MeEn 595R – Autonomous Systems
Homework #4

You are to design a particle filter to estimate the location of an autonomous two-wheeled robot operating on a 20 m by 20 m field. This is commonly called Monte Carlo Localization (MCL) because the particle filter uses conditional probabilities specific to localization: the motion model and the measurement model. Located on this field are three landmarks that are continuously visible to the robot. The robot can measure the range and bearing to each of these landmarks as it moves about. The dynamics of the moving robot are described by a velocity motion model as discussed in Chapter 5.

The robot has initial conditions x_0 = -5 m, y_0 = -3 m, θ_0 = 90 deg. Its commanded velocities are given by

$$v^c = 1 + 0.5\cos(2\pi(0.2)t)$$
$$\omega^c = -0.2 + 2\cos(2\pi(0.6)t).$$

You should assume that the velocities experienced by the robot are noisy versions of the commanded velocities where the noise characteristics are modeled by $\alpha_1 = \alpha_4 = 0.1$ and $\alpha_2 = \alpha_3 = 0.01$.

The landmark locations are (6, 4), (-7, 8), and (6, -4). The standard deviation of the range and bearing sensor noise for each of the landmarks is given by $\sigma_r = 0.1$ m and $\sigma_\phi = 0.05$ rad respectively.

You are to implement your MCL algorithm with 1000 particles and a sample period of 0.1 s for a duration of 20 s.

Tasks:

1. Implement the full MCL algorithm found in Table 8.2 with the three landmarks. The particle filter algorithm of Table 4.3 may be useful in helping you understand how the MCL algorithm works. For resampling, use the low-variance resampling algorithm of Table 4.4. Create plots comparing your true states to the estimated states versus time. Plot estimation error and 95-percent uncertainty bounds (from your variances) versus time for each of your states.
2. Exercise your MCL by changing the input velocities, landmark location, sensor noise levels, and control noise levels. Does your MCL behave as expected?
3. Vary the number of points used by your MCL. How does this affect the performance (speed and accuracy) and numerical stability of your implementation?

Hints:

- Just one hint: Take it step by step and carefully check your code along the way. At each step, verify that your code is producing results that make sense.