This assignment consists of three parts.

Please use Python for implementation using only standard python libraries (e.g., numpy, matplotlib etc.). Please do not use any third party libraries/implementations for algorithms.

Please prepare a report to accompany your implementation. The report should contain responses to questions and the desired plots/graphs. Briefly describe the key findings/insights for the graphs. Ensure the reproduction of graphs (modulo probabilistic execution) for your submission.

Additional readings: Artificial Intelligence: A Modern Approach (Ch. 15) and Probabilistic Robotics (Ch. 2 and Ch. 3).

Please refer to the submission instructions on the course webpage.

1. Consider an airplane flying in a (x, y)-plane. A noisy sensor (e.g., GPS positions) provides measurements $z_t = [x'_t, y'_t]$. The plane is controlled by providing velocity increments $u_t = [\delta\dot{x}_t, \delta\dot{y}_t]$, which get added to the velocity components $\dot{x}_t$ and $\dot{y}_t$ at each time step. The uncertainty in the motions is characterized by a presence of Gaussian noise $\epsilon_t \sim \mathcal{N}(0, R)$. Similarly, the measurement uncertainty is characterized by presence of Gaussian noise $\delta_t \sim \mathcal{N}(0, Q)$. Our goal is to estimate its positions $[x_t, y_t]$ and velocities $[\dot{x}_t, \dot{y}_t]$ at time instant t from noisy observations $[x'_t, y'_t]$.

   (a) Implement the motion model for this problem. Initially, assume that the control inputs are both zero. Assume the noise parameters as: $\sigma_{rx} = 1.0$, $\sigma_{ry} = 1.0$, $\sigma_{r\dot{x}} = 0.01$ and $\sigma_{r\dot{y}} = 0.01$ forming the covariance matrix R as $diag(\sigma_{rx}^2, \sigma_{ry}^2, \sigma_{r\dot{x}}^2, \sigma_{r\dot{y}}^2)$ where $diag$ denotes the diagonal elements of R with off-diagonals as zero. Next, implement the observation model for this problem. Assume that the observation noise is distributed as an isotropic Gaussian with a standard deviation of 10. Simulate the motion and the sensor models for $T = 200$ time steps. Plot the actual trajectory and the observed trajectory of the vehicle.

   (b) Implement a Kalman filter for the problem to estimate the vehicle state given the assumptions above. Please formally write down the model for estimation. Let $\hat{x}_t$ denote the estimated state at time t. Implement a control policy where the velocity $\delta\dot{x}_t$ varies as a sine wave and the velocity $\delta\dot{y}_t$ varies as a cosine wave. Assume $(0, 0)$ as the vehicle's starting position. Assume a prior belief over the vehicle's initial state has a standard deviation of 0.01 for each state variable.

   (c) Plot the actual trajectory $[x_t, y_t]$, the noisy observations $[x'_t, y'_t]$ and the trajectory estimated by the filter $[\hat{x}_t, \hat{y}_t]$. Additionally, plot the uncertainty ellipses for the estimated trajectory $[\hat{x}_t, \hat{y}_t]$. An uncertainty ellipse denotes the locus of points that are one standard deviation away from the mean.

   (d) Assume that the sensor observations drop out at time instants $t = 10$ and $t = 60$ for a period of 20 time steps and are re-acquired after that period. Simulate and show the evolution of uncertainty in the vehicle's $[x_t, y_t]$ position by plotting the uncertainty ellipses. Explain your findings.

   (e) Plot the estimated velocities $[\hat{\dot{x}}_t, \hat{\dot{y}}_t]$ and the true velocities of the vehicle $[\dot{x}_t, \dot{y}_t]$. Briefly explain if the estimator can or cannot track the true values.

   (f) Simulate a second vehicle in the environments with linear-Gaussian motion and sensor models (as above) with different initial state estimates and noise characteristics. Let a and b index the two vehicles. The sensor receives two sets of measurements, $z_t^1 = [x_t'^1, y_t'^1]$ and $z_t^2 = [x_t'^2, y_t'^2]$. Estimate the latent states x_t^a and x_t^b at time t for the two vehicles. The estimator requires a strategy for associating observations z_t^1 and z_t^2 with the latent states x_t^a and x_t^b, known as the *data association* problem. Implement a data association strategy and study its behavior in your simulation. Scale your data association strategy to more number of agents. Evaluate your approach on (4-5) agents.

2. In this part, we extend the problem setup of the previous question (in the single agent setting) by incorporating an additional observation.

   Assume that there are certain landmarks (e.g., air traffic control towers) at the following known locations in the environment: $(100, 100)$, $(-100, -100)$, $(-100, 100)$, $(100, -100)$ and $(0, 0)$. The aircraft can measure the Euclidean distance (range) to a landmark when its true position within a certain range of the landmark and the measurement is corrupted with Gaussian noise $\eta_t \sim \mathcal{N}(0, S)$. At any instant, the agent receives a single measurement from the nearest landmark and the measurement can be uniquely associated with the corresponding landmark. Note that the landmark-distance observation is in addition to the GPS-positions that the agent is receiving. The remaining problem setup is considered same as the previous question.

