A Novel Adaptive Marginalized Particle Filter for Mixed Linear / Nonlinear State-Space Models

Sanil Jayamohan [1], Santhi Nagarajan [2], Ramar Kadarkaraiyandi [3]

[1, 2] *Department of Electronics & Communication Engineering,* Noorul Islam Centre for Higher Education, Kumaracoil, Kanyakumari District, Thuckalay, Tamil Nadu, India.
[3] *Department of Computer Science & Engineering*, Einstein College of Engineering, Sir C.V.Raman Nagar, Seethaparpanalur, Tirunelveli, Tamil Nadu, India.
Email: [1] saniljayamohan@gmail.com, [2] santhiram@yahoo.com, [3] kramar.einstein@gmail.com.

***Abstract* - The speed with which the state parameters can be estimated with acceptable variance is an important issue in any target tracking application. In most of these problems, a linear Gaussian substructure is present in the model which is made use of and estimation is usually done using Marginalized Particle Filter (MPF) than the most general Particle Filter (PF). An important parameter that affects the performance of such filters is the amount of noise present in the measurement which may not remain constant, varying between a low value and a high value. The main aim of this paper is to introduce a new filter known as Adaptive Marginalized Particle Filter (AMPF) which exploits this property of noise thereby adapting the number of particles whenever possible and hence improving the estimation speed. The proposed filter is able to estimate the state parameters much faster than the existing filters. The performance of AMPF is compared with that of MPF and PF using a typical target tracking example. From the simulation, it can be concluded that AMPF is better than MPF and PF in terms of execution time, without much affecting its performance in terms of RMSE.**

Keywords - State Estimation, Target Tracking, Marginalized Particle Filter, Particle Filter, Gaussian.

I. INTRODUCTION

In Science, in order to understand how a system works, it is important to get access to certain important parameters of the system. In most of the situations, these parameters may not be directly available and they need to be estimated from noisy measurements which may or may not be directly related. Dynamic state estimation is the technique by which these parameters can be estimated in order to find out the nature and properties of the system. Such problems are usually solved using Bayesian approach. In many applications like collision avoidance, surveillance and guidance systems, target tracking is considered as one of the important element.

Target Tracking is a state estimation technique in which certain parameters of the system is estimated from a set of noisy measurements. There are several parameters that determine the performance of the filter such as the quality of the dynamic [1], [2] and measurement models, amount of noise present in the measurements etc. There are different types of models associated with target tracking problems. It is very important that the model that is used must exactly resemble the actual system. No matter how good the filter or the estimation method is, if the dynamic model does not accurately resemble the actual system, the performance of the system will not be satisfactory. Other factors that affect the performance of the system are how effective the parameters can be extracted from the noisy measurement and also the quality of the filter that is being used.

The main advantage of target tracking problems is that it can be modeled as state estimation problem. Two models are associated with such estimation problems known as the state transition model which shows how, different states of the target such as position, velocity, acceleration etc. evolve with respect to time and the measurement model that gives the relation between the states and observations.

Based on the type of the model and the noise associated with the model different types of filter are available. If the model is linear and the noise associated with the model is Gaussian, then optimal filter like Kalman filter (KF) [3], [4] can be made use of. When nonlinearities are associated with the model, other versions of KF like Extended Kalman Filter can be made use of which are found to be efficient when the nonlinearities are small. The other versions of Kalman Filter are based on linearization and Gaussian approximation techniques. When nonlinearities are large, the performance of these techniques are low. In order to address such type of situations, another type of state estimation techniques known as Sequential Monte Carlo (SMC) methods are used which have better performance. SMC methods are also known as Particle Filters (PF) [5], [6].

A set of weighed samples known as particles are used to represent the aposterior probability density function in Particle filters. PF will obtain the optimal estimate when the distribution is represented using large number of particles. The performance of PF is satisfactory if the state variable dimension is small and it provides the most general solution. However, the performance of PF deteriorates beyond the acceptable level as soon as the system dimension increases.

There is a special type of PF known as the Gaussian Particle Filter [7], [8] which is found to be effective as well as it reduces the complexity of PF when the system is nonlinear and the noise associated with the model is Gaussian.

Another problem that affects the performance of PF is Degeneracy, in which the weights associated with the particles reduce to zero, which makes those particles ineffective in the estimation process. The solution to this problem is obtained by a technique known as resampling [9], [10] in which a new set of particles are created. Resampling process of PF is unavoidable in most of the situations, which increases the complexity of the filter.

When the model associated with the system is composed of a linear Gaussian substructure there is a possibility of improvement in the performance. There is a special type of PF known as Marginalized Particle Filter [11], [12], [13] which is based on marginalization or Rao-Blackwellization process. In this technique, the state vector is partitioned into linear and non-linear parts that are estimated as two independent processes. The final result is obtained by combining the two results. Suppose $\boldsymbol{x}_t$ represent the state vector consisting of both linear and non-linear dynamics, then in MPF it will be portioned as:

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{x}_t^l \\ \boldsymbol{x}_t^n \end{bmatrix} \quad (1)$$

where $\boldsymbol{x}_t^l$ and $\boldsymbol{x}_t^n$ represents linear and non-linear state variable vector respectively. In this paper superscript l and n are used to denote linear & nonlinear components. Many engineering applications such as passive target tracking, bearing only tracking, positioning, collision avoidance, spiraling Ballistic Missile state estimation etc. [14], [15], [16] contains this type of model. MPF [12] is a filter that utilizes this structure present in the model to obtain estimates having less variance when compared to that of standard PF. In MPF, Kalman filter is used to estimate the linear state variables and complex PF estimates the non-linear state variables. This process makes sure that the dimension of PF is small thereby reducing the overall complexity of MPF. The main advantages of MPF, when compared to that of standard PF, is that MPF has less complexity as well as the estimates obtained will be much better than that obtained from Particle Filters.

Noise present in the measurement data is another important parameter that determines the performance of these filters. In a practical situation, the amount of noise present in the measurement may not always be the same. Sometime the noise may be very low and other times, it may be very high, which makes the filter difficult to operate. Filters like MPF and PF are designed to operate with a constant number of particles. However, as the performance of MPF is better than PF hence there is a possibility to make use this aspect of noise present in the measurement. Based on the varying nature of measurement noise, a new filter is proposed known as the Adaptive Marginalized Particle Filter (AMPF) which adapts itself according to the amount of the noise present. The proposed filter will be able to perform estimation much faster than the existing MPF with an acceptable increase in the variance.

This paper is involved in the development & performance analysis of AMPF in the case of models containing a Linear-Gaussian substructure. The performance of AMPF is much better than MPF and PF, in terms of speed of execution and it does not much affect the variance of the estimate. In this paper, the analysis of the performance of AMPF is done with the assumption that, there is no signal propagation delays and both measurement and process noise are assumed to be independent of each other. The effect of signal propagation delays and dependent noise processes in the case of PF is discussed in [17,18]

Section 2 gives a brief discussion about the related work, Section 3 explains the basic idea behind AMPF in the case of general linear/nonlinear state space models. An important model class is discussed in section 4. To illustrate the performance of AMPF, a typical target tracking application is considered in section 5, simulation results are given in section 6 and section 7 vividly comprehend the conclusions.

II. RELATED WORK

Kalman filter, Particle filter and Marginalized Particle Filter are the most popular three algorithms that dominated state estimation problems. Of which Kalman filter obtained the optimal estimate in the case of linear systems, while Particle Filter obtained state estimates in most general linear/ nonlinear models. Presence of linear Gaussian substructure in the models of state estimation problems have made Marginalized Particle Filter a better choice when compared to the standard particle filter and Kalman filter where nonlinearity cannot be handled. Complexity of Particle filter is an important factor that decides the applicability of the algorithm in different state estimation problems. When compared to PF, MPF is much less complex and at the same time achieve state estimation with reduced variance. The complexity analysis MPF and PF from a theoretical point of view based on floating point operations is best explained in [19]. Noise is another important parameter that determines which types of these algorithms is used in state estimation. While Kalman filter and Marginalized Particle Filter can handle only Gaussian type of noise, Particle filter being a standard filter can work well in the presence of both Gaussian and Non Gaussian Noise. Noise Tolerance capabilities of Marginalized Particle Filter and Particle Filter in the case of Non-Maneuvering trajectory in terms of Constant RMSE simulation, Constant Execution Time simulation and Different Measurement noise covariance's is well discussed in [20]. In order to understand the effect of noise present in the measurement on the performance of MPF & PF, a detailed analysis including Different Measurement Noise Covariance's and Effect of

Measurement Noise Spike is done in [21]. The noise tolerance property of MPF obtained in [20] & [21] is the motivation behind the proposed algorithm.

