CSE 276A - Homework 3

Chhabra, Rachit (rachhabra@ucsd.edu)
Department of Mechanical and Aerospace Engineering, University of California San Diego

I. PROBLEM STATEMENT

The objective of Homework 3 is to implement a version of the Simultaneous Localization and Mapping (SLAM) technique on the robot and evaluate its performance. In this project, first of all, Odometry Localisation is done. Then using the camera images and April Tags as landmarks, the mapping is carried out. Finally, the two parts are combined, and SLAM is carried out. The final output is the updated trajectory and landmark map of the robot's environment, and it is finally plotted.

II. SETUP

The setup of the environment is follows:

1. We have 8 April Tags which would be considered as unkown landmarks during the SLAM problem. Initially, we do not know the number and the posiiton of landmarks. Fig. 1 shows a sketch of the ground-truth map in the top-down view layout.

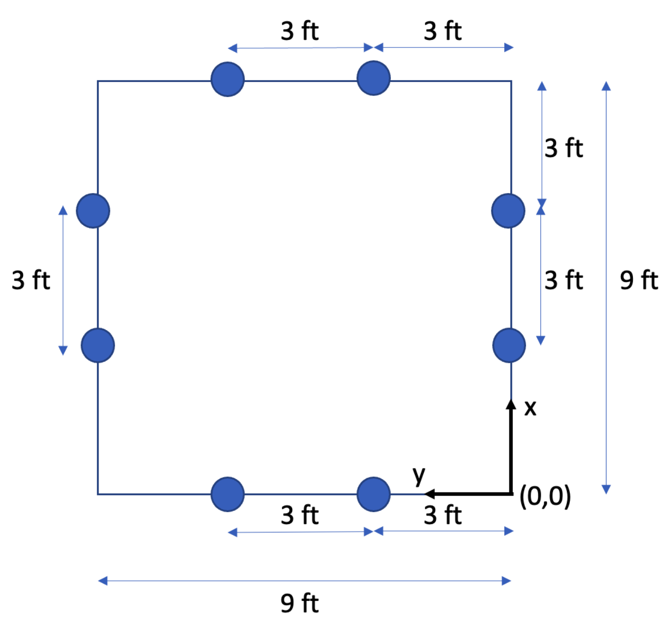

Fig. 1. Ground Truth Map

2. Each landmark has a distinct ID (April Tag id) and this can be used to distinguish to landmarks seen.
3. We do not know the initial starting position of the robot.

III. PROBLEM FORMULATION

The first step of the homework is to build a Kalman Filter. As our motion model and observation model are non linear, we should ideally me constructing an Extended Kalman Filter. But due to the simplicity of Kalman Filter, we choose Kalman Filter to perform SLAM. We move the robot at a very low speed to make the models as close to linear as possible so that the Kalman Filter which we are using gives us good approximations and in turn an acceptable result.

We consider the state vector s_t at time t which consists of the mean position of the robot, and the mean positions of the landmarks which we have seen at least once.

$$s_t = \begin{pmatrix} x_r \\ y_r \\ \theta_r \\ x_{1w} \\ y_{1w} \\ \theta_{1w} \\ \vdots \end{pmatrix}$$

where all the coordinates are in world frame coordinates.

The measurements z_t is the position of the landmarks visible at time t. The coordinates of the landmarks are in robot frame.

$$z_t = \begin{pmatrix} x_{1r} \\ y_{1r} \\ \theta_{1r} \\ x_{2r} \\ y_{2r} \\ \theta_{2r} \\ \vdots \end{pmatrix}$$

The motion model is given by:

$$s_t = Fs_{t-1} + Gu_t + w_t$$

where s_{t-1} is the state vector at time $t-1$, u_t is the control given to the robot and w_t is the noise in the motion model with $w_t \sim \mathcal{N}(0, W)$

F and G are system matricies which relates the state at time t with the previous time step.

Here,

$$F = \mathcal{I}_{nxn}$$

which is the identity matrix of size n which is the length of s_t and

$$G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ - & - & - \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \end{pmatrix}_{(no\ \ of\ \ states)\ \text{x}\ 3}$$

The G matrix ensures that the effect of control is only on the robot position and not the landmark positions.

The observation model is given by

$$z_t = Hs_t + v_t$$

v_t is the noise in the motion model with $w_t \sim \mathcal{N}(0, V)$ and H is the matrix which relates the landmark observations with the state vector. The formulation of H is discussed in the next section.

