CSCI–4190 Introduction to Robotic Algorithms, Spring 2002
Assignment 2: out February 15, due February 26 and March 5

Localization and the extended Kalman filter

In this assignment, you will write a program that simulates the motion of a mobile robot and keeps track of the uncertainty in its configuration. In the first part of the assignment, this uncertainty will increase with each movement due to the accumulation of dead reckoning errors. In the second part of the assignment, the robot will use a range sensor to measure the distance to the closest obstacle, so you will implement an extended Kalman filter to improve the configuration estimate and reduce its uncertainty.

Your program should do the following:

- Read the problem description from files. This will include a description of the world and a list of commands for the robot.
- Graphically display the world and show the robot at its starting position.
- Execute the robot commands.
  - After each motion command, the configuration of the robot and the uncertainty of its configuration should be drawn on the screen.
  - For each sensing command, the sensor reading should be drawn on the screen, and then the updated configuration uncertainty should be drawn.

  The program should also print a narrative of this process. More details of this step are contained in the following sections.

I will be providing support code for:

- Reading the problem description from files
- Graphical display
- Simulating the robot's motion and its range sensor
- Calculating derivatives related to the range sensor that you will need for the extended Kalman filter.

I am only officially supporting the CS department Sun (UNIX) environment. However, the support code will compile and run under Linux. See the assignment web page for more information.

In addition to the program, there will be some written questions. The written questions at the end of this handout are due on February 27. Your program and any additional questions are due on March 6. (Look for a forthcoming handout with these additional questions and more details on the support code.)

Support code

I expect to have the support code out before Monday February 18, but definitely by Tuesday the 19th. Check the Assignment 2 web page for more details.

Getting started

As before, login to a CS machine, get an `xterm`, and type the following commands at the prompt (`$`). (I assume you already have created a directory called `ira` in your home directory.)

```
$ cp -r /projects/ira/assign2 ira        copy the support code
$ cd ira/assign2
$ gmake                                  compile everything
$ ./assign2 Problem1.txt                 run the program
```

This program will first display the world draw the robot at its starting location. Pressing the "n" key will make the robot execute the next motion command (if there are any remaining). Press the "q" key to quit.

Localization

In this assignment, the motion commands to the robot consist of a distance and an angle. The robot first moves forward the given distance (in the direction of its current heading) and then turns the given angle. However, the robot does not move the exact distance or turn the exact angle that was commanded because of imprecision in the robot's control system. The odometry will measure the translation and turn, but there is some error in this measurement. This will be the sole input to the extended Kalman filter you will implement in this assignment. The simulated robot will be equipped with either a SONAR sensor or a laser rangefinder. (The difference is only whether the beam is a wide cone which detects the range to the closest obstacle in the cone or whether the beam is a straight line.) We will be able to specify the direction that this sensor points.

The state of the robot will be the vector $x = [\; x \quad y \quad \theta \;]^T$ and the odometry measurements will be the input vector $u = [\; d \quad a \;]^T$. where d is the distance and a the angle. Generally, we will be working with estimates of the state at a given time step; we would write $\hat{x}(k)$ to represent the estimate of the state at step k. (The "hat" indicates that the variable represents an estimate.) We will write $u(k)$ for the odometry measurement of the movement made from the state $x(k)$.

We use the function Φ to calculate the state at the "next step" from the current state and the current input. We can write this as:

$$\hat{x}(k|k-1) = \Phi[\hat{x}(k-1), u(k-1)] \tag{1}$$

The "bar" notation means that these quantities are calculated at step k using information through step $k-1$. When there is no measurement at step k we will set $\hat{x}(k) = \hat{x}(k|k-1)$.

For our problem this will be a nonlinear function, so we will need to apply the extended Kalman filter. The linearization (i.e. first order approximation) we need is:

$$\Phi[x, u] \approx \Phi[\hat{x}(k-1), u(k-1)] + J_x(x - \hat{x}(k-1)) + J_u(u - u(k-1)) \tag{2}$$

This is a linearization about the "point" $(\hat{x}(k-1), u(k-1)$ where J_x and J_u are the Jacobians of Φ:

$$J_x = \begin{bmatrix} \frac{\partial \Phi_x}{\partial x} & \frac{\partial \Phi_x}{\partial y} & \frac{\partial \Phi_x}{\partial \theta} \\ \frac{\partial \Phi_y}{\partial x} & \frac{\partial \Phi_y}{\partial y} & \frac{\partial \Phi_y}{\partial \theta} \\ \frac{\partial \Phi_\theta}{\partial x} & \frac{\partial \Phi_\theta}{\partial y} & \frac{\partial \Phi_\theta}{\partial \theta} \end{bmatrix} \qquad J_u = \begin{bmatrix} \frac{\partial \Phi_x}{\partial d} & \frac{\partial \Phi_x}{\partial u} \\ \frac{\partial \Phi_y}{\partial d} & \frac{\partial \Phi_y}{\partial u} \\ \frac{\partial \Phi_\theta}{\partial d} & \frac{\partial \Phi_\theta}{\partial u} \end{bmatrix} \tag{3}$$

To characterize the uncertainty, we use the state covariance matrix:

$$P_x = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_\theta^2 \end{bmatrix} \tag{4}$$

At the beginning, when we know the robot's configuration exactly, the elements of the state covariance matrix are all zero. After each motion command, the covariance matrix is updated according to the equation:

$$P_x(k|k-1) = J_x P_x(k-1) J_x^T + J_u P_u(k-1) J_u^T \tag{5}$$

The matrix P_u is the covariance matrix of the input vector u. We will assume that the translational and rotational errors in the robot's motion control and dead reckoning are uncorrelated. The variance of the translational and rotational errors is contained in the problem specification file and is available through the `LocProblem` class. As with the state estimate, when there is no measurement to be incorporated, then we will set $P_x(k) = P_x(k|k-1)$.

When we do have a measurement at step k, this information will be incorporated before setting the final values of $\hat{x}(k)$ and $P_x(k)$. We assume that this measurement process can be modeled by the equation:

$$z = H[x] + v(k) \tag{6}$$

The H function computes the sensor value from the state, and v is a random variable representing the noise in the measurement. In our problem, the measurement z will be a scalar, the measured range from the sensor.

We compute the error between the actual measurement $z(k)$ and the predicted measurement (without noise) based on our best estimate of the state:

$$r(k) = z(k) - H[\hat{x}(k|k-1)] \tag{7}$$

Now we can compute the final state estimate at step k

$$\hat{x}(k) = \hat{x}(k|k-1) + K(k)r(k) \tag{8}$$

where $K(k)$ is the Kalman gain:

$$K(k) = P(k|k-1) J_H^T (J_H P(k|k-1) J_H^T + C_v)^{-1} \tag{9}$$

Here, J_H is the Jacobian of the H function and C_v is the covariance matrix for the measurement noise. The new state covariance matrix is:

$$P(k) = (I - K(k)J_H)P(k|k-1) \tag{10}$$

Written questions

1. Write out the specific formulas you will use in this assignment, in particular:
   - $\Phi[x,u]$ from Equation 1
   - the Jacobians J_x and J_u from Equation 3
   - the covariance matrix P_u (state this matrix in terms of the values σ_d and σ_a which be given in the Problem specification file)
2. Copy the equation for the Kalman gain (Equation 9) and indicate the size of each matrix in this formula. Also indicate the size of the matrix resulting from the inverse operation.
3. Write out the form of the Jacobian J_H. (See Equation 3 for an example). Explain what each term is. (Relate it to the sensor or measurement process, not just "this is the derivative of z with respect to x.") Explain how you would compute these derivatives.