Group 10 – HA1 Report

SF2955: Computer Intensive Methods in Mathematical Statistics

David Haapanen Noah Eriksson Tewolde Berhane

April 15, 2025

Model

The project considered a moving vehicle in $\mathbb{R}^2$ were its dynamics could be described by

$$\mathbf{X}_{n+1} = \mathbf{\Phi}\mathbf{X}_n + \mathbf{\Psi}_z\mathbf{Z}_n + \mathbf{\Psi}_w\mathbf{W}_{n+1}, \quad n \in \mathbb{N} \tag{1}$$

with $\mathbf{X}_n$ denoting a vector containing location in space, velocity and acceleration for time step n. $\mathbf{Z}_n$ in eq. (1) denotes the stage of an independent Markov chain $\{\mathbf{Z}_n\}_{n\in\mathbb{N}}$ with a probability transition matrix $\mathbf{P}$ given in the instructions to the project. The state space for $\{\mathbf{Z}_n\}_{n\in\mathbb{N}}$ corresponds to the vehicles driving command at time step n, these include steering forward, east, west, north or south. Furthermore the terms $\mathbf{\Phi}$, $\mathbf{\Psi}_z$ and $\mathbf{\Psi}_w$ are given matrices, while $\{\mathbf{W}_n\}_{n\in\mathbb{N}}$ are bivariate i.i.d Gaussian distributed noise variables.

The vehicles dynamics as n increases by eq. (1) is however not directly observable, rather observations can only be made from six basis stations inorder to evaluate $\mathbf{X}_n$ for each time step. Thus the project consist of evaluating a hidden Markov process (HMM) to determine the vehicles trajectory as n increases.

Problem 1

Markov properties of $\{\mathbf{X}_n\}_{n\in\mathbb{N}}$ and $\{\tilde{\mathbf{X}}_n\}_{n\in\mathbb{N}}$

To determine whether $\{\mathbf{X}_n\}_{n\in\mathbb{N}}$ forms a Markov chain, we consider the motion model given eq. (1), where $\mathbf{Z}_n$ is an external input corresponding to the driving command and $\mathbf{W}_{n+1}$ is Gaussian noise. It is clear from the problem formulation that $\{\mathbf{Z}_n\}_{n\in\mathbb{N}}$ is a separate Markov chain with transition probabilities defined by the matrix $\mathbf{P}$. We know that if $\{\mathbf{X}_n\}_{n\in\mathbb{N}}$ is a Markov process, its transition probability to the next state should depend only on its previous state. Thus $\{\mathbf{X}_n\}_{n\in\mathbb{N}}$ is not a Markov chain since $\mathbf{X}_{n+1}$ depends on the state $\mathbf{Z}_n$ which is independent of the previous state $\mathbf{X}_n$.

However, since $\{\mathbf{Z}_n\}_{n\in\mathbb{N}}$ is a Markov chain and the evolution of $\{\mathbf{X}_n\}_{n\in\mathbb{N}}$ only depends on $\mathbf{X_n}$, $\mathbf{Z}_n$, and $\mathbf{W}_{n+1}$, the state $\tilde{\mathbf{X}}_n = (\mathbf{X_n}^\top, \mathbf{Z}_n^\top)^\top$ therefore includes all the relevant state information needed to describe the future state $\tilde{\mathbf{X}}_{n+1}$. Therefore, the process $\{\tilde{\mathbf{X}}_n\}_{n\in\mathbb{N}}$ satisfies the Markov property.

Simulation of the target trajectory

The trajectory was simulated by recursively generating the state $\mathbf{X}_{n+1}$ using the motion model described in eq. (1). The initial state $\mathbf{X}_0$ was drawn from a multivariate normal distribution provided by the project instructions while $\mathbf{Z}_0$ was initialized by a uniform random draw from its possible states. At each time step n, Gaussian noise $\mathbf{W}_{n+1}$ was sampled and utilized together with $\mathbf{Z}_n$ to compute the new state $\mathbf{X}_{n+1}$, according to eq. (1). Furthermore $\mathbf{Z}_{n+1}$ was computed using $\mathbf{Z}_n$ the transition matrix $\mathbf{P}$. The resulting trajectory of the vehicles location coordinates in $\mathbb{R}^2$, (X_n^1, X_n^2), was then plotted based on computed $\{\mathbf{X}_n\}_{n=0}^{m=0}$.

