

Project 3: SLAM

Collaboration in the sense of discussion is allowed, however, the work you turn in should be your own - you should not split parts of the assignments with other students and you should certainly not copy other students' code or papers. See the collaboration and academic integrity statement here: <https://natanaso.github.io/ece276a>. Books may be consulted but not copied from.

Submission

You should submit the following two files by the deadline shown on the top right corner.

1. **FirstnameLastname_P3.pdf** on **Gradescope**: upload your solutions to the theoretical problems (Problems 1-2). You may use latex, scanned handwritten notes (write legibly!), or any other method to prepare a pdf file. Do not just write the final result. Present your work in detail explaining your approach at every step. Also, attach to the **same** pdf the report for Problem 4. You are encouraged but not required to use an IEEE conference template¹ for your report.
2. **FirstnameLastname_P3.zip** on **TritonEd**: upload all code you have written for Problem 3 (do not include the training and test datasets) and a README file with clear instructions for running it.

Problems

In square brackets are the points assigned to each problem.

1. [2 pts] Consider the following motion and observation models:

$$\begin{aligned} x_{t+1} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_t + w_t, & w_t &\sim \mathcal{N}(0, \text{diag}(1/10^2, 1)) \\ z_t &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_t + v_t, & v_t &\sim \mathcal{N}(0, 10^2) \end{aligned}$$

- (a) Write down the Kalman filter equations for this model
- (b) Now, consider using a particle filter for the model above. When deriving the particle filter, we approximated the mixture probability density function (pdf) resulting from the prediction step with particles:

$$\sum_{k=1}^{N_{t|t}} \alpha_{t|t}^{(k)} p_a(x | \mu_{t|t}^{(k)}, u_t) \approx \sum_{k=1}^{N_{t+1|t}} \alpha_{t+1|t}^{(k)} \delta(x; \mu_{t+1|t}^{(k)})$$

We can obtain a more accurate filter if we use the actual predicted pdf in the update step instead of the particle approximation above. More precisely, the update step without approximation becomes:

$$p_{t+1|t+1}(x) = \frac{p_h(z_{t+1} | x) \sum_{k=1}^{N_{t|t}} \alpha_{t|t}^{(k)} p_a(x | \mu_{t|t}^{(k)}, u_t)}{\int p_h(z_{t+1} | s) \sum_{j=1}^{N_{t|t}} \alpha_{t|t}^{(j)} p_a(s | \mu_{t|t}^{(j)}, u_t) ds} = \sum_{k=1}^{N_{t+1|t}} \left[\frac{\alpha_{t+1|t}^{(k)} p_h(z_{t+1} | x) p_a(x | \mu_{t|t}^{(k)}, u_t)}{\sum_{j=1}^{N_{t+1|t}} \alpha_{t+1|t}^{(j)} \int p_h(z_{t+1} | s) p_a(s | \mu_{t|t}^{(j)}, u_t) ds} \right]$$

We can see that the above pdf is a (normalized) mixture of pdfs of the form:

$$\pi^{(k)}(x) := \frac{p_h(z_{t+1} | x) p_a(x | \mu_{t|t}^{(k)}, u_t)}{\int p_h(z_{t+1} | s) p_a(s | \mu_{t|t}^{(k)}, u_t) ds}$$

We can use the exact mixture to directly approximate the update step with particles (instead of the prediction step as in class). In other words, we need to sample from the *optimal proposal distributions* $\pi^{(k)}(x)$ by repeating the following steps $N_{t+1|t+1}$ times and normalizing the weights at the end:

- Draw $k \in \{1, \dots, N_{t+1|t}\}$ with probability $\alpha_{t|t}^{(k)}$

¹https://www.ieee.org/conferences_events/conferences/publishing/templates.html

- Draw $\mu_{t+1|t+1}^{(k)} \sim \pi^{(k)}(\cdot)$
- Add the weighted sample $\left(\mu_{t+1|t+1}^{(k)}, p_{t+1|t+1} \left(\mu_{t+1|t+1}^{(k)}\right)\right)$ to the new particle set

Derive the optimal proposal distribution $\pi^{(k)}(x)$ for the model in part a). You do not need to carry out all the other steps above.