III. ADAPTIVE MARGINALIZED PARTICLE FILTER

The amount of noise present in the observation is an important parameter which affects the performance of the filter. In practical scenarios it can be noted that the amount of noise affecting the observations may not be same at all the time, varying between a low value and a high value. The main reason behind the reduction of complexity of MPF when compared with that of PF is that, the particles in MPF have lesser dimension. Thus, if the number of particles in MPF can be reduced furthermore, then the complexity can be further reduced. As the noise tolerance property of MPF is much better than PF, when the amount of noise present in the observations is low, the number of particles associated with the filter can be reduced thereby reducing the complexity. As MPF is more tolerant to noise, the effect of reducing the number of particles on the performance will be less. The noise tolerant property of MPF [20,21] is the basic idea behind AMPF.

In AMPF, the number of particles associated with the filter adapts itself according to the amount of noise present in the observations. So whenever there is a scope for improvement AMPF will adapt itself thereby producing estimates faster without affecting its variance much. AMPF has all the advantages of MPF with further reduced complexity. AMPF reduces the complexity compared to that of MPF and PF by the adapting itself according to the amount of noise present in the measurement data. So whenever the amount of noise is low, the number of particles is reduced which results in the reduction of the complexity. This reduction in the number of particles will not affect the performance of the filter much, due to the re-initialization step of AMPF.

The detailed algorithm of AMPF is explained by considering the general model. Consider the general model [12] given below.

$$\boldsymbol{x}_{t+1}^{n} = \boldsymbol{f}_t^n(\boldsymbol{x}_t^n) + \boldsymbol{A}_t^n(\boldsymbol{x}_t^n)\boldsymbol{x}_t^l + \boldsymbol{G}_t^n(\boldsymbol{x}_t^n)\boldsymbol{v}_t^n \quad (2a)$$

$$\boldsymbol{x}_{t+1}^{l} = \boldsymbol{f}_t^l(\boldsymbol{x}_t^n) + \boldsymbol{A}_t^l(\boldsymbol{x}_t^n)\boldsymbol{x}_t^l + \boldsymbol{G}_t^l(\boldsymbol{x}_t^n)\boldsymbol{v}_t^l \quad (2b)$$

$$\boldsymbol{y}_t = \boldsymbol{h}_t(\boldsymbol{x}_t^n) + \boldsymbol{C}_t(\boldsymbol{x}_t^n)\boldsymbol{x}_t^l + \boldsymbol{e}_t \quad (2c)$$

where $\boldsymbol{x}_{t+1}^n, \boldsymbol{x}_{t+1}^l, \boldsymbol{x}_t^n, \boldsymbol{x}_t^l$ are the nonlinear and linear state variables for $(t+1)^{th}$ and t^{th} time interval, $\boldsymbol{y}_t$ represents the measurement vector at t^{th} time interval, $\boldsymbol{A}_t^n(\boldsymbol{x}_t^n), \boldsymbol{A}_t^l(\boldsymbol{x}_t^n), \boldsymbol{G}_t^n(\boldsymbol{x}_t^n), \boldsymbol{G}_t^l(\boldsymbol{x}_t^n), \boldsymbol{C}_t(\boldsymbol{x}_t^n)$ represents constant matrices, $\boldsymbol{f}_t^n(\boldsymbol{x}_t^n), \boldsymbol{f}_t^l(\boldsymbol{x}_t^n), \boldsymbol{h}_t(\boldsymbol{x}_t^n)$ represents functions related to nonlinear, linear and measurement variables, the process noise and measurement noise affecting the system is represented by $\boldsymbol{v}_t, \boldsymbol{e}_t$, is assumed to be Gaussian white noise with distribution:

$$\boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{v}_t^l \\ \boldsymbol{v}_t^n \end{bmatrix} \sim \mathcal{N}(0, \boldsymbol{Q}_t)\,;\, \boldsymbol{Q}_t = \begin{bmatrix} \boldsymbol{Q}_t^l & \boldsymbol{Q}_t^{ln} \\ \boldsymbol{Q}_t^{ln^T} & \boldsymbol{Q}_t^n \end{bmatrix} \quad (3)$$

$$\boldsymbol{e}_t \sim \mathcal{N}(0, \boldsymbol{R}_t) \quad (4)$$

Furthermore, $\boldsymbol{x}_0^l$ represents initial linear state variables that is assumed to be Gaussian and $\boldsymbol{R}_t$, $\boldsymbol{Q}_t$ are the variance of measurement noise and process noise respectively.

$$\boldsymbol{x}_0^l \sim \mathcal{N}(\overline{x_0}, \overline{P_0}) \quad (5)$$

where $\overline{x_o}, \overline{P_o}$ are the mean value and covariance matrix of the initial linear state variables.

The probability density of $\boldsymbol{x}_0^n$ is assumed to be known and arbitrary. The linear state variables from $p(\boldsymbol{x}_t|\boldsymbol{Y}_t)$ is marginalized out and Bayes theorem is applied giving: $(\boldsymbol{X}_t^n = \{\boldsymbol{x}_i^n\}_{i=0}^t)$.

$$p(\boldsymbol{x}_t^l, \boldsymbol{X}_t^n|\boldsymbol{Y}_t) = \underbrace{p(\boldsymbol{x}_t^l|\boldsymbol{X}_t^n, \boldsymbol{Y}_t)}_{OPTIMAL\ KF}\underbrace{p(\boldsymbol{X}_t^n|\boldsymbol{Y}_t)}_{PF} \quad (6)$$

Here the probability density $p(\boldsymbol{x}_t^l|\boldsymbol{X}_t^n, \boldsymbol{Y}_t)$ is analytically tractable. That means, the probability density $p(\boldsymbol{x}_t^l|\boldsymbol{X}_t^n, \boldsymbol{Y}_t)$ can be estimated using Kalman filter and Particle Filters are used to estimate $p(\boldsymbol{X}_t^n|\boldsymbol{Y}_t)$. The dimension of $p(\boldsymbol{X}_t^n|\boldsymbol{Y}_t)$ is less than $p(\boldsymbol{x}_t^l|\boldsymbol{X}_t^n, \boldsymbol{Y}_t)$ i.e., the dimensional requirement of AMPF is less compared to that of PF which is the reason behind the improvement in its performance. The general formulation of the AMPF can be explained as follows. Initialization step of AMPF draws N particles from a normal distribution having initial mean and covariance values. This step also initializes other variables used in the AMPF algorithm like t_{count}, cumulative ($\sigma_{c_{wt}}^2$) and average cumulative variance ($\sigma_{avgc_{wt}}^2$) of weights to zero. Once initialization step is over, next step is evaluation of importance weights which is done whenever a measurement is available. Importance weight is an indication of how good the measurement data is. That is, if the measurement data contains less noise (accurate data) then the weights will have large value and spread throughout the distributions, i.e. variance of weight will be large and if the noise affecting the measurement data is more. which means useful information is less, then all the weights will have value closer to zero and hence the variance of importance weight will be less.

Thus the variance of importance weight is an indication of the quality of the measurement data, i.e. more the importance weight variance better the measurement data. This idea is incorporated in the adaptations step of AMPF which in turn is done by calculating the cumulative and average cumulative variance of importance weight. Average cumulative variance of importance weight is evaluated using the cumulative variance of weights over different time samples. Average cumulative variance of weights is evaluated to get a stable indication regarding the quality of measurement data. When the average cumulative variance of weight is large enough, which is a clear indication that the amount of noise affecting the filter is less over the time samples, there is a scope for reduction in the number of particles, thereby reducing the complexity and hence making the filter faster. This improvement in the execution speed does not affect the performance as the noise affecting the system is less.

The reduction in the number of particles depends on two parameters. First, the value of variance of weights that determines the noise affecting the system is less, which is defined as the variance threshold value ($\sigma^2_{thresval_{wt}}$). That is, if the variance of weights is above the $\sigma^2_{thresval_{wt}}$, it shows that the noise affecting the system is less and a value less than $\sigma^2_{thresval_{wt}}$, indicates that the noise is more and it is not possible to reduce the number of particles without affecting the performance of the filter. $\sigma^2_{thresval_{wt}}$ lies between 0 and 1. If $\sigma^2_{thresval_{wt}}$ is 0 then it means that AMPF always works with reduced number of particles while a value of 1 indicates that it will work with maximum number of particles of the system. The $\sigma^2_{thresval_{wt}}$ depends on the required accuracy of the filter, i.e. performance. Variance threshold value is a tradeoff between speed and performance. A value closer to 0 improves speed but affects performance while a value closer to 1 improves performance but affects speed. Thus threshold value is chosen based on the applications requirement.