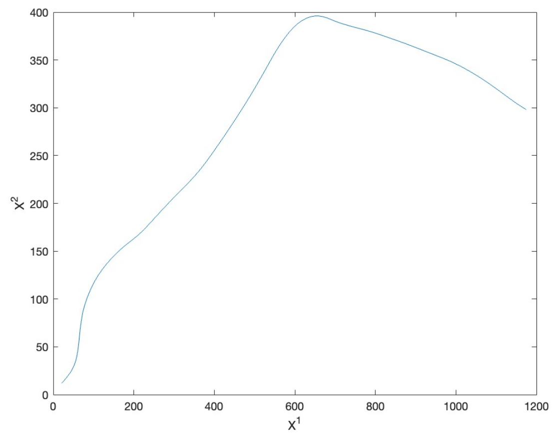


Figure 1: Trajectory path $\{(X_n^1, X_n^2)\}_{n=0}^{m}$ given $m = 100$.

Problem 2

Is $\{(\tilde{\mathbf{X}}_n, \mathbf{Y}_n)\}_{n\in\mathbb{N}}$ a hidden Markov model?

A stochastic process $\{(\tilde{\mathbf{X}}_n, \mathbf{Y}_n)\}_{n\in\mathbb{N}}$ is a HMM if it satisfies two criteria:

1. The hidden state process $\{\tilde{\mathbf{X}}_n\}_{n\in\mathbb{N}}$ forms a Markov chain.
2. The observations $\{\mathbf{Y}_n\}_{n\in\mathbb{N}}$ are conditionally independent given the hidden states, and each $\mathbf{Y_n}$ depends only on $\tilde{\mathbf{X}}_n$.

We already established that $\{\tilde{\mathbf{X}}_n\}_{n\in\mathbb{N}}$ is a Markov chain. The observations $\{\mathbf{Y_n} = (Y_n^1, Y_n^2, ..., Y_n^6)\}_{n\in\mathbb{N}}$ contains the observed data for each basis station (BS), where the observations for BS number ℓ are generated according to the model

$$Y_n^{(\ell)} = \nu - 10\eta \log_{10}\left(\left\| \begin{pmatrix} X_n^1 \\ X_n^2 \end{pmatrix} - \pi_\ell \right\|\right) + V_n^{(\ell)}, \tag{2}$$

where $V_n^{(\ell)}$ are independent Gaussian noise terms. Thus each observation $\mathbf{Y}_n$ depends only on the current position of the target, which is part of $\tilde{\mathbf{X}}_n$, and is conditionally independent of past observations and states given $\tilde{\mathbf{X}}_n$. Therefore, both criteria are fulfilled, and $\{(\tilde{\mathbf{X}}_n, \mathbf{Y}_n)\}_{n\in\mathbb{N}}$ is indeed a hidden Markov model.

Derivation of the transition density $p(\mathbf{y}_n \mid \tilde{\mathbf{x}}_n)$ of $\mathbf{Y}_n \mid \tilde{\mathbf{X}}_n$

Given $\tilde{\mathbf{X}}_n = \tilde{\mathbf{x}}_n$, each observation in the vector $\mathbf{Y}_n = (Y_n^1, Y_n^2, ..., Y_n^6)$ are conditionally independent Gaussian random variables. Thus for BS number ℓ, having location coordinates denoted as $\boldsymbol{\pi}_\ell$, the conditional distribution becomes

$$Y_n^\ell \mid \tilde{\mathbf{X}}_n = \tilde{\mathbf{x}}_n \sim \mathcal{N}\left(\nu - 10\eta\log_{10}\left(\left\|\begin{pmatrix} x_n^1 \\ x_n^2 \end{pmatrix} - \boldsymbol{\pi}_\ell\right\|\right), \varsigma^2\right).$$