   (a) Formally describe how the additional *landmark*-distance observation can be incorporated in your estimator developed in the last question.

   (b) Extend your simulation to account for the landmark observations as the agent moves through the environment. Assume an range of 30m for observing a landmark. Assume isotropic Gaussian noise with standard deviation of 0.01, 10 and 1 in the motion model, GPS-position observations and landmark-distance observations respectively.

   (c) Simulate at least 800 steps with the filter updates at 1Hz with the agent starting at $(30, -70)$ with a velocity of 4 units with a 0.3 radian heading direction. Plot the true trajectory of the vehicle and the trajectory estimated by your filter. Overlay the uncertainty ellipses for the estimated trajectory and observe how the ellipse changes as the agent comes close to a landmark.

   (d) Next, increase and decrease the uncertainty in the landmark measurements in relation to the uncertainty in the position measurements (as in the previous question). Vary the standard deviation in the landmark observations as 1 and 20 and explain your observations.

   (e) Add an additional landmark in the environment and observe the impact on localization performance.

3. Consider the problem of localizing a robot in a grid world using noisy sensor observations. The robot's environment is a grid of size (X, Y) discretized with grid cells of size 1×1, see figure below. The dark-shaded grid cells represent walls and hence represent infeasible regions. The feasible grid cells are shown in white.

   The robot's action space is discrete with four actions: *MoveNorth*, *MoveSouth*, *MoveEast* or *MoveWest* that move the robot to an adjacent (feasible) grid cell. Assume that the probability of action selection is uniformly distributed among the set of actions that lead the robot to a feasible grid cell. Actions that lead to infeasible grid cells have zero likelihood. It is assumed that there is at least one feasible action for any grid cell, i.e., the a grid cell is not enclosed on all sides with an infeasible grid cells and that there is an enclosing wall in the grid.

   The robot has four noisy sensors pointing in the $(N)orth$, $(S)outh$, $(E)ast$ or $(W)est$ directions. The sensors report a discrete binary observation that a wall is *Close* or *Far*. The robot's observation at any time instance is a four tuple $\{d_N, d_S, d_E, d_W\}$ where each d_i is a binary discrete output as *Close* or *Far* measured independently in each direction. The probability of $p(Close|Distance)$ drops linearly from one to zero till a distance of R_{max} for each sensor. The inter-grid distances are measured from the centre of a grid cell. For instance, if $R_{max} = 5$ then the likelihood $p(Close|Distance)$ at discrete distances 1, 2, 3 and 4 decreases as 1, 0.75, 0.5, 0.25 and 0 (for a distance of 5 and beyond). The likelihood $p(Far|distance)$ are complementary and hence the corresponding likelihoods are 0, 0.25, 0.5, 0.75 and 1.0 (for distances ≥ 5).

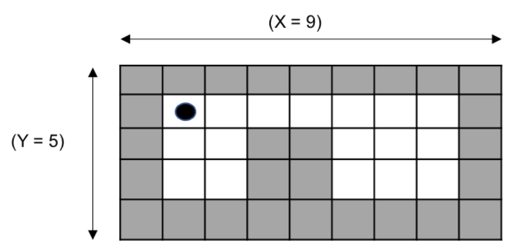

Figure 1: A robot (shown as a black dot) is moving in a grid world of size (X=9, Y=5) with four noisy sensors along the N, S, E and W directions. The white-colored and gray-colored cells are feasible and infeasible grid cells respectively. There is a surrounding wall in the grid.

   (a) Simulate the robot's motion in the grid ($X = 20$, $Y = 20$) by adding obstacles appropriately. The initial position can be randomly picked. Simulate and record the sequence of sensor observations for $T = 25$ time steps. Estimate (i) the robot's current position at time t and (ii) the robot's *most-likely* path, given the sequence of observations generated by the simulation above till time T.

   (b) Visualize and plot the log-likelihood for the robot's estimated position at time t as a spatial distribution over the grid. Observe how the belief updates with the arrival of new observations. Record the changing belief as a video.

   (c) Compare the trajectory obtained by estimating the robot's current position with the actual location using the Manhattan distance metric. Similarly, compare the robot's *most-likely* path given observations with the actual trajectory. Perform at least $N = 50$ runs and report the average and variance in your results.

   (d) Increase and decrease the parameter R_{max} as $\{1, 10\}$ and evaluate its effect on the current state estimate of the robot in relation to the ground truth.

   (e) Modify the grid by changing the layout of obstacles. Report a grid layout where localization is challenging (takes many observations to converge) and a grid layout where localization occurs very rapidly.