Second parameter that determines the reduction in the number of particles is improvement factor. Improvement factor indicates the maximum improvement in speed that can be achieved. The number of particles will be reduced depending upon the improvement factor. Improvement factor is an integer value greater than 1. A value of 1 indicates no reduction in the number of particles. If improvement factor is 2, then it indicates the number of particles will be reduced by a factor of 2, thereby increasing the speed by a factor of 2. i.e. 50 % reduction in the number of particles which will result in a maximum of 50 % improvement in execution speed. Thus number of particles will be varied between 50 % and 100 % of the total number of particles depending upon on the amount of noise present in the measurement data which is determined by $\sigma^2_{thresval_{wt}}$. Improvement factor depends on the improvement in speed required by the application.

Updated number of particles are calculated based on the $\sigma^2_{thresval_{wt}}$ and improvement factor. Once the updated number of particles is calculated re-initialization step is done creating a new pdf with updated number of particles. Re-Initialization step has two effects, first it changes the dimension of particles to the updated number of particles and secondly, as a new distribution is created, the accumulated noise will be removed and thereby improving the performance of the filter further.

Once the distribution is re-initialized using the updated number of particles, importance weight is again calculated and measurement update is done. Resampling is done if the particle degeneracy problem occurs. After resampling, Kalman filter measurement update is done to update the linear state variables. The distribution is propagated by performing particle filter and Kalman filter time update which will predict the next time sample values of state variables. Thus prediction and correction are done in tandem to perform estimation process. Summary of AMPF algorithm is given below.

Algorithm: Adaptive Marginalized Particle Filter (AMPF)

1. Initialization of the particles, i = 1,2…N, N is the number of particles:

$$x^{n,(i)}_{0\,|-1} \sim p_{x^n_0}(x^n_0)\ set, \left\{x^{l,(i)}_{0\,|-1}, P^{(i)}_{0\,|-1}\right\} = \{\bar{x}^l_0, \bar{P}_0\}, t_{count} = 0, \sigma^2_{c_{wt}} = 0.$$

where t_{count} is the number of time samples taken to calculate the average cumulative variance of weight ($\sigma^2_{avgc_{wt}}$), $\sigma^2_{c_{wt}}$ is the cumulative variance of weight.

2. For i=1, 2, …N, evaluate the importance weights

$$w^{(i)}_t = p(y_t | X^{n,(i)}_t, Y_{t-1})$$

3. Updating the number of particles

3.1 Calculate the cumulative variance of weight, $\sigma^2_{c_{wt}}$

$$\sigma^2_{c_{wt}} = \sigma^2_{c_{wt-1}} + \sigma^2_{wt}$$

3.2 Calculate the average cumulative variance of weight, $\sigma^2_{avgc_{wt}}$

$$\sigma^2_{avgc_{wt}} = \frac{\sigma^2_{c_{wt}}}{t_{count}}$$

3.3 Update N

$$N_{New} = \frac{N}{improvement\ factor}; if\ \ \sigma^2_{avgc_{wt}} > \sigma^2_{thresval_{wt}}$$

$$\begin{matrix} N_{New} = N \\ t_{count} = t_{count} + 1 \end{matrix}; if\ \ \sigma^2_{avgc_{wt}} < \sigma^2_{thresval_{wt}}$$

where $0 < \sigma^2_{thresval_{wt}} < 1$, improvement factor >1 and N_{New} is the updated number of particles.

3.4 Re-Initialization of particles

Do step 3.4 for $N_{New} \neq N$

Re-Initialization of the particles, i = 1,2… N_{New} :

$$\hat{x}^{l_u,(i)}_{t|t-1} \sim \mathcal{N}\left(\bar{x}^l_{t|t-1}, P_{t|t}\right)$$
$$x^{n_u,(i)}_{t|t-1} \sim \mathcal{N}\left(\bar{x}^n_{t|t-1}, \sigma^2_{x^n_{t|t-1}}\right)$$

3.5 $$set\ N = N_{New}, \hat{x}^{l,(i)}_{t|t-1} = \hat{x}^{l_u,(i)}_{t|t-1}, \hat{x}^{n,(i)}_{t|t-1} = \hat{x}^{n_u,(i)}_{t|t-1}, \sigma^2_{avgc_{wt}} = 0, t_{count} = 0.$$

where $\hat{x}^{l_u,(i)}_{t|t-1}, \hat{x}^{n_u,(i)}_{t|t-1}$ represents the updated distribution of linear & non-linear state variables respectively.

4. For i=1,2…..N, evaluate the importance weights $w^{(i)}_t = p(y_t \mid X^{n,(i)}_t, Y_{t-1})$ and normalize

$$\tilde{w}^{(i)}_t = \frac{w^{(i)}_t}{\sum_{j=1}^{N} w^{(i)}_t}$$

5. PF measurement update (resampling): Resample N particles with replacement

$$\Pr(x^{n,(i)}_{t|t} = x^{n,(j)}_{t|t-1}) = \tilde{w}^{(i)}_t$$

6. PF time update and KF:

6.1 KF measurement update:

$$\hat{x}^l_{t|t} = \hat{x}^l_{t|t-1} + K_t(y_t - h_t - C_t\hat{x}^l_{t|t-1})$$
$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T$$
$$M_t = C_t P_{t|t-1} C_t^T + R_t$$
$$K_t = P_{t|t-1} C_t^T M_t^{-1}$$

6.2 PF time update (prediction): for i=1, 2, … N, predict new particles:

$$x^{n,(i)}_{t+1|t} \sim p\left(x^{n,(i)}_{t+1|t} \mid X^{n,(i)}_t, Y_t\right)$$

6.3 KF time update:

$$\hat{x}^l_{t+1|t} = \bar{A}^l_t \hat{x}^l_{t|t} + G^l_t (Q^{ln}_t)^T (G^n_t Q^n_t)^{-1} z_t + f^l_t + L_t (z_t - A^n_t \hat{x}^l_{t|t})$$
$$P_{t+1|t} = \bar{A}^l_t P_{t|t} (\bar{A}^l_t)^T + G^l_t \bar{Q}^l_t (G^l_t)^T - L_t N_t L^T_t$$
$$N_t = A^n_t P_{t|t} (A^n_t)^T + G^n_t Q^n_t (G^n_t)^T$$
$$L_t = \bar{A}^l_t P_{t|t} (A^n_t)^T N^{-1}_t$$
where
$$z_t = x^n_{t+1} - f^n_t$$
$$\bar{A}^l_t = A^l_t - G^l_t (Q^{ln}_t)^T (G^n_t Q^n_t)^{-1} A^n_t$$
$$\bar{Q}^l_t = Q^l_t - (Q^{ln}_t)^T (Q^n_t)^{-1} Q^{ln}_t$$

7. Set time, $t = t + 1$ and repeat from step 2.

Note: If the cross-covariance Q^{ln}_t between the two noise sources v^n_t and v^l_t is zero , then $\bar{A}^l_t = A^l_t$ and $\bar{Q}^l_t = Q^l_t$. Variance threshold value and improvement factor are the two parameters that are chosen based on the application and maximum amount of noise that can affect the measurement in the system. Improvement factor of 2 indicates 50% reduction in the number of particles which is taken as a limiting case in the simulation as further reduction in the number of particles increases the chance of filter getting diverged. Threshold value determines the quality of the measurement data which is used to determine whether there is a scope for improvement or not. Typical value of variance threshold value used in simulation is 0.04.

IV. IMPORTANT MODEL CLASS

The model given by (2) is the general model used in state estimation problems. Sometimes there will be a situation when models will have state equations which are linear and a nonlinear measurement equation. This is a special case of the general model which is very important as it is very common in several state estimation problems such as positioning, target tracking and collision avoidance. Important special case model [12] is given below:

$$\boldsymbol{x}^n_{t+1} = \boldsymbol{A}^n_{n,t}\, \boldsymbol{x}^n_t + \boldsymbol{A}^n_{l,t} \boldsymbol{x}^l_t + \boldsymbol{G}^n_t \boldsymbol{v}^n_t \qquad (7a)$$

$$\boldsymbol{x}^l_{t+1} = \boldsymbol{A}^l_{n,t} \boldsymbol{x}^n_t + \boldsymbol{A}^l_{l,t} \boldsymbol{x}^l_t + \boldsymbol{G}^l_t \boldsymbol{v}^l_t \qquad (7b)$$

$$\boldsymbol{y}_t = \boldsymbol{h}_t(\boldsymbol{x}^n_t) + \qquad + \boldsymbol{e}_t \qquad (7c)$$

where $\boldsymbol{v}^n_t \sim N(0, \boldsymbol{Q}^n_t)$ and $\boldsymbol{v}^l_t \sim N(0, \boldsymbol{Q}^l_t)$ represents the process noise, $\boldsymbol{Q}^n_t, \boldsymbol{Q}^l_t$ are the covariance of nonlinear and linear component of process noise, $\boldsymbol{A}^n_{l,t}$, $\boldsymbol{A}^l_{l,t}$, $\boldsymbol{A}^n_{n,t}$, $\boldsymbol{A}^l_{n,t}$, denotes constant matrices related to linear and nonlinear variables. $\boldsymbol{e}_t$ is assumed to be known with arbitrary distribution. The measurement equation (7c) given in the above model does not have any information regarding the linear state variable $\boldsymbol{x}^l_t$ .The detailed description about the important special model is given in [12].