2. [6 pts] You are given the following schematic diagram of a tricycle (Fig. 1). The kinematic quantities available for this vehicle are: (i) the linear velocity v of the front wheel, and (ii) the steering angle θ of the front wheel. Assume that a frame of reference $\{R\}$ is attached to the vehicle with origin in the middle of the back wheels and x axis as shown in Fig. 1. Also note that the length of the vehicle (distance from the front wheel to the origin of frame $\{R\}$) is known to be L .

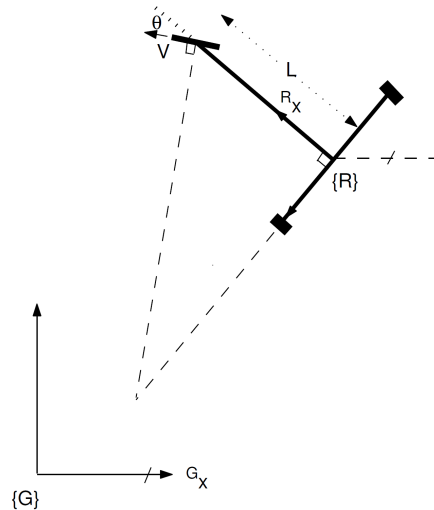


Figure 1: Tricycle diagram. The 3 dashed lines shown are (from right to left): parallel to the global frame's x axis, perpendicular to the x axis of the vehicle, and perpendicular to the front wheel. The dotted line with arrows shows the length of the vehicle.

- Determine the continuous-time system equations that describe the time evolution of the 2-D position and orientation (pose in $SE(2)$) of the robot with respect to the global frame of reference $\{G\}$. Assume that v and θ are noise-free.
- Discretize the continuous-time equations that you found in the previous part assuming that v and θ remain constant over a small time step of duration τ . Assume that v and θ are noise-free.
- Based on the motion model you derived in the previous part, write the Extended Kalman Filter equations that you would use for the mean and covariance propagation. Assume that the control inputs are now perturbed by additive Gaussian noise:

$$v + w_v \quad \theta + w_\theta,$$

where $w_v \sim \mathcal{N}(0, \sigma_v^2)$ and $w_\theta \sim \mathcal{N}(0, \sigma_\theta^2)$.

- The same vehicle (Fig. 1) has a distance measuring device placed right on top of the front wheel to measure its distance from a beacon whose 2-D position $y \in \mathbb{R}^2$ in the global frame $\{G\}$ is known.
 - Write the nonlinear equation for this distance measurement model.
 - Derive the measurement Jacobian.

Assume that the system state vector contains the position and orientation of the vehicle as described by frame $\{R\}$.

3. [14 pts] Implement simultaneous localization and mapping (SLAM) using odometry, inertial, 2-D laser range, and RGBD measurements from a humanoid robot. Use the IMU, odometry, and laser measurements to localize the robot and build a 2-D occupancy grid map of the environment. Use the RGBD information to color the floor of your 2-D map.
- **Training data:** now available at:
<https://drive.google.com/open?id=0B241vEW29598Zm5LT241b2xLdWs>
 - **Test data:** released on 12/06/17 at:
<https://drive.google.com/open?id=0B241vEW29598UTJTM2hnMnNfZGs>
 - Additional information about the data is provided in ref/THOR_Configuration.pdf
 - **Mapping:** Try mapping from the first scan and plot the map to make sure your transforms are correct before you start estimating the robot pose.
 - **Prediction:** Implement a prediction-only particle filter at first. In other words, use the walking odometry and the yaw gyro measurements to estimate the robot trajectory and build a 2-D map based on this estimate before correcting it with the laser readings.
 - **Update:** Once the prediction-only filter works, include an update step that uses scan matching to correct the robot pose. Remove scan points that are too close, too far, or hit the ground.
 - **Texture map:** Project colored points from the RGBD sensor onto your occupancy grid in order to color the floor. Find the ground plane in the transformed data via RANSAC or simple thresholding on the height.
4. [6 pts] Write a project report describing your approach to the SLAM and texture mapping problems. Your report should include the following sections:
- **Introduction:** discuss why the problem is important and present a brief overview of your approach
 - **Problem Formulation:** state the problem you are trying to solve in mathematical terms. This section should be short and clear and should define the quantities you are interested in.
 - **Technical Approach:** describe your approach to SLAM and texture mapping
 - **Results:** present your training results, test results, and discuss them – what worked, what did not, and why. Make sure your results include (a) images of your SLAM system over time, (b) textured maps of the training and test datasets. If you have videos do include them in the zip file and refer to them in your report!