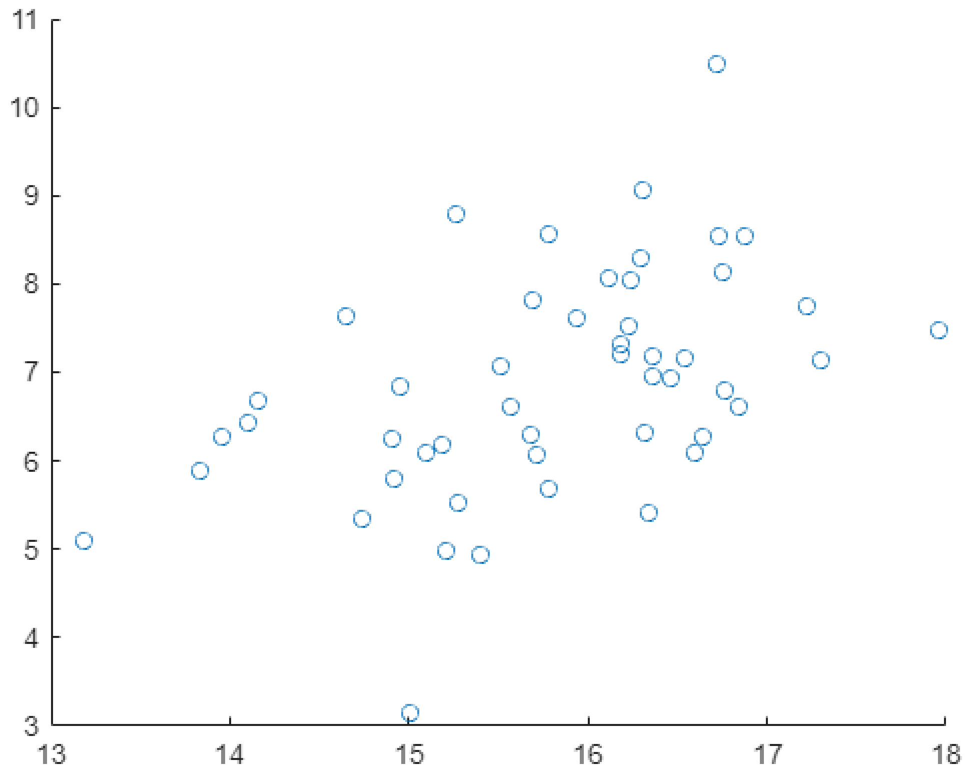


Ellipse error

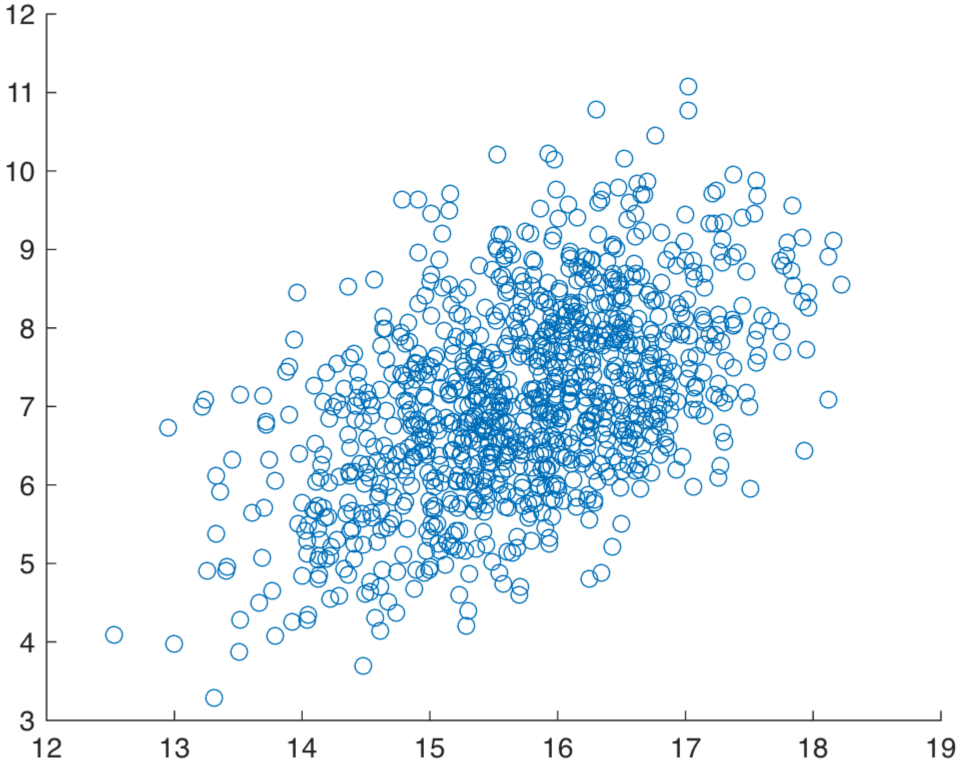
It is of interest to launch many time the dices (our trajectory with noise) and check for the last position and orientation

```
Initial_pose = 4×4
         0    1.0000         0    8.6500
   -1.0000         0         0   17.2000
         0         0    1.0000         0
         0         0         0    1.0000
```

Visualizing the experimental ellipse error of final position



This image is with 1000 launch the dices


12
11
10
9
8
7
6
5
4
3
12
13
14
15
16
17
18
19