V. TYPICAL TARGET TRACKING EXAMPLE

A typical target tracking example [19-21] is considered in this section in order to check the performance of AMPF. In this example, a 2D model having constant acceleration is used to estimate the position and velocity of an aircraft. Here it is assumed that the height of the aircraft is constant i.e., a level flight is considered. Also it is assumed that the range and bearing angle are the two measurements applied to the filter. It can be seen from the dynamic state space model of the aircraft that, the model consists of linear state equations and nonlinear measurement equation.

The dynamic state space model of the target tracking example is given below; the height component is discarded as a level flight is considered.

$$\boldsymbol{x}_{t+1} = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_t + \boldsymbol{v}_t \qquad (8a)$$

$$\boldsymbol{y}_t = \begin{pmatrix} r \\ \phi \end{pmatrix} = \begin{pmatrix} \sqrt{p_x^2 + p_y^2} \\ \arctan\left(\frac{p_y}{p_x}\right) \end{pmatrix} + \boldsymbol{e}_t \qquad (8b)$$

where $\boldsymbol{x}_t = (p_x, p_y, v_x, v_y, a_x, a_y)^T$ i.e. position, velocity, acceleration is the state vector, r is the range and θ is the bearing angle. For simulation purpose the sampling time is assumed to be 1 sec, i.e. T=1 sec. Here $\boldsymbol{e}_t, \boldsymbol{v}_t$ represents measurement noise and process noise and are assumed to be Gaussian having zero mean and covariance

$$\boldsymbol{R} = \operatorname{cov} \boldsymbol{e} = diag(100, 0.01) \qquad (9a)$$

and

$$\boldsymbol{Q}^{\boldsymbol{n}} = cov\left(\boldsymbol{v}^{\boldsymbol{n}}\right) = diag(100,100) \quad (9b)$$

$$\boldsymbol{Q}^{\boldsymbol{l}} = cov\left(\boldsymbol{v}^{\boldsymbol{l}}\right) = diag(2,3,0.09,0.09) \quad (9c)$$

respectively.

Here the two position states $[p_x, p_y]$ are nonlinear and the remaining states $[v_x, v_y, a_x, a_y]$ are linear. Therefore, by marginalizing out the linear state variables,

$$\boldsymbol{x}_t^{\boldsymbol{n}} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \boldsymbol{x}_t^{\boldsymbol{l}} = \begin{bmatrix} v_x \\ v_y \\ a_x \\ a_y \end{bmatrix} \quad (10)$$

The model given by (8) is similar to the important model mentioned in section 3 with the terms $\boldsymbol{A}_{n,t}^{l}\boldsymbol{x}_t^{n}$ zero. Comparing with the model,

$$\boldsymbol{G}_t^{n} = \boldsymbol{I}_{2X2}; \boldsymbol{G}_t^{l} = \boldsymbol{I}_{4X4}$$

$$\boldsymbol{A}_{n,t}^{n} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \boldsymbol{A}_{l,t}^{n} = \begin{pmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.5 \end{pmatrix}$$

$$\boldsymbol{A}_{n,t}^{l} = \boldsymbol{0}_{4X2}; \boldsymbol{A}_{l,t}^{l} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (11)$$

AMPF given in section 3 is applied to this target tracking model.

VI. Simulation Results

In order to vividly comprehend the benefits of AMPF when compared to that of MPF let us consider an aircraft target tracking example whose state space model is given by (8). A stable estimate of the state variables is obtained using Monte Carlo (MC) simulations. With an aim to compare the performance of AMPF the same aircraft target tracking example is applied to both MPF and PF, and simultaneously the performance of all the three filters are compared. For simulation purpose, an aircraft trajectory (assuming level flight) and corresponding bearing angle and range observations have been generated using (8a) and (8b) for 600 samples of time. Table 1 gives the parameters used for generating the model, trajectory unless specified.

Two types of aircraft trajectories are generated using the model, one in which aircraft follows a straight line path and other in which the aircraft is performing some Maneuvering. AMPF, MPF and PF are applied to both aircraft trajectory and its performance is analyzed.

The performance of the filters is compared using Root mean square error (RMSE), which is given by

$$\left(\frac{1}{\mathrm{N}}\sum_{t=1}^{N}(1/N_{MC})\sum_{j=1}^{N_{MC}}\left\|\boldsymbol{x}_t^{true} - \hat{\boldsymbol{x}}_t^{(j)}\right\|_2^2\right)^{1/2} \quad (12)$$

TABLE I. Parameter Values

Parameter	Values
Number of Monte Carlo Simulations	200
Initial Position $[p_x, p_y]$ in m	[-1000*10,1000*5]
Initial Velocity $[v_x, v_y]$ (in m/s)	56
Acceleration $[a_x, a_y]$ (in m/s^2)	0.4
Initial state covariance $\boldsymbol{P_o}$	diag(0.01,0.01,0.01,0.01 ,0.01,0.01)
Measurement Noise Covariance $\boldsymbol{R}$	diag(100,0.01)
Process Noise Covariance $\boldsymbol{Q}^{\boldsymbol{n}}$ (nonlinear)	diag(200,200)
Process Noise Covariance $\boldsymbol{Q}^{\boldsymbol{l}}$ (linear)	diag(2,3,0.2,0.2)

where $\boldsymbol{x}_t^{true}$ represents the true location at time t, $\hat{\boldsymbol{x}}_t^{(j)}$ gives the estimated location at time t in the j^{th} simulation, N is the number of samples and N_{MC} represents the number of MC simulations used. MC simulations are used to obtain a stable estimate. To get a clear picture about the performance of the filter two types of analyses are done.

1. Execution Time Analysis.
2. Performance Analysis.

From the above two analyses, it can be concluded that AMPF being an advanced version of MPF is better than MPF and PF in terms of speed of execution. This improvement in the execution speed is achieved without much affecting the filter performance mainly due to the noise tolerance property of MPF [20,21] which is the basic idea behind AMPF. Execution time and performance of the filter will be evaluated in terms of time improvement factor and performance index respectively. The results are obtained using 1000 (Non-Maneuvering) & 5000 (Maneuvering) particles for 600-time samples. In the simulations execution time is mentioned in seconds (s), RMSE of position in meters (m) and RMSE of Velocities in m/s .Fig. 1(a) and Fig. 1(b) represents the non - Maneuvering trajectory and Maneuvering trajectory where the aircraft is assumed initially at coordinates [-1000*10, 1000*5] traveling with constant acceleration of 0.4 m/s^2 .

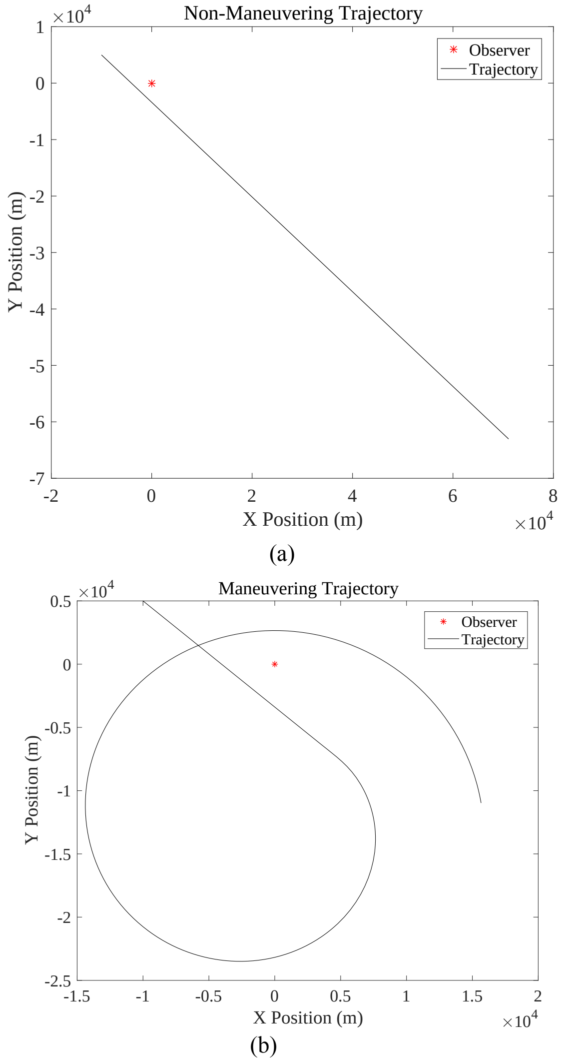


Figure 1. Different target trajectories, black line (solid line), observer at origin shown as '*'. (a) True Non Maneuvering Target Trajectory, (b) True Maneuvering Target Trajectory.

