CSE 483: Mobile Robotics - Assignment 01

Due: 17th August 2016, 2300 hrs

August 16, 2016

General Instructions

1. The assignments are to be done in Matlab. The code provided along with the assignments may run in Octave as well, but we provide no guarantees on that front.
2. Plagiarism is strictly prohibited. The popular *MOSS* tool will be employed for code plagiarism detection.
3. The functions that you write for this assigments will be called directly in future assignments. So, keep the functions modular, the code clean, and commented.
4. **In all your scripts and functions, add the following random seed '201601', for repeatability**. If you do not know what a random seed is, look it up. In Matlab, just adding the line *rng(201601)* at the top of the script, or at the beginning of a function definition would suffice. Ensure that this is called only once per script/function.

1 Warm-up tasks

The following set of tasks aims to familiarize you with manipulating multivariate gaussians in Matlab.

1. Sample 1000 values from a multivariate gaussian distribution with mean $[xyz, xyz]^T$, where xyz are the last three digits of your (IIIT) ID number[1], and covariance $diag([2, 4])$. Use the Matlab function `mvnrnd` for the same.
2. For the samples generated in the above question, compute the mean and the covariance. Do NOT use the Matlab functions `mean` and `cov` for this task.

[1]Before selecting xyz to be the last three digits of your ID number, kindly check that the matrix $diag([x, y])$ is non-singular and has a positive determinant. **If this is not the case, choose xyz as 483 (the course code!).**

3. Draw a (2D) scatter plot that shows the points sampled. Compute the value of the probability density function (PDF) at each of the points. Color each point according to its PDF value (use any color scheme you like, but make sure there is a color index on the right side of the plot). For reference, you could look at the lecture slides on multivariate gaussians.

4. Transform the samples generated in the first part of this task by applying a linear transform $Ax + b$, where $A = diag([x, y])$, and $b = [z, z]^T$ (x, y, z are the last three digits of your ID number respectively).

5. For the transformed samples, compute the mean and the covariance.Do NOT use the Matlab functions `mean` and `cov` for this task.

6. Draw a (2D) scatter plot that shows the **transformed points**. Compute the value of the probability density function (PDF) at each of the points. Color each point according to its PDF value (use any color scheme you like, but make sure there is a color index on the right side of the plot). For reference, you could look at the lecture slides on multivariate gaussians.

2 Sampling sanely from mvnrnd

By now, you would have (hopefully) used `mvnrnd` multiple times. This task is to make sure that the values you obtain from `mvnrnd` are *sane*, i.e., they are confined to a particular confidence ellipse of the gaussian.

Very frequently, `mvnrnd` ends up producing values well outside the 75% confidence ellipse of the chosen multivariate gaussian. In this assignment, we will write our own wrapper for `mvnrnd`. Let's call it `sane_mvnrnd`.

1. The function file `sane_mvnrnd.m` has been provided along with the assignment. It currently has stubs that you need to fill out to get the function working *correctly*. The function file has relevant documentation, listed out in the form of comments. Look out for all `TODO` comments. They indicate the positions in the file where you would add code. Edit the file and ensure it works, i.e, it generates *sane* samples.

3 Plotting the error ellipse

In this task, we aim to visualize the error ellipse for a two-dimensional gaussian. The task would comprise of the following steps.

1. Take the 1000 samples you obtained in **question 1** of the warm-up tasks (part 1). Use the mean and covariance computed in **question 2** of the warp-up tasks.

2. Compute the eigenvalues and eigenvectors for the covariance matrix. Determine which of them is the larger eigenvalue.

3. Follow the other steps listed out in the comments of `get_error_ellipse.m` and fill in the stubs. For reference, visit the following webpage `http://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/`

4. Once you complete the function correctly, run the function to get the values `ellipse_x` and `ellipse_y`. Then, run the following command to plot the ellipse.
`plot(ellipse_x, ellipse_y)`

4 Deliverables

1. A zipped (`.zip`) folder (try not to use any other format such as `.bz2`, `.tar.gz`, etc.) whose name is only your ID number, eg. `201507666.zip`. The folder should have a `report.pdf` file which contains the answers for the warm-up tasks, i.e, section 1 of the assignment, and the error ellipse plots for section 3 of the assignment.

2. Additionally the folder `01_Sampling`, provided along with the assignment, must be present. The `.m` files inside this folder must contain the modified code.

3. Please keep the input/output signatures for each function the same. Also do not change the directory structure, and do not add additional files. The code will be evaluated on a set of test cases, and common input/output signatures are required for automated testing.