NA 568 Mobile Robotics: Methods & Algorithms Winter 2022 – Homework 2 – Estimation & Kalman Filtering

Maani Ghaffari
University of Michigan

January 14, 2022

This is a reminder that no late HW is accepted. We drop your lowest grade from HW 1-6. It is perfectly fine to drop a zero as a HW grade. We are using Gradescope for turning in HW; see relevant information on the course Canvas site.

This problem set counts for about 7% of your course grade. You are encouraged to talk at the conceptual level with other students, but you must complete all work individually and may not share any non-trivial code or solution steps. See the syllabus for the full collaboration policy.

Submission Instructions

Your assignment must be received by 11:55 pm on Friday, January 28 (Anywhere on Earth Time). This corresponds to 6:55 AM on January 29 in Eastern Time. This is selected out of fairness to all our students, including those who take the course remotely. You are to upload your assignment directly to the Gradescope website as two attachments:

1. A `.tar.gz` or `.zip` file *containing a directory* named after your uniqname with the structure shown below.

   ```
   alincoln_hw2.tgz:
   alincoln_hw2/
   alincoln_hw2/task4.m
   alincoln_hw2/task5c.m
   alincoln_hw2/task5d.m
   ```

   Or a Jupyter notebook per task using Python or Julia kernels for your programming. Follow the same naming convention for the notebooks .

2. A PDF with the written portion of your write-up. Scanned versions of hand-written documents, converted to PDFs, are perfectly acceptable. No other formats (e.g., .doc) are acceptable. Your PDF file should adhere to the following naming convention: `alincoln_hw2.pdf`.

Important: For all (x, y) plots you will want to use the `axis equal` (matlab) or `plt.axis('equal')` (python) command to set the aspect ratio so that equal tick mark increments on the x-,y- and z-axis are equal in size.

1 Kalman Filter Background (10 points)

A. (5 pts) What are the assumptions of the Kalman Filter? List and explain all the assumptions thoroughly.

B. (5 pts) Describe the Kalman filter estimation cycle, and intuitively explain the meaning of each step in connection with Bayes' filter.

2 Target Tracking (15 points)

A target is moving in a 1D plane. The ownship position is known and fixed at the origin, thus the target's position at time step k is x_k. We have no information on the target's trajectory and thus we model the motion as a random walk with variance of 4 m^2. Based on previous measurements, we have an initial guess of the target position as $\mu_0 = 20$ m with variance $\sigma_0^2 = 9$ m^2. We then receive measurement from each of the next two time steps $z_1 = 22$ m and $z_2 = 23$ m. Each with variance 1 m^2.

Use a Kalman filter to estimate the state of the target at time step 2 (μ_2 and σ_2^2).

3 Uncertainty Propagation (10 points)

A. (5 pts) Explain how a probability distribution can be propagated through a nonlinear function. Elaborate on the methods and reason why they work.

B. (5 pts) A projection model from the 3D Cartesian space, $\mathbb{R}^3$, to a 2D image is given by

$$p = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x \frac{x}{z} + c_x \\ f_y \frac{y}{z} + c_y \end{bmatrix} \in \mathbb{R}^2, \quad \text{and} \quad l = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3$$

where f_x and f_y are focal lengths in x and y directions, respectively, and c_x and c_y are the coordinates of optical center of the image in pixels. Suppose a 3D landmark $l \in \mathbb{R}^3$ is observed via a noisy sensor, i.e., $l \sim \mathcal{N}(\mu_l, \Sigma_l)$. Derive an analytical model for the mean and covariance of p. Show work and provide necessary equations.

4 First-Order Covariance Propagation (25 points)

In most SLAM approaches, the noise characteristics of individual observations are linearized, resulting in a Gaussian covariance matrix. In this task, we explore the effects of the error introduced via linearization.

Consider a robot sensor that observes range and bearing to nearby landmarks. In this case, the range error is relatively small, but the bearing error is large. We are interested in determining the (x, y) position of the beacon based on observations obtained by the robot at the origin $(0, 0)$. For this problem, the robot does not move.

Suppose you obtain a (range, bearing) observation with mean $(10.0\,\text{m}, 0\,\text{rad})$ whose range standard deviation is $0.5\,\text{m}$, and whose bearing standard deviation is $0.25\,\text{rad}$. The range and bearing measurements are Gaussian and independent. Note: bearing is the angle from the x-axis to the beacon counterclockwise.

You should use the code template we provided and implement your code for each problems in the TODO block. Remember to include the results/figures in your PDF write-up.

A. (4 pts) Generate a point cloud representing 10,000 samples from the distribution over the position of the beacon as measured in

   i) the sensor frame, i.e. (r, θ) space and

   ii) the Cartesian (x, y) coordinate frame.

   In other words, generate observations of (range, bearing) and project these points into (x, y). (Hint: use the `randn()` function in Matlab or `numpy.random.randn()` function in python, and recall that you can sample from a univariate Gaussian with mean `mu` and standard deviation `sigma` with `mu + sigma*randn`.)

B. (4 pts) What is the (linearized) covariance of the beacon position in (x, y) coordinates? In other words, write the covariance of an observation in (x, y) coordinates in terms of the covariance of the observation in (range, bearing) coordinates. The transformation is non-linear, so you will need to compute a first-order approximation (Taylor expansion) of the transformation function. Make the appropriate Jacobians easy to read in your source code, using comments if necessary.

C. (4 pts) Draw in red the 1-sigma, 2-sigma, and 3-sigma contours of the analytical (linearized) covariance ellipses, super-imposed over the point clouds generated in parts A.i and A.ii. Now overlay in blue the actual covariance ellipses computed using sample-based expressions for the first and second moments. Do they agree? Why or why not?