A. Execution Time Analysis of AMPF

The main aim of AMPF is to reduce the execution time when compared to existing filters. From this section, it will be concluded that AMPF is able to find the estimate much faster than other filters without much affecting the performance of the filter.

A1. Execution Time of AMPF with respect to different MC simulations

The execution time of AMPF is compared with that of MPF and PF in the case of both Non-Maneuvering & Maneuvering Trajectory. Here the execution time is plotted for different Monte Carlo simulations and from the simulation, it will be clear that AMPF is able to find the estimates much faster than MPF and PF.

In Fig. 2, the execution time of AMPF (blue line / line with diamond shape) is much less when compared to that of MPF (red line / line with square shape) and that of PF (black line / line with circle shape) which indicates that AMPF is able to find the estimates much faster when compared to other filters. Similar results are obtained in the case of both Non-Maneuvering and Maneuvering Trajectory. The average execution time of AMPF, MPF & PF for 600-time samples are given below in table II & III.

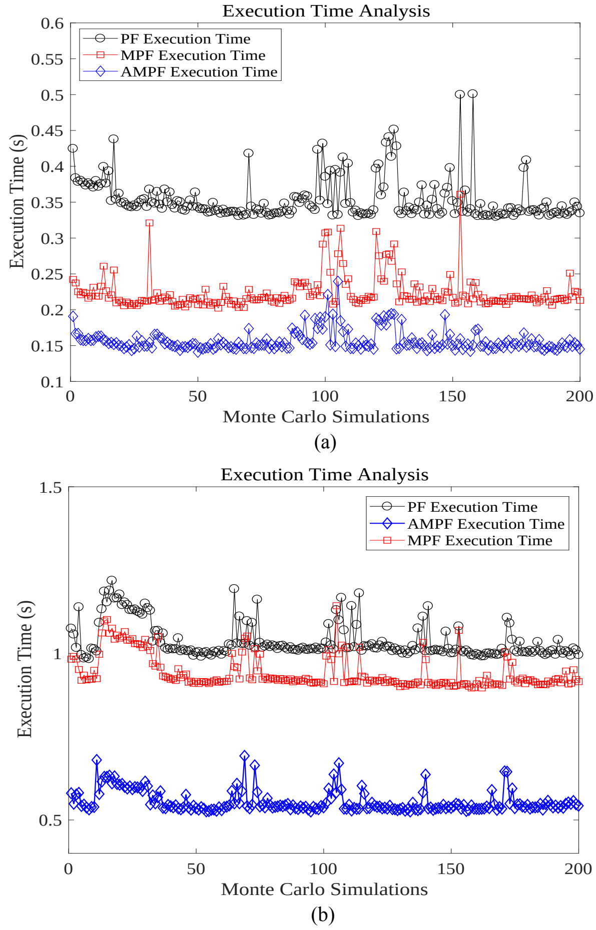


Figure 2. Execution time of AMPF (blue / diamond shape), MPF (Red / Square shape) & PF (Black / Circle shape) for different MC Simulations. (a) Execution time of filters for different MC Simulations in the case of Non- Maneuvering Trajectory, (b) Execution time of filters for different MC Simulations in the case of Maneuvering Trajectory.

From Table II & III it is clear that the execution time of AMPF is less compared to that of MPF and PF without affecting the performance of the filter. Here when PF took 0.3514 (Non-Maneuvering), 1.1239 (Maneuvering) sec to complete estimation, MPF took 0.2222, 1.0361 sec and the proposed method was able to estimate the parameter much faster in 0.1567 sec in the case of Non-Maneuvering and 0.5999 sec in the case of Maneuvering trajectory thereby introducing a considerable reduction in the execution time. This reduction in the execution is achieved without affecting the performance of the filter much.

TABLE II. SIMULATION RESULT OF PF, MPF, AMPF IN THE CASE OF NON-MANEUVERING TRAJECTORY WITH 1000 PARTICLES

Parameter	**PF**	**MPF**	**AMPF**
Execution Time	0.3514	0.2222	0.1567
RMSE X Position	25.9988	23.3155	24.1375
RMSE Y Position	27.5476	25.8006	26.0678
RMSE X Velocity	3.7726	3.0209	3.2242
RMSE Y Velocity	3.9646	3.1396	3.4406

TABLE III. SIMULATION RESULT OF PF, MPF, AMPF IN THE CASE OF MANEUVERING TRAJECTORY WITH 5000 PARTICLES

Parameter	**PF**	**MPF**	**AMPF**
Execution Time	1.1239	1.0361	0.5999
RMSE X Position	13.4459	13.2284	13.3823
RMSE Y Position	10.6721	10.4522	10.6264
RMSE X Velocity	3.4284	3.3178	3.4078
RMSE Y Velocity	3.2449	3.1391	3.2080

A2. Comparison of Execution Time of AMPF in terms of Time Improvement factor.

A Little more insight about the execution time of AMPF is obtained by introducing a new parameter known as Time Improvement Factor. Time Improvement Factor (TIF) is defined as the ratio of the difference between execution time of reference filter and AMPF to the execution time of reference filter. Reference filter can be either PF or MPF. The value of TIF depends mainly on two factors. First, it depends on the improvement factor of AMPF, and secondly, the variation of the amount of noise present in the measurements.

