MeEn 595R – Autonomous Systems
Homework #3

You are to design an unscented Kalman filter (UKF) to estimate the location of an autonomous two-wheeled robot operating on a 20 m by 20 m field. 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 UKF with a sample period of 0.1 s for a duration of 20 s.

Tasks:

1. Implement the full UKF algorithm found in Table 7.4 with a single landmark at (6, 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. Plot your Kalman gains versus time.
2. Exercise your UKF by changing the input velocities, landmark location, sensor noise levels, and control noise levels. Does your UKF behave as expected?
3. Vary the tuning parameters of the unscented transform. How does this affect the performance and numerical stability of your UKF implementation?
4. Modify your UKF algorithm to work with three landmarks by performing one measurement update each time through the UKF and cycling through the three landmarks. Does sensing three landmarks improve estimation performance over a single landmark?

Hints:

- The Matlab command chol produces an upper-triangular Cholesky factorization. We need the lower-triangular factorization, which can be obtained by taking the transpose of the Matlab output.
- 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.