By letting $\mu_n^\ell := \nu - 10\eta\log_{10}\left(\left\|(x_n^1, x_n^2)^\top - \boldsymbol{\pi}_{\boldsymbol{\ell}}\right\|\right)$, and the full mean vector be $\boldsymbol{\mu}_n = (\mu_n^1, \ldots, \mu_n^6)^\top$, it hold that $\mathbf{Y}_n \mid \tilde{\mathbf{X}}_n = \tilde{\boldsymbol{x}}_n \sim \mathcal{N}(\boldsymbol{\mu}_n, \varsigma^2\mathbf{I}_{6\times6})$. Therefore and the transition density is

$$p(\mathbf{y}_n \mid \tilde{\mathbf{x}}_n) = \frac{1}{(2\pi\varsigma^2)^3}\exp\left(-\frac{1}{2\varsigma^2}\sum_{\ell=1}^{6}\left(y_n^\ell - \mu_n^\ell\right)^2\right). \tag{3}$$

Problem 3

Sequential importance sampling (SIS) was implemented with means of computing estimates of the vehicles coordinates (X_n^1, X_n^2) for every time step n given observational data of the received signal strength indication (RSSI) from all BS, $\mathbf{y}_{0:m} = (\mathbf{y}_0, \ldots, \mathbf{y}_m)$, with $m = 500$. This was done with sequential Monte Carlo (SMC) methods by evolving $N = 10,000$ particles and corresponding weights $\{\tilde{\mathbf{X}}_{0:n}, \omega_n^i\}_{i=1}^N$, utilizing the smoothing distribution $\tilde{\mathbf{X}}_{0:n} \mid \mathbf{Y}_{0:n}$. Using the given expression of the density $f(\tilde{\mathbf{x}}_{0:n} \mid \mathbf{y}_{0:n})$ and suggested proposal kernel $q(\tilde{\mathbf{x}}_n \mid \tilde{\mathbf{x}}_{n-1})$ from the assignment the recursive equation for updating weights was obtained by applying the SIS methodology such that

$$\omega_{n+1}^i = p(\mathbf{y}_n \mid \tilde{\mathbf{x}}_n) \cdot \omega_n^i$$

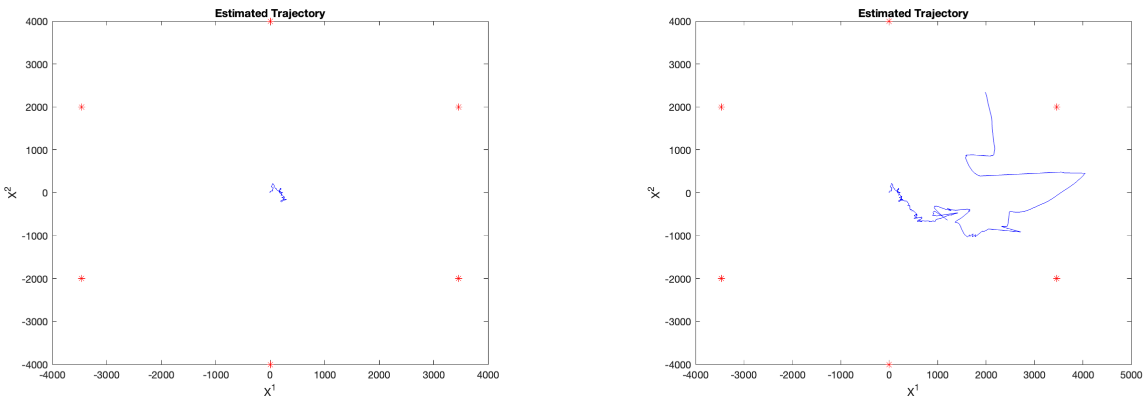
Thus the estimates $\tau_n^1 = \mathbb{E}\left[X_n^1 \mid \mathbf{Y_{0:n}} = \mathbf{y_{0:n}}\right]$ and $\tau_n^2 = \mathbb{E}\left[X_n^2 \mid \mathbf{Y_{0:n}} = \mathbf{y_{0:n}}\right]$ could be evaluated sequentially for every updated particle. A pseudo code of the algorithm is illustrated as:

Input: RSSI measurements $\mathbf{Y}_{0:m}$, number of particles N, initial distributions, state matrices;
Sample $X_0^i \sim \mathcal{N}(\mu, \sigma)$ and $Z_0^i \sim \mathcal{U}(\{z_1, \ldots, z_5\})$ for $i = 1, \ldots, N$;
Compute initial weights: $w_0^i = p(Y_0 \mid X_0^i)$;
Estimate: $\tau_0^1 = \frac{\sum_i w_0^i X_0^{1,i}}{\sum_j w_0^j}$, $\quad \tau_0^2 = \frac{\sum_i w_0^i X_0^{2,i}}{\sum_j w_0^j}$;
for $n = 1$ **to** m **do**
    Sample next state of Z: $Z_n^i \sim P(Z_n \mid Z_{n-1}^i)$;
    Sample independent white noise: $W_n^i \sim \mathcal{N}(0, \sigma^2 I)$;
    Propagate the next particle state: $X_n^i = \Phi X_{n-1}^i + \Psi_z Z_{n-1}^i + \Psi_w W_n^i$;
    Update weights: $w_n^i = w_{n-1}^i \cdot p(Y_n \mid X_n^i)$;
    Estimate: $\tau_n^1 = \frac{\sum_i w_n^i X_n^{1,i}}{\sum_j w_n^j}$, $\quad \tau_n^2 = \frac{\sum_i w_n^i X_n^{2,i}}{\sum_j w_n^j}$;
end

This algorithm, however, resulted in weights quickly approaching zero, since $p(Y_n \mid X_n^i)$ is typically a small number, and multiplying such values across many steps causes the weights to approach zero quickly. To address this, max normalization with log-weights, $\ell_n^i = \log \omega_n^i$, were used in the algorithm. For time step n, sampling N particles, estimates were obtained by transforming weights such that

$$\begin{aligned} \bar{\omega}_n^i &= \exp\left(\ell_n^i - \max_j \ell_n^j\right) \\ \bar{\Omega}_n &= \sum_{i=1}^{N} \bar{\omega}_n^i \end{aligned} \quad \Rightarrow \quad \begin{aligned} \tau_n^1 &= \frac{\sum_{i=1}^N \bar{\omega}_n^i X_n^{1,i}}{\bar{\Omega}_n} \\ \tau_n^2 &= \frac{\sum_{i=1}^N \bar{\omega}_n^i X_n^{2,i}}{\bar{\Omega}_n} \end{aligned}$$

Thus with $\{(\tau_n^1, \tau_n^2)\}_{n=0}^{m}$ obtained the estimated trajectory of the vehicle was plotted. The trajectories obtained can be seen in fig. 2



(a) SIS without log-normalized weights.

(b) SIS with log-normalized weights.

Figure 2: Trajectory of vehicle using $N = 10,000$ and regular SIS.

Comparing fig. 2a to fig. 2b it is evident that there is a difference in magnitude with regards to the distance traveled by the vehicle. It could be considered very unlikely that the vehicle would move such a small route as seen in fig. 2a, thus the small weights were likely impacting the computations of the estimates, which seems to have been adressed by the use of log-normalized weights. Furthermore analysis regarding the Effective Sample Size (ESS) and weight distribution was conducted.

Histograms of the importance weights at selected time points $n = 0$, $n = 15$, and $n = 25$ were plotted in order to visualize how concentrated the weights were. As seen in fig. 3 the weights go from initially being spread out to obtaining values closer to zero as n increases. The skewed distribution for larger n indicates a growing weight degeneracy, which would imply that the very few weight with larger values will dominate the estimates. Thus the estimation accuracy becomes worse over time, highlighting the need using resampling.

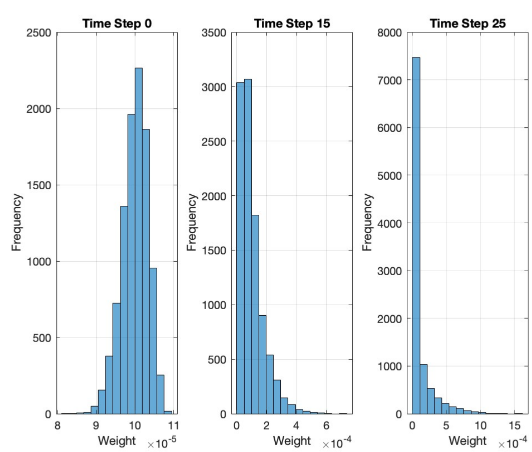


Figure 3: Histogram of weights using SIS at different time points.