$$TIF = 1 - \frac{ExecutionTime_{AMPF}}{ExecutionTime_{ReferenceFilter}}$$

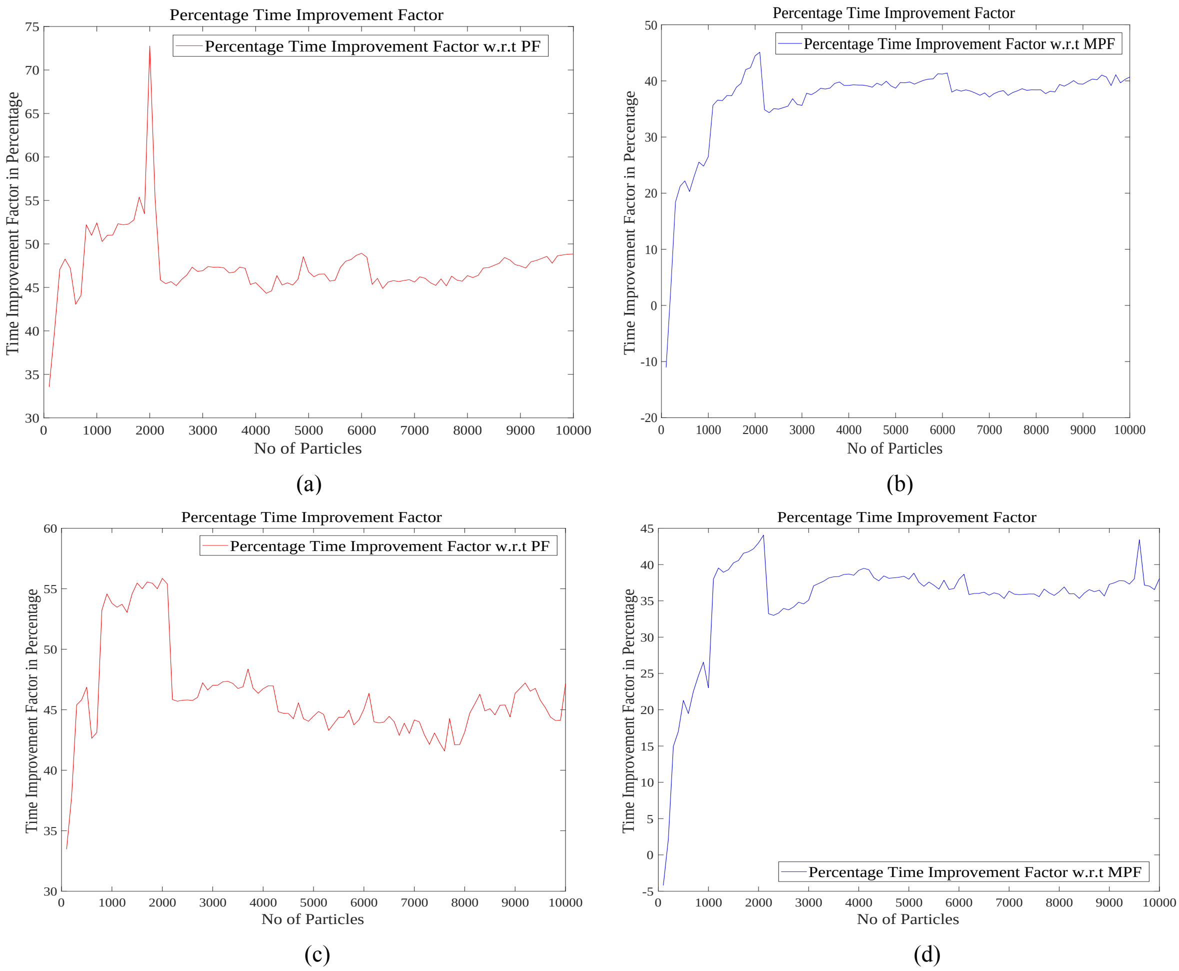


Figure 3. Percentage Time Improvement factor in terms of number of particles. (a) Percentage TIF with respect to PF as reference filter in the case of Non-Maneuvering Trajectory, (b) Percentage TIF with respect to MPF as reference filter in the case of Non-Maneuvering Trajectory, (c) Percentage TIF with respect to PF as reference filter in the case of Maneuvering Trajectory, (d) Percentage TIF with respect to MPF as reference filter in the case of Maneuvering Trajectory.

DOI 10.5013/IJSSST.a.19.06.54 ISSN: 1473-804x online, 1473-8031 print

Time Improvement factor of 0 indicates that execution time of AMPF is same as that of Reference filter while Time improvement factor closer to 1, which is desirable indicates that AMPF is faster than Reference filter. Higher the Time Improvement Factor better the filter performance. Time improvement factor can also be expressed in percentage as given below.

$$TIF = \left(1 - \frac{ExecutionTime_{AMPF}}{ExecutionTime_{ReferenceFilter}}\right) X \ 100 \ \%$$

TABLE IV. COMPARISON OF PERFORMANCE OF AMPF WITH RESPECT TO PF AND MPF IN TERMS OF PERCENTAGE TIF.

Parameter	Non-Maneuvering Trajectory		Maneuvering Trajectory	
	PF	MPF	PF	MPF
TIF	55	28	49	41

Percentage Time Improvement factor indicates how much faster the proposed filter is when compared to the reference filter. From Table IV, Percentage Time improvement factor obtained with PF as reference filter is 55% in the case of Non-Maneuvering Trajectory which indicates that the proposed filter is 55% faster than PF. In the same situation with MPF as reference filter Percentage Time Improvement factor is 28%, which indicates that proposed filter is 28% faster than MPF. A similar result is obtained in the case of Maneuvering Trajectory. Thus it is clear that proposed filter is 25 – 55 % faster than the existing filter for the typical target tracking application.

The effect of total number of particles on Time improvement factor is analyzed in Fig.3. It can be seen from simulation that, as the total number of particles of the system is increased, the time improvement factor also increases and reaches a maximum value which is determined by the Improvement factor given in step 3 of AMPF algorithm given in section 2.

From Fig. 3, it can be concluded that if the number of particles of the system is more, then the time improvement factor of adaptive marginalized particles filter is more thereby resulting in considerable improvement in the execution time when compared to other filters.

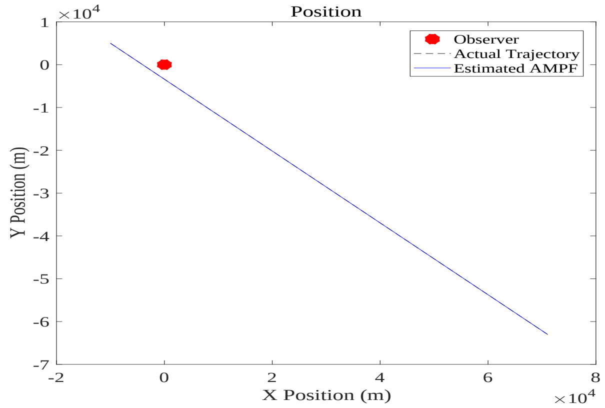


Figure 4. Tracking of Non- Maneuvering trajectory by AMPF (Blue / Solid line).

B. Performance Analysis of Adaptive Marginalized Particle Filter.

In section VI.A, it was concluded that AMPF was able to perform estimation much faster than existing filters. Here the performance of the AMPF is compared with that of MPF and PF. The analysis will be done in the case of both Non-Maneuvering Trajectory and Maneuvering Trajectory as shown in Fig. 1. The range and the bearing angle corresponding to the trajectory are calculated and both are applied as measurement inputs to all the three filters and their performance are compared in terms of execution time and RMSE value of X, Y position and velocity.

B1. Performance Analysis of AMPF in the case of Non-Maneuvering Trajectory.

The Non-Maneuvering Trajectory is given in Fig. 1 (a) shows an aircraft traveling along a straight line which is shown by black line (solid line) and the observer is assumed to be at the origin and is represented by a star symbol (red star). The calculated range and bearing angle are applied as an input to the filter.

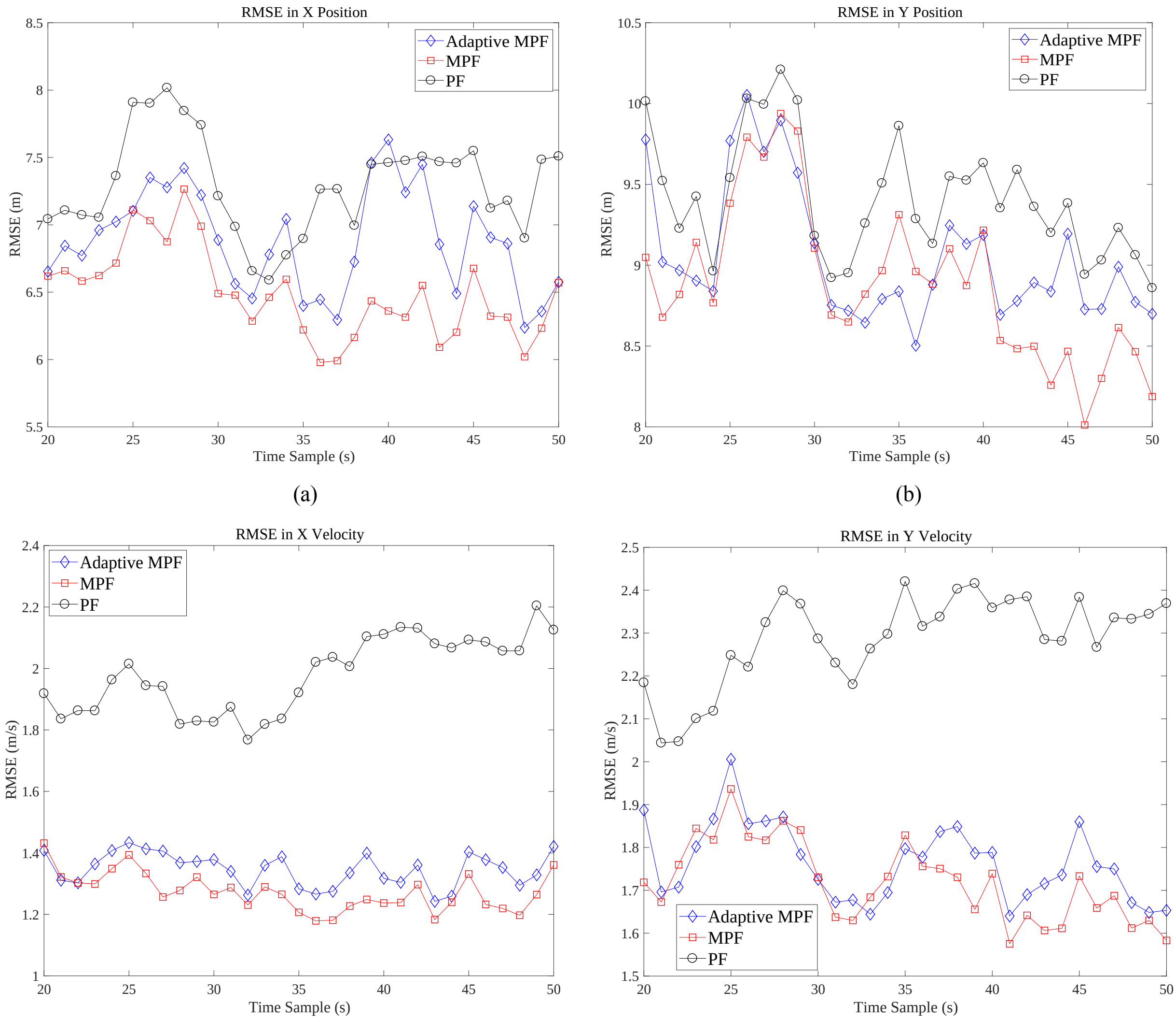
The comparison between the performance of AMPF, MPF and PF is given in Table 5. The simulation is done with 1000 particles and 200 Monte Carlo simulations are done.

