Intelligent mobile robotics
Assignment 4

Sami Sellami

April 30, 2019

Task1:

The main objectif of the task is to implement an extended Kalman Filter in the case of robot localization and to design a relationship between the successive pose of a differential drive robot.

1. **The state variable vector of the robot:**
   We choose as state vector for the differential drive robot the x, y positon wrt the world coordiante frame, the orientation θ of the robot frame wrt the world frame 1 and the successive positions of the landmarks

$$x_k = \begin{bmatrix} x \\ y \\ \\ \theta \\ m_x^1 \\ m_y^1 \\ .... \\ m_x^n \\ m_y^n \end{bmatrix}$$

   Where n is the numbre of landmarks

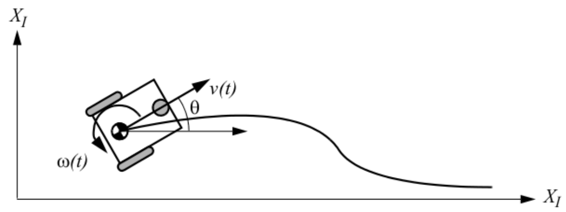

Figure 1: Movement of the differential drive robot

2. **Designing the system model Φ_k:** For a differential-drive robot, the position can be estimated starting from a known position by integrating the movement (summing the incremental travel distances). For a discrete system with a fixed sampling interval Δt , the incremental travel distances are:

$$\begin{aligned} \Delta x &= \Delta s \cdot \cos(\theta + \Delta\theta/2) \\ \Delta y &= \Delta s \cdot \sin(\theta + \Delta\theta/2) \\ \Delta\theta &= \frac{\Delta s_r - \Delta s_l}{b} \quad (1) \\ \Delta s &= \frac{\Delta s_r + \Delta s_l}{2} \quad (2) \end{aligned}$$

   Where : $\begin{cases} \Delta x, \Delta y, \Delta\theta \quad \text{path tavelled in the last sampling interval} \\ \Delta s_l, \Delta s_r \quad \text{are the travelled distance for the left and right wheel respectively} \\ b \quad \text{is the distace between two wheels of the differential drive robot} \end{cases}$

   Thus we get the motion model of the diffrential drive robot:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta s \cdot \cos(\theta_{k-1} + \Delta\theta/2) \\ \Delta s \cdot \sin(\theta_{k-1} + \Delta\theta/2) \\ \Delta\theta \end{bmatrix}$$

   Using the relations 1 and 2 we find:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cdot \cos(\theta_{k-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \cdot \sin(\theta_{k-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} = f(x, y, \theta)$$

The control input is the actuation present on the wheels, however we consider that the motion velocity of the robot is constant so the input model is given by $u = [\Delta s_l \quad \Delta s_r] = [v_l \cdot \Delta t \quad v_r \cdot \Delta t]$

Regarding the positions of the landmark, we consider that they dont change with respect to the global frame so:

$$
\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ m_x^1 \\ m_y^1 \\ .... \\ m_x^n \\ m_y^n \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \\ m_x^1 \\ m_y^1 \\ .... \\ m_x^n \\ m_y^n \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cdot \cos(\theta_{k-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \cdot \sin(\theta_{k-1} + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \\ 0 \\ 0 \\ ... \\ 0 \\ 0 \end{bmatrix} = f(x, y, \theta)
$$

If we linearize the system with respect to the state vector we obtain the expression of $\Phi(x_k)$

$$
\Phi(x_k) = \nabla_x f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta + \Delta\theta/2) & 0 & 0 & 0 & ... & 0 \\ 0 & 1 & \Delta s \cos(\theta + \Delta\theta/2) & 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & ... & ... & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & ... & 1 \end{bmatrix}
$$

3. **Deriving the expressions of x_k^- and P_k^-:** The expression of the a priori estimate of state vector in the extended Kalman filter is given by

$$
x_k^- = f(x_{k-1}, u_k)
$$

The update covariance matrix of the extended Kalman filter is given by;

$$
P_k^- = \Phi_k P_k \Phi_k^T + Q
$$

Where P_{k-1} is the covariance of the previous robot state x_{k-1}
and Q is the covariance of the noise associated to the motion model

4. **Obtaining the measurement model:**
We suppose that the position of the landmarks are given by p_x^i and p_y^i, the position and orientation of the landmark relative to the robot is

$$
\begin{cases} r^i = \sqrt{(m_x^i - x)^2 + (m_y^i - y)^2} \\ \phi^i = \arctan(\frac{m_y^i - y}{m_x^i - x}) - \theta \end{cases}
$$

The measurement model is given by the expression :

$$
h^i(x_k^-) = \begin{bmatrix} \sqrt{(m_x^i - x)^2 + (m_y^i - y)^2} \\ \arctan(\frac{m_y^i - y}{m_x^i - x}) - \theta \end{bmatrix}
$$