   You may use the function `draw_ellipse()` provided in the `hw2-code.zip` or the helper function in the Jupyter Notebook. You should include two plots each with 3 red ellipses and 3 blue ellipses in the PDF.

D. (4 pts) From a purely theoretical perspective, assuming that the underlying process is truly Gaussian, we expect 39.35% of all samples to lie within the 1-sigma contour, 86.47% of samples to lie within the 2-sigma contour, and 98.89% to lie within the 3-sigma contour. (These frequencies were computed using the cumulative chi-square distribution for two degrees-of-freedom, i.e. χ^2_2. In Matlab, you can do this using the function `chi2cdf()` evaluated at `chi2cdf(1,2)`, `chi2cdf(4,2)`, and `chi2cdf(9,2)`, respectively.)

   Modify your software to count the samples falling within each (analytical) ellipse for parts A.i and A.ii. The error of a particular sample x, measured in "units" of sigma, is known as the Mahalanobis distance, and can be computed as $\sqrt{(x-\mu)^\top \Sigma^{-1}(x-\mu)}$.

E. (4 pts) If the point samples were truly distributed as a Gaussian (clearly the (x, y) are not), your counts would match the theoretically predicted values. Try varying the noise parameters: under what conditions do the counts come close to matching the theoretically predicted values? What consequences to a state estimation algorithm could these sorts of errors have?

F. (5 pts) Suppose now that the (range, bearing) measurements are *not independent* but instead jointly correlated under the following three scenarios: a) $\rho_{r\theta} = 0.1$, b) $\rho_{r\theta} = 0.5$, and c) $\rho_{r\theta} = 0.9$. Repeat parts A and C. (Hint: use the `chol()` function in Matlab.)

5 Estimation (40 points)

Assume that we want to estimate an unobserved population parameter θ on the basis of observations x. Let f be the sampling distribution of x so that $f(x|\theta)$ is the probability of x when the underlying population parameter is θ. The function $L(\theta) = f(x|\theta)$ when viewed as a function of the parameter θ is called the likelihood function or just the likelihood. For example, if x follows a Gaussian distribution, we will have $\theta = (\mu, \sigma^2)$ and

$$\mu, \sigma \mapsto f(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$

is the likelihood.

Maximum Likelihood Estimator (MLE): The maximum likelihood method maximizes the likelihood function, leading to the MLE

$$\hat{\theta}_{MLE} = \arg\max_{\theta} L(\theta) = \arg\max_{\theta} f(x|\theta).$$

Maximum A Posteriori (MAP) Estimator: In the Bayesian framework, one can place a prior distribution over the parameter, i.e., $g(\theta)$. Then, the MAP estimator maximizes the posterior probability density function $f(\theta|x)$ as follows.

$$\hat{\theta}_{MAP} = \arg\max_{\theta} f(\theta|x) = \arg\max_{\theta} \frac{f(x|\theta)g(\theta)}{\int_{\Theta} f(x|\vartheta)g(\vartheta)d\vartheta} = \arg\max_{\theta} f(x|\theta)g(\theta),$$

were the last equality is true because the normalization constant in the Bayes' formula is independent of θ.

Remark 1. *Since* log *is a monotonic function, it is often the case that we use the logarithm of the likelihood or posterior for maximization (or negative of the logarithm for minimization).*

Remark 2. *In this case, x is a known noisy measurement of θ and its distribution is $\mathcal{N}(\mu_x, \sigma_x^2)$.*

Now suppose we have a continuous random variable $\theta \sim \mathcal{N}(\mu, \sigma^2)$. We wish to infer its mean and variance as we obtain normally distributed measurements sequentially. For the case of a random mean, μ, and fixed variance, σ^2:

A. (5 pts) Derive a formula that provides a point-estimate of the posterior θ using a MAP estimator. Place a prior over the random variable as $\theta \sim \mathcal{N}(\mu_0, \sigma_0^2)$.

B. (10 pts) Derive formulas to make a Bayesian inference so that we can infer both the mean and the variance. **Hint:** use Bayes' formula and substitute the Gaussian prior and likelihood in the formula.

C. (20 pts) You are responsible for purchasing a sensor that can measure the range (distance) to an object. Sensor I (\$100) and II (\$500) are both used to measure the range to an object. Suppose the measurements are noisy values of the range, x, such that $z \sim \mathcal{N}(\mu_z, \sigma_z^2)$ with variances of 1 (I) and 0.64 (II). The measurements obtained from these sensors can be seen in Table I and II. Parameterize the prior of x with $\mu = 0$ and $\sigma^2 = 1000$. Using the derivations from part B, write a Matlab/Python/Julia function that takes data as input and solves the inference recursively. The coding templates of Python and Julia are provided in Jupyter notebooks.

   C.1 Use the sensor data and the Matlab/Python/Julia function to infer the mean and variance of the normally distributed random variable x conditioned only on z_1.

   C.2 Use the sensor data and the Matlab/Python/Julia function to infer the mean and variance of the normally distributed random variable x conditioned only on z_2.

   C.3 Why is it that x is more precise [1] when conditioned on z_1 even though sensor II is more accurate? which sensor do you recommend to be purchased?

[1] https://en.wikipedia.org/wiki/Precision_(statistics)

Table 1: Sensor I Data

N	Z_1
1	10.6715
2	8.7925
3	10.7172
4	11.6302
5	10.4889
6	11.0347
7	10.7269
8	9.6966
9	10.2939
10	9.2127

Table 2: Sensor II Data

N	Z_2
1	10.7107
2	9.0823
3	9.1449
4	9.3524
5	10.2602

D. (5 pts) Use the Kalman filter class and write a program to solve C1 and C2. Compare your results with the MAP estimator. What is the conclusion now and why?