System Identification and Estimation

Master of Science in Aerospace Engineering

Particle Filter Lab

Authors:

Rajashree SRIKANTH rajashree.srikanth@student.isae-supaero.fr
Nivetha SRINIVASAN nivetha.srinivasan@student.isae-supaero.fr

Tutor:
Yves BRIERE

October 20, 2023

2023/2024 – 3rd Semester

1 Introduction

This report highlights the results of the particle filter exercise conducted during the course. The objective of this exercise is to use a particle filter to conduction the trajectory estimation of a given particle. The given template was modified to include the prediction and the update algorithm for the particle filter. For the analysis, the state and the measurement parameters were analysed with respect to the variance of the state. Additionally, the number of particles were varied to perform an analysis.

2 Theory

A particle filter is a recursive Bayesian filter used for state-estimation. The algorithm gives an approximation of the posterior distribution taking a discrete density of the form. It represents the probability density of the state as a set of weighted particles. These particles are updates with the likelihood of these particles. Particle Filters are useful for tracking the state of dynamic systems in cases where the system is non-linear and/or the noise is non-Gaussian. They are commonly used in fields like robotics, computer vision, and localization, among others, to estimate the state of a system given noisy measurements. The key to the effectiveness of Particle Filters is the representation of the state as a set of particles, allowing it to handle complex, non-linear, and non-Gaussian dynamics.

2.1 Particle Filter Algorithm

The particle filter estimates the state through an iterative process involving a reference dynamic model through information given by the measurement and the estimate of the state from the previous time step. The steps involves in this process are:

- Initialization:
$$p(x_0) \tag{1}$$
- Prediction:
$$p(x_k|y_k) \propto p(y_k|x_k) * p(x_k|y_{k+1}) \tag{2}$$
- Update:
$$p(x_{k+1}|y_k) = p(x_k|y_k) * p(x_{k+1}|x_k)dx_k \tag{3}$$

2.2 Algorithm implemented in the code

In the given particle filter exercise, the main objectives include figuring out the prediction step and the update step.

The state evolution is predicted using the given circular trajectory. The process noise is introduced at this step to simulate uncertainty. (x_i) is the state that is evolving with respect to the trajectory and v is the noise introduced. Here, the iteration is done through all particles (from 1 to N), and for each particle, the motion model is applied to predict the next position.

After the prediction step, the measurements are obtained from the system. In the measurement update step, particle weights are adjusted based on the likelihood of each particle's measurement compared to the actual measurement. The prediction obtained in the previous is used to determine how well each particle's predicted state matches the measurements. This likelihood for each particle is multiplied with the weights of each particle to update the weights of the particles.

The implementation of this algorithm on MATLAB is illustrated below:

```
%Predict
        v = chol(Q)*randn(2,N);
         xi = xi + [-2*pi/Tmax*sin(2*pi*t/Tmax);
         2*pi/Tmax*cos(2*pi*t/Tmax)] + v;

    %update particles weights
    for i = 1:N
        %Wi(i) = 1/N;
        particle_measurement = xi(:, i);
        likelihood = 1 /(2 * pi * sqrt(det(R)))
        * exp(-0.5 * (ytrue - particle_measurement)' * inv(R) * (ytrue -
   particle_measurement));

        % Update the particle weight
        Wi(i) = Wi(i) * likelihood;
    end
    Wi = Wi/sum(Wi); %Normalize
```

Figure 1 indicates the results of the prediction algorithm for the case of $N_{eff} <= 0.4$, $N = 1000$ and noise variance factor $= 1$. The black marker depicts the true position of the particle, while the red scatter represents the possible locations of the particle distributed along a probability density curve. The red ellipsoid curve encloses the locations of the most heavily weighted particles.

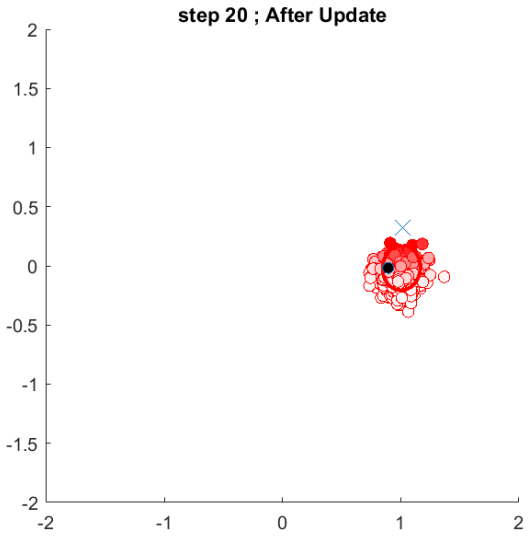

Figure 1: Prediction of True Position of Particle

3 Analysis

3.1 Parameters Tuned

Key parameters in the particle filtering algorithm such as the state noise variance (Q), sensor measurement noise variance (R), re-sampling threshold (N_{eff}) and particle count (N) are varied to understand their effect on the confidence in prediction of particle true position. The corresponding RMS error and variance for these parameters are studied and are tabulated in Tables 1, 2 and 3.