The value of ESS_n approximates the number of effective particles that meaningfully contribute to the estimate at time step n. Using the normalized weights, the values of ESS_n for all time steps n were computed via the expression $\text{ESS}_n = \frac{1}{\sum_{i=1}^{N}(\bar{w}_n^i)^2}$. These were then plotted a graph, as seen in fig. 4, which show that the ESS value decreases from nearly $10,000$ to zero for $n \approx 100$. This means that almost all samples contribute to the estimates in the beginning but this number will decrease rapidly until only a few particle samples will dominate in contribution to the estimates, making the variance very high. This is also an indication of the need of resampling.

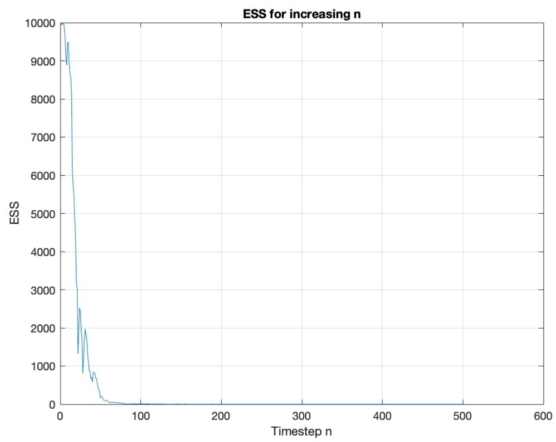


Figure 4: ESS for increasing n.

Problem 4

The trajectories were now estimated using sequential importance sampling with resampling (SISR). This was done with small adjustments to the algorithm described in the previous problem. Specifically, a resampling step was introduced at each iteration before propagating the particle samples and their weights. Given the sample $\{\tilde{\mathbf{X}}_{0:n}^i\}_{i=1}^N$, particles were selected according to their normalized importance weights before propagation, that is the particles with larger weights would be selected with a higher probability. Thus the particles with higher importance would be selected in the resampling step which would combat the problem of weight degeneracy. The recursive formula for updating the weight would also be changed to that each weight after the resampling step would be set to one, hence the updating equation would be $\omega_{n+1}^i = p(\mathbf{y}_n \mid \tilde{\mathbf{x}}_n)$

The estimated trajectory of the vehicle using SISR was plotted, seen in fig. 5, which shows a more smooth and precise vehicle path than fig. 2b indicating a more plausible trajectory with less variance.

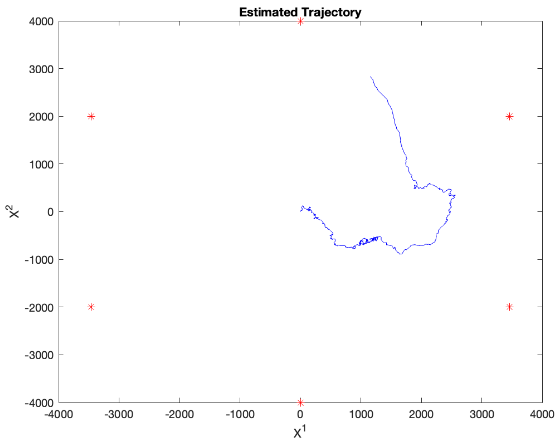


Figure 5: Estimated vehicle trajectory using SISR and N = 10000.

Histograms of the importance weights at $n = 0$, $n = 15$, and $n = 25$ were once again plotted which can be seen in fig. 5 which illustrates how the values of the weight are spread out even for larger n values. Thus the weight degeneracy is not as prevalent as when SIS was implemented.

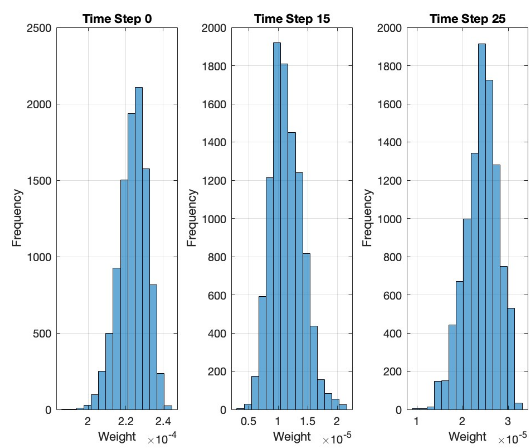


Figure 6: Histogram of weights using SISR at different time points.