Its linearization with respect to the state variables give the matrix H^i

$$
H^i = \begin{bmatrix} \frac{-m_x^i + x}{\sqrt{(m_x^i - x)^2 + (m_y^i - y)^2}} & \frac{-m_y^i + y}{\sqrt{(m_x^i - x)^2 + (m_y^i - y)^2}} & 0 & \cdots & \frac{m_x^i - x}{\sqrt{(m_x^i - x)^2 + (m_y^i - y)^2}} & \frac{m_y^i - y}{\sqrt{(m_x^i - x)^2 + (m_y^i - y)^2}} & \cdots \\ \frac{m_y^i - y}{(m_x^i - x)^2 + (m_y^i - y)^2} & \frac{x - m_x^i}{(m_x^i - x)^2 + (m_y^i - y)^2} & -1 & \cdots & \frac{-m_y^i + y}{\sqrt{(m_x^i - x)^2 + (m_y^i - y)^2}} & \frac{m_x^i - x}{\sqrt{(m_x^i - x)^2 + (m_y^i - y)^2}} & \cdots \end{bmatrix}
$$

Where i is referring to the landmark number i
We assume that we have three landmarks positioned like this:

$$\begin{aligned} p_x^1 &= 5 & p_y^1 &= 10 \\ p_x^2 &= 10 & p_y^2 &= 5 \\ p_x^3 &= 15 & p_y^3 &= 15 \\ p_x^3 &= 15 & p_y^3 &= 5 \\ p_x^3 &= 20 & p_y^3 &= 10 \end{aligned}$$

like that we will the final H matrix equal to

$$H = \begin{bmatrix} H^1 \\ H^2 \\ H^3 \\ H^4 \\ H^5 \end{bmatrix}$$

5. **Writing the assumptions abouts the errors of the filter:**
The update covariance matrix of the extended Kalman filter is given by;

$$P_k^- = \Phi_k P_k \Phi_k^T + Q$$

Where P_{k-1} is the covariance of the previous robot state x_{k-1} and Q is the covariance of the noise associated to the motion model

$$Q = \nabla_u f \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix} \nabla_u f^T$$

where k_r and k_l are error constants representing the nondeterministic parameters of the motor drive and the wheel floor interaction, to obtain this we make the assumption that error of the individually driven wheels are independant and that the variance of the error are proportional the absolute value of the travelled distace Δs_r, Δs_l

$$\nabla_u f = \begin{bmatrix} \frac{1}{2}\cos(\theta + \Delta\theta/2) - \frac{\Delta s}{2b}\sin(\theta + \Delta\theta/2) & \frac{1}{2}\cos(\theta + \Delta\theta/2) + \frac{\Delta s}{2b}\sin(\theta + \Delta\theta/2) \\ \frac{1}{2}\sin(\theta + \Delta\theta/2) + \frac{\Delta s}{2b}\cos(\theta + \Delta\theta/2) & \frac{1}{2}\sin(\theta + \Delta\theta/2) - \frac{\Delta s}{2b}\cos(\theta + \Delta\theta/2) \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix}$$

We suppose also that the covariance matrix P_0 is known and equal to

$$P_0 = \begin{bmatrix} 0.2 & 0 & 0 & ... & 0 \\ 0 & 0.2 & 0 & ... & 0 \\ 0 & 0 & 1 & ... & 0 \\ 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & ... & 1 \end{bmatrix}$$

also in our implementation, we approximate for simplicity the matrix Q as identity multiplied by a certain constant

$$Q = \begin{bmatrix} 1 & 0 & 0 & .. & 0 \\ 0 & 1 & 0 & .. & 0 \\ 0 & 0 & 1 & .. & 0 \\ 0 & 0 & 0 & .. & 1 \end{bmatrix}$$

After aquiring the sensors data , ie the position and orientation of the three landmarks, we need to compute the covariance matrix associated with each landmark, it can be approximated as

$$R^i = \begin{bmatrix} \sigma_{rr}^i & 0 \\ 0 & \sigma_{\phi\phi}^i \end{bmatrix}$$

We suppose that the measurement of distance is independant from the orientation measurement , and for our code we take the following parameters $\sigma_{rr} = 0.3$ and $\sigma_{\phi\phi} = 0.1$

Thus the total covariance matrix for the whole set of landmarks should be like:

$$R = \begin{bmatrix} R^1 & 0 & 0 & & \\ 0 & R^2 & 0 & 0 & 0 \\ 0 & 0 & R^3 & 0 & 0 \\ 0 & 0 & 0 & R^4 & 0 \\ 0 & 0 & 0 & 0 & R^5 \end{bmatrix}$$

6. **Drawing P_k changes over time for two path(straight line and turn):** Applying the extended kalman Filter to our system means following the equation depicted below:

$$\begin{aligned} x_k^- &= f(x_{k-1}, u_k) && (3) \\ P_k^- &= P_k \Phi_k P_k^T + Q && (4) \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^- 1 && (5) \\ x_k &= x_k^- + K_k(z_k - h(x_k^-)) && (6) \\ P_k &= (1 - K_k H_k) P_k^- && (7) \end{aligned}$$

After implementing the code and plotting the changes of the covariance matrix P over time we obtain the following graphs:

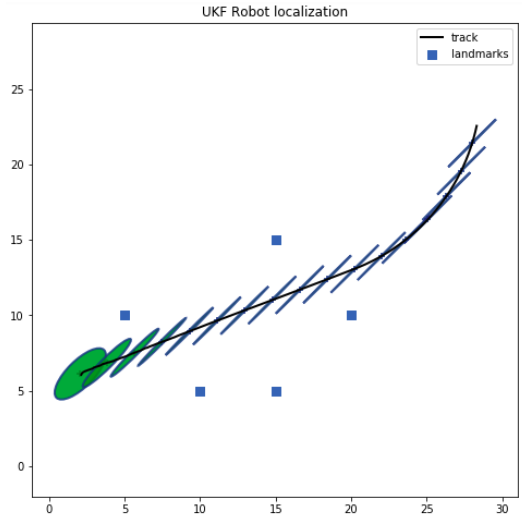

Figure 2: Pk changes over time

To obtain this graph we set the commands of the robots ie the velocities of the right and left wheel equal in the first part of the trajectory and then we set the right wheel velocity slightly greater than the left one, in a way that robot take a turn like shown in the above picture.

Task2:

In this task we are provided with a set of images where we need to locate a bounding box for an object that is moving, and extract some features points , the following figure shows the result for one keypoints that is plotted for 5 consecutive images

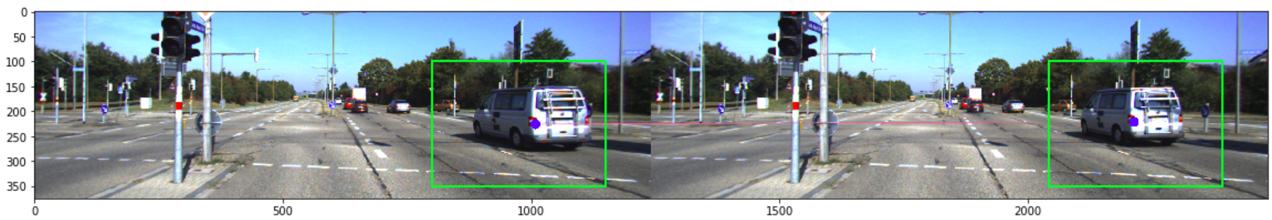

Figure 3: one keypoints in five consecutive images

Then we suppose that the position of the robot is in the position of the keypoint aforementioned, after that, we need to extract landmarks to inject them in our EKF implementation, for this we extract new keypoints from the background and we consider them as landmarks, the following graph shows the keypoints extraction:

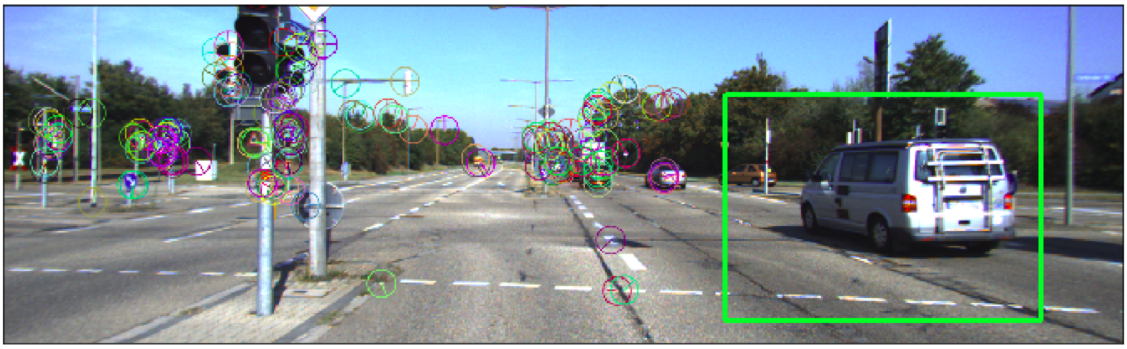

Figure 4: landmark extraction

After injecting all this informations in our EKF implementation (initial position of the robot and the landmarks) we obtain the following graph showing of the trajectory of the robot(we suppose its moving on a straight line):

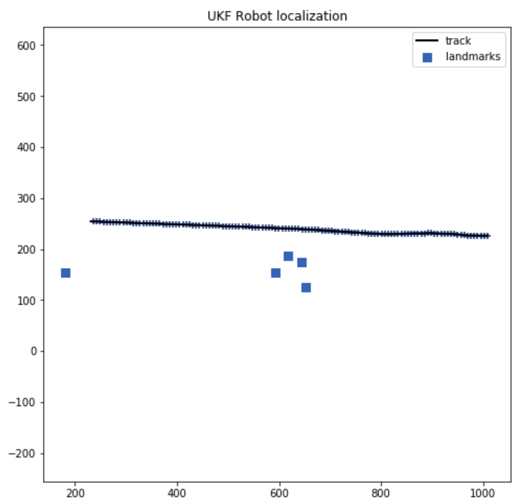

Figure 5: Trajectory estimation using extended kalman filter