

Autonomous Navigation and Perception (086762)

Homework #1

Submission guidelines:

- Submission in pairs by: [05 April 2022, 14:30](#).
- Email submission as a *single zip or pdf file* to anpsubmission@gmail.com.
- Make sure to name the submitted file ID1-ID2.zip (or .pdf) with ID1, ID2 - the students' ids.
- For hands-on parts, your source code should be submitted as well (include it in the above zip file).

Theoretical questions:

1. (Bayesian inference) Consider probabilistic motion and observation models

$$\mathbb{P}(x_{k+1} \mid x_k, a_k) \quad , \quad \mathbb{P}(z_k \mid x_k).$$

Assume at each time instant i a robot/agent gets an observation z_i and performs action a_i , that transitions its state from x_i to x_{i+1} . Assume a prior $\mathbb{P}(x_0)$ is available.

Provide a full and detailed derivation of the following (indicate clearly all assumptions in derivation):

- (a) Recursive formulation: Express the posterior distribution over state x_k at time instant k in terms of the posterior distribution over x_{k-1} from the previous time instant $k-1$.
 - (b) Smoothing formulation: Express the posterior distribution over past and current states, $x_{0:k}$ at time instant k , in terms of the corresponding posterior distribution from the previous time instant $k-1$.
 - (c) Smoothing formulation: Express the posterior distribution over past and current states, $x_{0:k}$ at time instant k , in terms of all actions and observations obtained thus far.
2. (Occupancy grid mapping) Consider a robot maps an unknown environment and assume localization is known (i.e. robot pose x_i at each time instant i is known). Denote by M an occupancy grid map that partitions an environment into a set of cells (or voxels): $M = \{m_i\}$. Develop equations of a binary Bayes filter considering a binary state $m \in \{\text{occ}, \text{free}\}$ (e.g. one of the cells m_i):
 - (a) Express the distribution over the hypotheses $m = \text{free}$ and $m = \text{occ}$ at time instant k , given observations $z_{1:k}$ captured from known locations/poses $x_{1:k}$, and considering an inverse observation model $\mathbb{P}(m \mid z, x)$.
 - (b) Define the log-odds ratio as $l(m) = \log \frac{\mathbb{P}(m=\text{free})}{1-\mathbb{P}(m=\text{free})}$. Using the result from clause [2a](#), express the log-odds ratio $l_k(m)$ at time instant k in terms of $l_{k-1}(m)$, z_k , x_k and prior $\mathbb{P}(m)$.
 - (c) Explain how to get a posterior distribution over M at time instant k based on clause [2a](#) and state any assumptions you take.

Hands-on tasks:

1. Read Appendix G of the book <https://algorithmsbook.com/#> to make yourself familiar with Julia language. The official Julia Debugger is based on Visual Studio Code <https://code.visualstudio.com/>. To set up the debugging environment, you will need to install VSCode, and an extension called Julia. The core component to be installed is Julia language itself. Download the latest release from here <https://julialang.org/downloads/> (currently 1.7.2) and unpack somewhere. In VSCode open the Julia extension settings and set the Executable Path (<full_path_to_julia_executable/julia>).

If everything was set up correctly by pressing `Alt+j`, `Alt+o`, you would be able to start Julia REPL tied to the VSCode debugger. You also can use the button “run” to start the REPL and run the current opened file.

Julia is bleeding edge just in time compiled high performance oriented language. Therefore, it does various caching and precompilation operations. Thus, some code changes require restarting the REPL. Luckily to us, there is an essential package called “Revise” to avoid this.

VSCoDe debugger can run the file in interpreted and compiled mode. Without a debugger, Julia always runs in compiled mode. This mode is faster, but the debugging capabilities in this mode are not mature yet. In the VSCoDe debugger in the window of breakpoints, there is a compiled mode option. Run the file “main.jl” in two modes and observe speedup in compiled mode. Note that there is an included file “utils.jl”. We will use functions from this file in the future.