The ESS values for all time steps were also plotted as in the previous problem, which can be seen in fig. 7. It can be seen that the ESS values fluctuate, yet does not decrease to have very small values as seen in fig. 4. This is to be expected since particles with larger weights are more probable to be resampled, hence the particles that are most likely to contribute to the estimates are selected and duplicated. This also shows how the weight degeneracy problem with SIS is combated by resampling. For SISR, log-normalization is not needed, and thus the formula used for computing the ESS is

$$\text{ESS}\left((\omega_n^i)_{i=1}^N\right) = \left(\sum_{i=1}^{N} \left(\frac{\omega_n^i}{\Omega_n}\right)^2\right)^{-1}$$

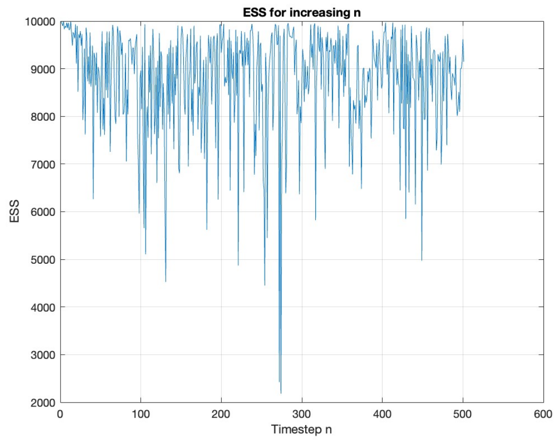


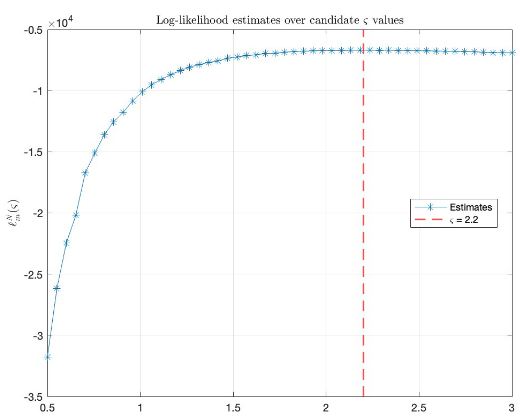
Figure 7: ESS for increasing n.

Problem 5

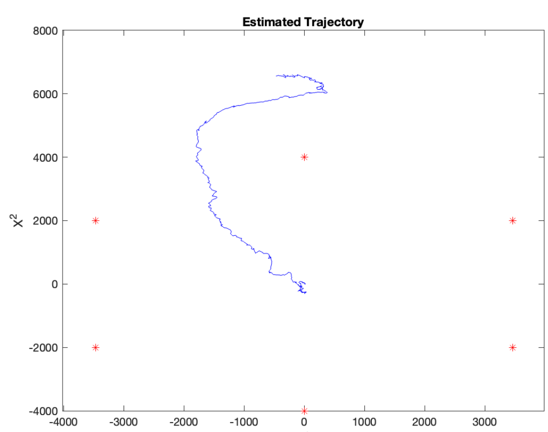
Considering new RSSI data the value of $\varsigma \in (0,3)$ was considered to be unkown. The objective was then to evaluate ς utilizing an approximate maximum likelihood approach coupled with the SISR algorithm developed from before. First a equidistant discretization of $\{\varsigma_j\}$ was defined for a set interval. Then for each ς_j, the SISR algorithm was applied to compute

$$\ell_m^N(\varsigma_j) = \sum_{n=0}^{m} \log\left(\frac{\Omega_n}{N}\right), \quad \text{where} \quad \Omega_n = \sum_{i=1}^{N} \omega_n^i \quad \text{, and } m = 501$$

which, approximates the normalized log-likelihood $\ell_m(\varsigma, y_{0:m})$. The value $\hat{\varsigma}$ that maximized $\ell_m^N(\varsigma_j)$ was then chosen to be an estimate of ς. A plot of the log-likelihood estimates depending on ς can be seen in fig. 8a, indicating that final estimate maximum the likelihood was determined to be $\hat{\varsigma} \approx 2.20$. This value was then finally used to estimate the trajectory based on the new given data, which is illustrated in fig. 8b



(a) Plot of log-likelihood values for the varying values of ς.



(b) Estimated trajectory using SISR with $\hat{\varsigma} = 2.20$.

Figure 8: $\hat{\varsigma}$ and its corresponding estimated trajectory.