TABLE V. SIMULATION RESULT OF PF, MPF, AMPF IN THE CASE OF NON-MANEUVERING TRAJECTORY WITH 1000 PARTICLES

Parameter	PF	MPF	AMPF
Execution Time	0.3380	0.2127	0.1515
RMSE X Position	25.3804	23.5876	24.6464
RMSE Y Position	27.4016	25.5923	26.6510
RMSE X Velocity	3.7457	2.9731	3.2763
RMSE Y Velocity	3.9628	3.1920	3.5124

DOI 10.5013/IJSSST.a.19.06.54

ISSN: 1473-804x online, 1473-8031 print



(c) (d)

Figure 5. RMSE in X, Y position and velocity coordinates in the case of Non-Maneuvering Trajectory between 20th and 50th time samples of AMPF (blue / diamond shape), MPF (Red / Square shape) & PF (Black / Circle shape). (a) RMSE in X position coordinate, (b) RMSE in Y position coordinate, (c) RMSE in X velocity coordinate, (d) RMSE in Y velocity coordinate..

From table V, it is identified that the AMPF is able to perform the estimation process much faster than the MPF and PF and also RMSE of AMPF is comparable with that MPF and PF. Fig 4 given below shows the performance of AMPF for 600-time samples. The figure shows the plot between X position and Y position. Black line (dashed line) shows the actual trajectory, Blue line (solid line) shows the performance of AMPF. From the figure it is clear that AMPF is able to track the trajectory. Black line (dashed line) which shows the actual trajectory is not visible in Fig 4 as AMPF perfectly tracks the actual trajectory. In order to get a clear picture about the filter performance, the RMSE of X and Y Position with respect to time samples is plotted in the case of three filters.

From Fig. 5, it is evident that there is not much increase in root mean square error introduced in X, Y position and velocity parameters. In the figure blue line (line with diamond shape) denotes AMPF, red line (line with square shape) indicates MPF and PF is represented by black line (line with circle shape). Thus it can be concluded from figures and tables that AMPF is able to track the trajectory without much increase in RMSE at the same time AMPF is able to track much faster than MPF and PF.

B2. Performance Analysis of AMPF in the case of Maneuvering Trajectory.

In order get a vivid picture about the performance of AMPF, the filter is applied in a scenario where an aircraft is assumed to make a number of maneuverings which is more relevant than Non-Maneuvering trajectory as in most of the practical situations, aircrafts will be going through such movements. Such a Maneuvering trajectory is given in Fig. 1 (b). Similar to the Non- Maneuvering case the range and bearing angle measurements are calculated and applied to these filters.

Table VI shows the performance of the filter in the case of Maneuvering Trajectory with 5000 particles and from the table, it can be seen that similar to the Non-Maneuvering case AMPF is able to perform the estimation process much faster than other two filters without causing much increase in the RMSE value.

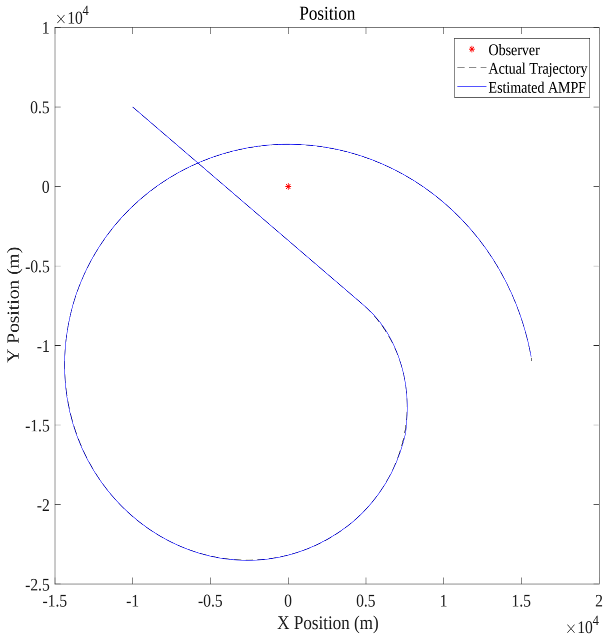


Figure 6. Tracking of Maneuvering trajectory by AMPF (Blue / Solid line).

TABLE VI. SIMULATION RESULT OF PF, MPF, AMPF IN THE CASE OF MANEUVERING TRAJECTORY WITH 5000 PARTICLES

Parameter	PF	MPF	AMPF
Execution Time	1.0481	0.9528	0.5546
RMSE X Position	13.4770	13.4309	13.4027
RMSE Y Position	10.6557	10.4314	10.5649
RMSE X Velocity	3.4289	3.3150	3.4248
RMSE Y Velocity	3.2468	3.1335	3.1898

Fig 6. Shows the tracking of an aircraft in the case of Maneuvering trajectory by AMPF. In the figure, the actual trajectory is shown as a black line (dashed line) and the path tracked by AMPF is shown by blue line (solid line). It is clear from the figure that AMPF is able to track the trajectory.

Fig. 7, shows the RMSE in X, Y position and velocity parameter in the case of Maneuvering trajectory. From the figure, it is can be concluded that the RMSE in the case of AMPF is not increased much which is similar to the result obtained in the case of Non-Maneuvering Trajectory. Here also it can be seen that the filter is able to find the estimates much faster than the existing filters.

B3. Comparison of performance of AMPF in terms of Performance index.

In order to get more insight into the performance of AMPF, a new parameter known as performance index is defined. Performance index (PI) is defined as the ratio of the difference between RMSE of reference filter and that of AMPF to the RMSE of reference filter (PF / MPF).

