Sheet 5

Topic: Sampling, Motion Models

Exercise 1: Sampling

Implement a function in *Python* which generate samples of a normal distribution $N(\mu, \sigma^2)$. The input parameters of these functions should be the mean μ and the standard deviation σ of the normal distribution. As only source of randomness, use samples of a uniform distribution.

a) Generate the normal distributed samples by summing up 12 uniform distributed samples, as explained in the lecture.
b) Using Python's built-in function timeit.default_timer(), compare the execution times of your own function to the built-in function numpy.random.normal.

Exercise 2: Odometry-based Motion Model

A working motion model is a requirement for all Bayes Filter implementations. In the following, you will implement the simple odometry-based motion model.

a) Implement the odometry-based motion model in Python. Your function should take the following three arguments

$$x_t = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \qquad u_t = \begin{pmatrix} \delta_{r1} \\ \delta_{r2} \\ \delta_t \end{pmatrix} \qquad \alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}$$

where x_t is the pose of the robot before moving, u_t is the odometry reading obtained from the robot, and α are the noise parameters of the motion model. The return value of the function should be the new pose x_{t+1} of the robot predicted by the model.
As we do not expect the odometry measurements to be perfect, you will have to take the measurement error into account when implementing your function. Use the sampling methods you implemented in Exercise 1 to draw normally distributed random numbers for the noise in the motion model (or use numpy.random.normal).

b) If you evaluate your motion model over and over again with the same starting position, odometry reading, and noise values what is the result you would expect?
c) Evaluate your motion model 5000 times for the following values

$$x_t = \begin{pmatrix} 2.0 \\ 4.0 \\ 0.0 \end{pmatrix} \qquad u_t = \begin{pmatrix} \frac{\pi}{2} \\ 0.0 \\ 1.0 \end{pmatrix} \qquad \alpha = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.01 \\ 0.01 \end{pmatrix}$$

Plot the resulting (x, y) positions for each of the 5000 evaluations in a single plot.