<u>INITIALISATION</u>

The state s_t is initialed as $(0, 0, 0)$. This means, where ever we start the robot, we consider that as the origin in world frame. We initialise matrices V and W as

$$V = 0.1\ \text{x}\ I_{\text{shape of H}}$$

$$W = 0.001\ \text{x}\ I_{\text{shape of state}}$$

We give more covariance to the measurement noise as we want to make the model resiliant to the noise in sensor data.

The covariance is initialiased as a 3 x 3 matrix with 0.01, 0.01, 0.1 along the main diagonal. The covaraiance matrix keeps on expanding whenever we see a new landmark and it's initialised in the same way. There is no correlation initialisation between any two landmarks or state, i.e., we only initialise the values on the main diagonal.

IV. FORMULATION OF H

The H matrix relates the landmark observations with the state vector using the equation,

$$z_t = Hs_t$$

Note, that this the the observation model with no noise. We use this H matrix to calculate the Kalman Gain, update the mean and the covariance in the update step.

The dimensions of the H matrix is dynamic. It depends on the no. of obervations seen at current time t. It's dimensions are (3 x no of observations) x (length of state vector). So, let's assume we see 3 observations at the first time, the dimesions would be (9 x 12). At the second instance, we only see the previously seen two landmarks. Now the dimensions of H would be (6 x 12). As the state vector is (12 x 1).

Let's assume landmark 1 with world coordinates x_1, y_1, θ_1 is visible to the robot with world coordinates x_r, y_r, θ_r. We use the following equations using coordinates transformation to convert the observations which are in robot frame coordinates into world coordinates.

$$x_{1r} = cos\theta_r(x_1 - x_r) - sin\theta_r(y_1 - y_r)$$
$$y_{1r} = sin\theta_r(x_1 - x_r) + cos\theta_r(y_1 - y_r)$$
$$\theta_{1r} = \theta_1 - \theta_r$$

Using these equations we can build sub matrices in the H matrix.

Below is an example of an H matrix where only two observations have been seen until now and currently only the second observation is visible. Thus you can see the middle three columns are zeros. The H matrix depends on in what order is the observations seen for the first time.

$$H = \begin{pmatrix} -cos\theta_r & sin\theta_r & 0 & 0 & 0 & 0 & cos\theta_r & -sin\theta_r & 0 \\ sin\theta_r & -cos\theta_r & 0 & 0 & 0 & 0 & sin\theta_r & cos\theta_r & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

When we see the landmark for the first time, we add the it's world coordinates in the state matrix, and increase the size of the covariance matrix and initialise the covariance matrix as 0.01, 0.01, 0.1 along the diagonal. When the landmark disappears, it no longers appear in the H matrix, but still continues to appear in the state matrix. Thus, once the landmark disappears, we do not update that landmark. Finally when the landmark reappears, we do add it to the H matrix to the position according to the first time we saw the landmark. We don't add it again to the covariance/ state matrix and do not reinitialise it's values.

The major tradeoffs are V and W as those are the hyperparameters and we have tuned those for better results. Sqaure size = 1 meter because it is sufficiently big enough so that we can see many landmarks at the same time.

V. RESULTS AND ANALYSIS

Following is the graph for square trajectory of the robot. The blue dots show the estimated landmark position in the world frame, where the orange dot in the origin.

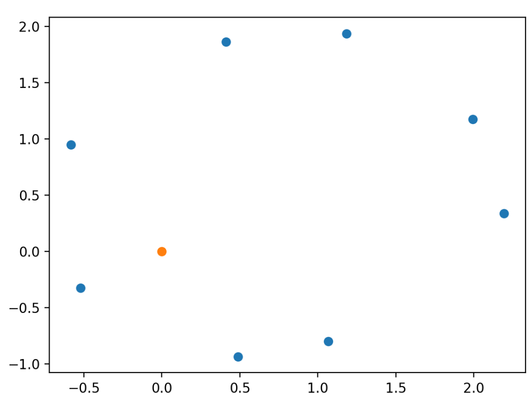

Fig. 2. Landmark Mapping

Average Error = 0.156m

Even though I didn't show the results for double square, prelimniary results show that we get better results and less average error when we run it twice. Also, moving it in a octagon would give us better results due to the longer distance travelled and seeing the landmarks from different thetas which wasnt there in square trajectory.