S.No.	N	Multiplier	Max Variance	MSE	Q	R
1	10	0.3	0.02	0.4456	1	1
2	100	0.3	0.0133	0.5709	1	1
3	1000	0.3	0.0104	0.6169	1	1
4	10000	0.3	0.0129	0.6758	1	1
5	100000	0.3	0.0144	0.8658	1	1
1	10	0.4	0.0162	0.4802	1	1
2	100	0.4	0.0151	0.6591	1	1
3	1000	0.4	0.0101	0.6525	1	1
4	10000	0.4	0.0126	0.6813	1	1
5	100000	0.4	0.0162	0.8288	1	1
1	10	0.5	0.0102	0.4343	1	1
2	100	0.5	0.0073	0.446	1	1
3	1000	0.5	0.0105	0.5847	1	1
4	10000	0.5	0.00103	0.6644	1	1
5	100000	0.5	0.0112	0.7903	1	1

Table 1: Tabulation of Data for Unitary Noise Factor

S.No.	N	Multiplier	Variance	MSE	Q	R
1	10	0.4	0.0501	0.8203	5	5
2	100	0.4	0.0376	0.9399	5	5
3	1000	0.4	0.0411	1.1586	5	5
4	10000	0.4	0.0451	1.4138	5	5

Table 2: Tabulation of Data for Noise Factor = 5

S.No.	N	Multiplier	Variance	MSE	Q	R
1	10	0.4	0.1124	1.6035	10	10
2	100	0.4	0.0759	1.3803	10	10
3	1000	0.4	0.093	1.7834	10	10
4	10000	0.4	0.0752	1.927	10	10

Table 3: Tabulation of Data for Unitary Noise Factor = 10

3.2 Varying the Multipliers

As the values of the multipliers for the state and the measurement noise are varied, the minimum variance for the number of particles is noted. The data collected above is used to generate plots to study the trends affected by these key parameters. Figures 2 and 3 are used to tune and determine the optimal particle count. It can be observed that $N = 1000$ provides the least variance and RMS error. This is especially evident from Figure 3.

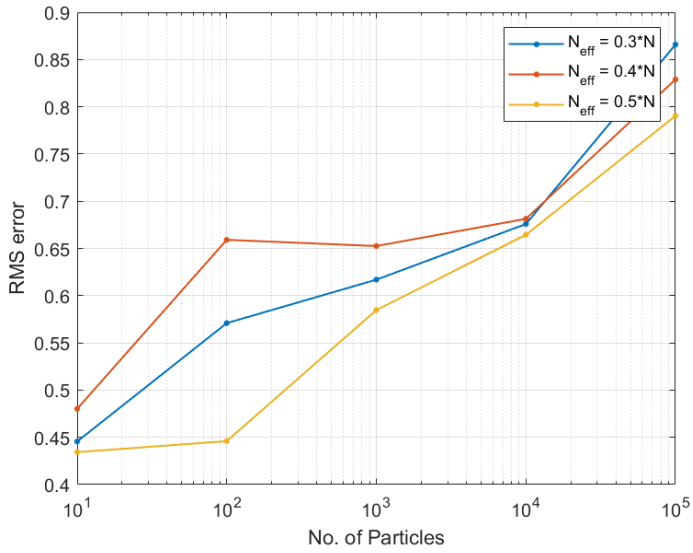

Figure 2: RMS Error for Unitary Noise Factor for Various N_{eff} Values

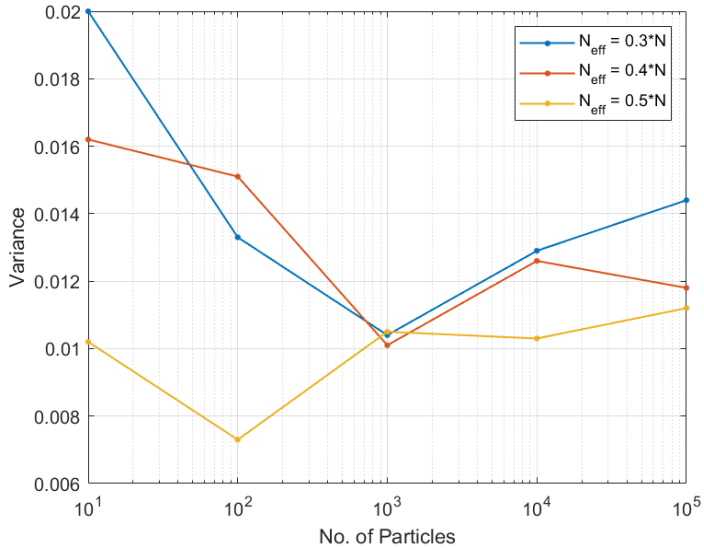

Figure 3: Variance for Unitary Noise Factor for Various N_{eff} Values

Tables 2 and 3 are translated into curves in Figures 4 and 5 respectively. The non-unitary noise factors introduce larger levels of noise into the readings, which result in increased variance,

implying more randomness and scattering of particles. The values of variance no longer follow a predictable trend, with this effect being more prominent for the case of noise factor = 10. Hence, it is required to keep the noise levels in check while performing the state estimation using this algorithm.