DOI 10.5013/IJSSST.a.19.06.54

ISSN: 1473-804x online, 1473-8031 print

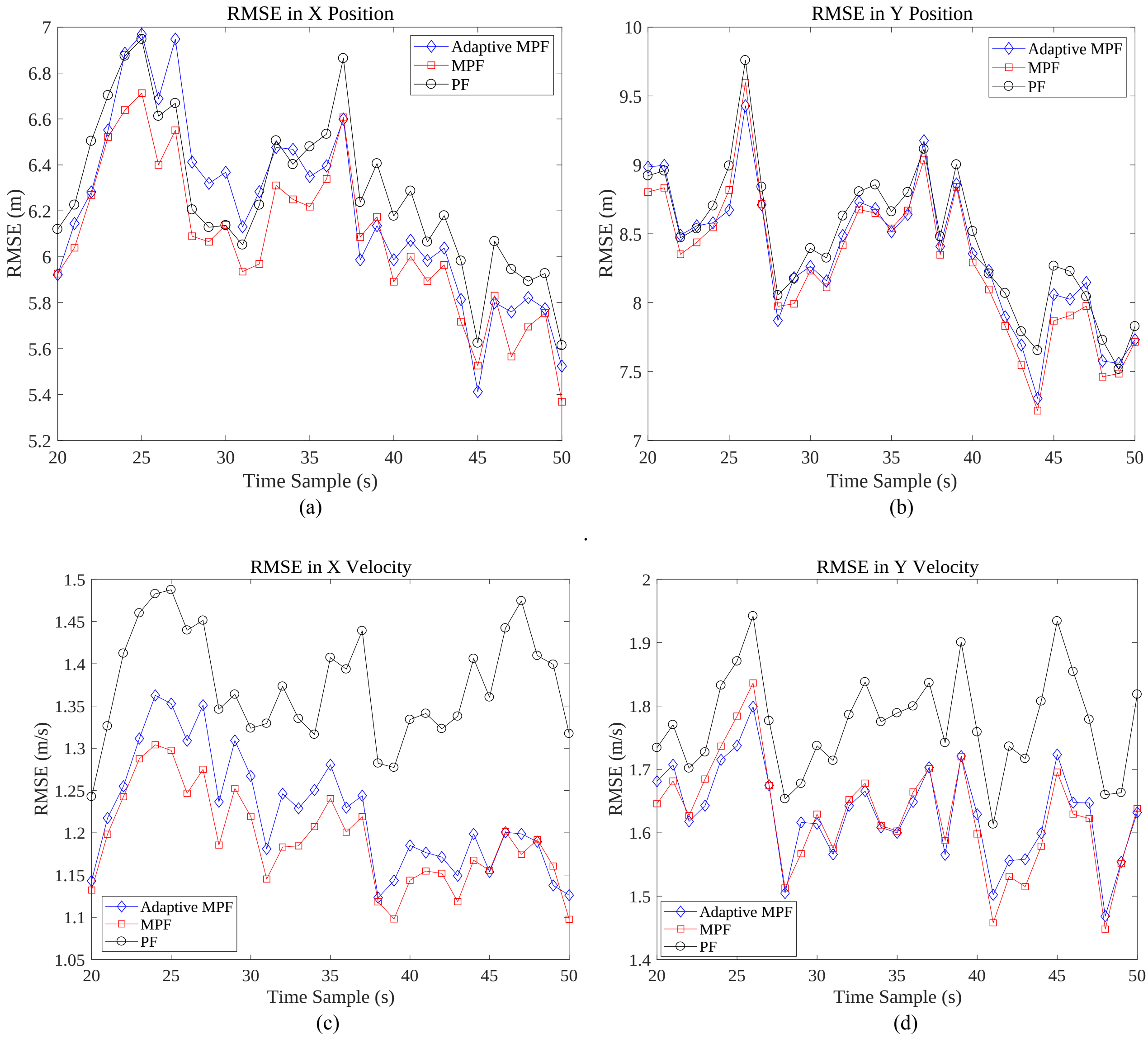


Figure 7. Root mean square error in X, Y position and velocity coordinates in the case of Maneuvering Trajectory between 20th and 50th time samples of AMPF (blue / diamond shape), MPF (Red / Square shape) & PF (Black / Circle shape). (a) RMSE in X position coordinate, (b) RMSE in Y position coordinate, (c) RMSE in X velocity coordinate, (d) RMSE in Y velocity coordinate.

$$PI = 1 - \frac{RMSE_{AMPF}}{RMSE_{\text{ReferenceFilter}}}$$

A value of performance index ‘0’ indicates that both AMPF and reference filter have the same amount of RMSE and a value ‘1’ indicates that RMSE of AMPF is zero. A negative value of performance index indicates that RMSE of AMPF is greater than that of reference filter. The real advantage of using PI is that it clearly shows how much close the performance of AMPF is with respect to the reference filter.

By calculating the Performance index, a clear picture about the performance of AMPF is obtained compared to existing filters. Performance index with respect to PF and MPF is simulated in the case of both Non-Maneuvering Trajectory and Maneuvering Trajectory.

Table 7 show the Performance Index with respect to PF & MPF of X Position Coordinate and Y Position Coordinate. From table 7, it can be noted that Performance Index with respect to PF of X position is 0.06335 and with MPF is -0.05268 in the case of Non-Maneuvering Trajectory. A negative Performance Index with respect to MPF indicates that the RMSE of AMPF is larger than that of the RMSE introduced in the case of MPF.

TABLE VII. COMPARISON OF PERFORMANCE OF AMPF WITH RESPECT TO PF AND MPF IN TERMS OF PI..

Parameter	Non-Maneuvering Trajectory		Maneuvering Trajectory	
	PF	MPF	PF	MPF
PI (X Position)	0.0633	-0.05268	0.360	-1.09
PI (Y Position)	0.0507	-0.04591	0.386	-1.495

Also, a positive PI with respect to PF indicates that RMSE of AMPF is lesser than that of PF. Similar results can be noticed in the case of PI of Y position. In the case of Maneuvering trajectory also, a similar behaviour is obtained. It can be also interpreted that the RMSE of AMPF lies in between that of MPF and PF. Thus using performance index, it can be concluded that the error introduced in AMPF, due to the reduction in the number of particles is only slightly more than the existing MPF. This result is similar to the one obtained in section 5.2.1 and 5.2.2.

Execution Time Analysis and Performance Analysis are performed to evaluate the performance of the proposed Adaptive Marginalized Particle Filter. These analyses are done in the case of both Non-Maneuvering and Maneuvering trajectory. Filter performance is also expressed in terms of performance index and time improvement factor. From the simulation, it can be concluded that the proposed filter is able to find the estimates much faster than existing filters without affecting the performance of the filter much. The amount of improvement in the execution speed depends on the amount of noise present in the measurement at different instant of time samples. This is in line with the expected result as proposed in the case of AMPF. Thus the proposed AMPF is better than MPF and PF in terms of speed of execution which find immense potential in applications like missile tracking, target tracking where execution speed is an important criterion.

VII. CONCLUSION

In this paper, a new filter known as Adaptive Marginalized Particle Filter is proposed. Various simulation scenarios are considered in order to check the performance of the proposed filter. The performance of AMPF is compared with that of existing filters like, MPF and the general PF. From the simulation, it is clear that the proposed filter is able to find the estimates much faster than the other two filters. Also, it was noted that this improvement in execution time is obtained without much affecting the performance of the filter. Thus it can be concluded that Adaptive Marginalized Particle Filter has a faster mode of operations than Marginalized Particle Filter and Particle Filter which is very useful in missile tracking, aircraft tracking, collision avoidance etc.

ACKNOWLEDGMENT

The authors would express their gratitude towards Editor in Chief and reviewers of this journal for their constructive feedback on the paper. The authors would also like to thank Noorul Islam Centre for higher education for providing the facilities.

REFERENCES

[1] X. Rong Li and V. P. Jilkov, "Survey of maneuvering targettracking . part I: dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[2] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part II: Motion Models of Ballistic and Space Targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 1, pp. 96–119, Jan. 2010.

[3] M. H. Hayes, "Optimum Filters," in *Statistical Digital Signal Processing and Modeling*, 1st ed., S.Elliot, Ed. John Wiley & Sons, INC, 1996, p. 608.

[4] P. Kalane, "Target Tracking Using Kalman Filter," *Int. J. Sci. Technol.*, vol. 2, no. 2, pp. 16–24, 2012.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[6] F. Gustafsson, "Particle Filter Theory and Practice with Positioning Applications," *IEEE A E Syst. Mag.*, vol. 25, no. 7, pp. 53–81, 2010.

[7] J. H. Kotecha and P. M. Djurić, "Gaussian sum particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2602–2612, 2003.

[8] M. Talha and R. Stolkin, "Particle filter tracking of camouflaged targets by adaptive fusion of thermal and visible spectra camera data," *IEEE Sens. J.*, vol. 14, no. 1, pp. 159–166, 2014.

[9] M. Bolić, P. M. Djuric, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP J. Appl. Signal Processing*, vol. 15, pp. 2267–2277, 2004.

[10] X. Fu and Y. Jia, "An improvement on resampling algorithm of particle filters," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5414–5420, 2010.

[11] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized Particle Filters for Mixed Linear / Nonlinear State-Space Models," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2279–2289, 2005.

[12] A. Murangira and C. M. K. Dahia, "A Mixture Regularized Rao-Blackwellized Particle Filter for Terrain Positioning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1967–1985, 2016.

[13] H. D. Hesar and M. Mohebbi, "ECG Denoising Using Marginalized Particle Extended Kalman Filter with an Automatic Particle Weighting Strategy," *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 3, pp. 635–644, 2017.

[14] R. Karlsson and F. Gustafsson, "Recursive Bayesian estimation: bearings-only applications," *IEE Proc.-Radar Sonar Navig*, vol. 152, no. 5, pp. 305–313, 2005.

[15] Z. Ronghui, W. Ling, W. Jianwei, and S. Zhongkang, "Passive target tracking using marginalized particle filter," *J. Syst. Eng. Electron.*, vol. 18, no. 3, pp. 503–508, 2007.

[16] Jinwhan Kim, S. S. Vaddi, P. K. Menon, and E. J. Ohlmeyer, "Comparison Between Nonlinear Filtering Techniques for Spiraling Ballistic Missile State Estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 313–328, 2012.

[17] U. Orguner and F. Gustafsson, "Target tracking with particle filters under signal propagation delays," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2485–2495, 2011.

[18] S. Saha and F. Gustafsson, "Particle filtering with dependent noise processes," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4497–4508, 2012.

[19] R. Karlsson, T. Schön, and F. Gustafsson, "Complexity Analysis of the Marginalized Particle Filter," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4408–4411, 2005.

[20] J. Sanil and M. Mathurakani, “Noise tolerance analysis of marginalized particle filter for target tracking,” in *International Conference on Microelectronics, Communication and Renewable Energy (ICMiCR-2013)*, 2013.

[21] J. Sanil, N. Santhi, and K. Ramar, “Measurement Noise Analysis of Marginalized Particle Filter for Target Tracking Applications,” *Int. J. Simul. Syst. Sci. Technol.*, vol. 19, no. 5, p. 32.1-32.14, 2018.