Perception in Robotics
Term 3, 2022. PS1

Gonzalo Ferrer
Skoltech

1 February 2022

This problem set has four tasks, comprising 15% of your course grade, which is individual work. You are encouraged to talk at the conceptual level with other students, discuss on the equations and even on results, but you may not show/share/copy any non-trivial code.

Submission Instructions

Your assignment must be submitted by 11:59pm on **February 9, Wednesday**. You are to upload your assignment directly to Canvas as **two** attachments:

1. A `.tar.gz` or `.zip` file *containing a directory* named after your uniquename with the structure shown below. Alternatively, submit a single `.ipybn` file named after your uniquename that includes solutions for all tasks.

   ```
   alincoln_ps1.zip:
   alincoln_ps1/
   alincoln_ps1/task1.py
   alincoln_ps1/task2.py
   alincoln_ps1/plot2dcov.py
   alincoln_ps1/task3.py
   ```

2. A PDF with the written portion of your write-up. Scanned versions of hand-written documents, converted to PDFs, are perfectly acceptable (reduced size). No other formats (e.g., .doc) are acceptable. Your PDF file should adhere to the following naming convention: `alincoln_ps1.pdf`.

Homework received after 11:59pm is considered late and will be penalized as per the course policy. The ultimate timestamp authority is the one assigned to your upload by Canvas. No exceptions to this policy will be made.

Important: For all (x, y) plots you will want to use the `Axes.set_aspect('equal')` command in the `matplotlib` library to set the aspect ratio so that equal tick mark increments on the x-,y- and z-axis are equal in size. This makes spheres look like spheres, instead of an ellipsoid.

Task 1: Probability (25 points)

For this task, write a code in Python that generates all the different figures, as requested for each problem.

A. (5 pts) Plot the *probability density function* $p(x)$ of a one dimensional Gaussian distribution $\mathcal{N}(x; 1, 1)$.

   Hint: you might want to look at the library `scipy.stats` and use the function `norm.pdf()`.

B. (5 pts) Calculate the probability mass that the random variable X is less than 0, that is, $Pr\{X \leq 0\} = \int_{-\infty}^{0} p(x)dx$.

   Hint: you might want to use the function `norm.cdf()`.

C. (15 pts) Consider the new observation variable z, it gives information about the variable x by the *likelihood function* $p(z|x) = \mathcal{N}(z; x, \sigma^2)$, with variance $\sigma^2 = 0.2$. Apply the Bayes' theorem to derive the posterior distribution, $p(x|z)$, given an observation $z = 0.75$ and plot it. For a better comparison, plot the prior distribution, $p(x)$, too.

   Hint: There are different ways in which the normalization factor for the posterior can be calculated (e.g. you can find the normalization factor numerically). Extra points will be given if two or more ways of calculation are implemented in the solution.

Task 2: Multivariate Gaussian (25 points)

A. (10 pts) Write the function `plot2dcov` which plots the 2d contour given three core parameters: mean, covariance, and the iso-contour value k. You may add any other parameter such as color, number of points, etc.

   Hint: Make use of the Cholesky decomposition `scipy.linalg.cholesky` or SVD and project a circumference with radius k, as explained in class. Use, for instance, 30 points.

   Hint: the Cholesky decomposition routine, could solve for the upper triangular matrix, $A^\top \cdot A = \Sigma$ which is not what you may want. Read the function help and make sure the decomposition maps back to the original covariance in the form $A \cdot A^\top = \Sigma$.

   Then, use `plot2dcov` to draw the iso-contours corresponding to 1,2,3-sigma of the following Gaussian distributions: $\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 2\end{bmatrix}\right)$, $\mathcal{N}\left(\begin{bmatrix}5\\0\end{bmatrix}, \begin{bmatrix}3 & -0.4\\-0.4 & 2\end{bmatrix}\right)$ and $\mathcal{N}\left(\begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}9.1 & 6\\6 & 4\end{bmatrix}\right)$. Use the `set_aspect('equal')` command and comment on them.

B. (5 pts) Write the equation of sample mean and sample covariance of a set of points $\{x_i\}$, in vector form as was shown during the lecture. You can provide your solution by using Markdown, latex, by hand, etc.

C. (10 pts) Draw random samples from a multivariate normal distribution. You can use the python function that draws samples from the univariate normal distribution $\mathcal{N}(0, 1)$. In particular, draw and plot 200 samples from $\mathcal{N}\left(\begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}1 & 1.3\\1.3 & 3\end{bmatrix}\right)$; also plot their corresponding 1-sigma iso-contour. Then calculate the sample mean and covariance in vector form and plot again the 1-sigma iso-contour for the estimated Gaussian parameters. Run the experiment multiple times and try different number of samples. Comment on the results.

Task 3: Covariance Propagation (25 points)

For this task, we will model an omni-directional robotic platform, i.e., a holonomic platform moving as a free point without restrictions.

The propagation model is the following: $\begin{bmatrix} x \\ y \end{bmatrix}_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}_t + \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix}_t$, where the controls $u = [v_x, v_y]^\top$ are the velocities which are commanded to the robot. Unfortunately, there exists some uncertainty on command execution $\begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix}_t \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \right)$. We will consider a time step of $\Delta t = 0.5$.

A. (5 pts) Write the equations corresponding to the mean and covariance after a single propagation of the holonomic platform. How can we use this result iteratively?

B. (5 pts) Draw the propagation state PDF (1-sigma iso-contour) for times indexes $t = 0, \ldots, 5$ and the control sequence $u_t = [3, 0]^\top$ for all times t. The PDF for the initial state is $\begin{bmatrix} x \\ y \end{bmatrix}_0 \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \right)$.

C. (5 pts) Somehow, the platform is malfunctioning; thus, it is moving strangely and its propagation model has changed: $\begin{bmatrix} x \\ y \end{bmatrix}_t = \begin{bmatrix} 1 & 0.3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{t-1} + \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}_t + \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix}_t$. All the other parameters and controls are the same as defined earlier.

   Draw the propagation state PDF (1-sigma iso-contour and 500 particles) for times indexes $t = 0, \ldots, 5$,

D. (5 pts) Now, suppose that the robotic platform is non-holonomic, and the corresponding propagation model is: $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t-1} + \begin{bmatrix} cos(\theta)\Delta t & 0 \\ sin(\theta)\Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}_t + \begin{bmatrix} \eta_x \\ \eta_y \\ \eta_\theta \end{bmatrix}_t$, being $\begin{bmatrix} \eta_x \\ \eta_y \\ \eta_\theta \end{bmatrix}_t \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \right)$ and the PDF for the initial state $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_0 \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \right)$.

   Propagate, as explained in class (linearize plus covariance propagation), for five time intervals, using the control $u_t = [3, 1.5]^\top$ showing the propagated Gaussian by plotting the 1-sigma iso-contour. Angles are in radians. *Hint:* you can marginalize out θ and plot the corresponding $\Sigma_{(xy)}$ as explained in class.

E. (5 pts) Repeat the same experiment as above, using the same control input u_t and initial state estimate, now considering that noise is expressed in the action space instead of state space: $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t-1} + \begin{bmatrix} cos(\theta)\Delta t & 0 \\ sin(\theta)\Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v + \eta_v \\ w + \eta_w \end{bmatrix}_t$, being $\begin{bmatrix} \eta_v \\ \eta_w \end{bmatrix}_t \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 0.1 \end{bmatrix} \right)$. Comment on the results.