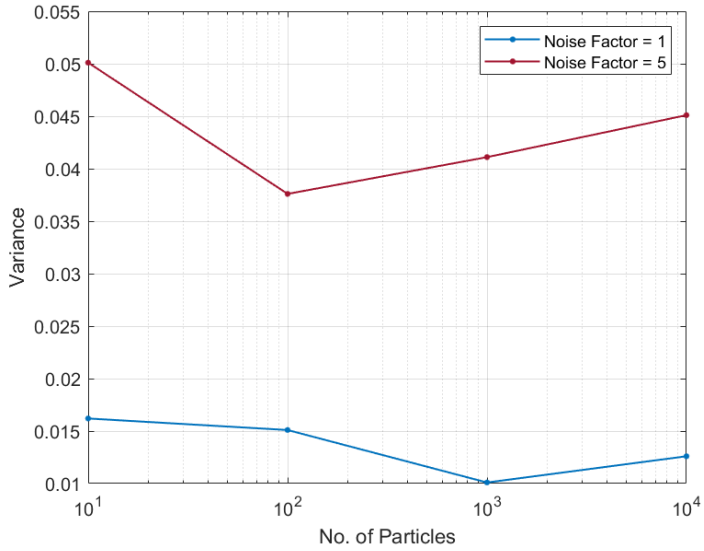

Figure 4: Variance Trend Curves for Noise Factor = 5

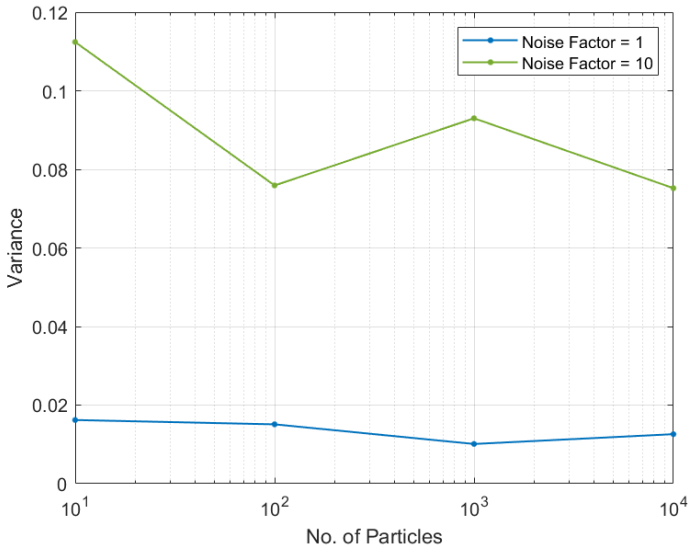

Figure 5: Variance Trend Curves for Noise Factor = 10

Finally, the variance for the chosen optimal value of particle count = 1000 for different conditions of resampling are presented in Figure 6. $N_{eff} = 0.4$ seems to have larger variance in comparison to the other values, suggesting that either of the other two values could be a better choice for confident prediction of the true state. On observing and comparing the variance data between $N_{eff} = 0.3$ and $N_{eff} = 0.5$ suggests that the former could be more optimal.

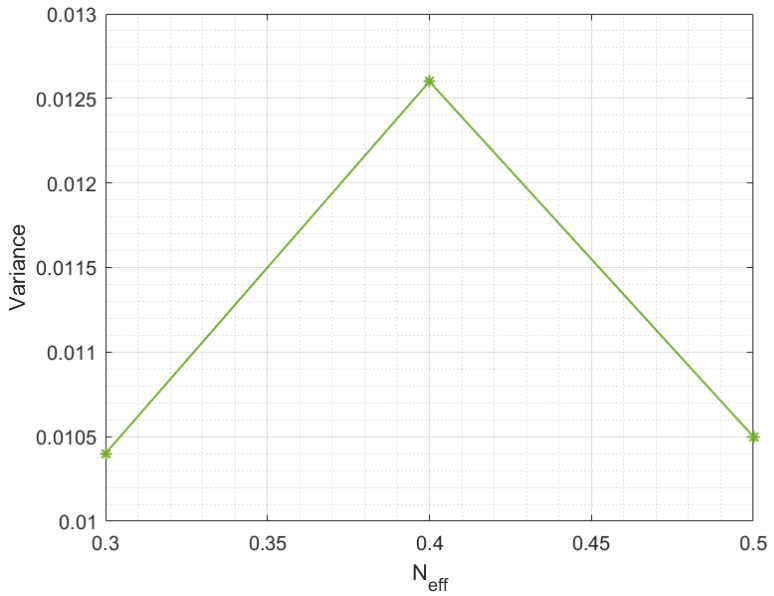

Figure 6: Dependence of Variance on N_{eff} parameter