

A Hybrid PSO–GA Based Clustered Sorted Weighting Lookahead Particle Filter For Track-Before-Detect Application

Ramakrishna Gurajala^{1*}, Praveen B. Choppala²

^{1*}Research Scholar, Department. of E.C.E., AU college of Engineering, Andhra University, Visakhapatnam, India

²Adhoc Professor, Department. of E.C.E., AU college of Engineering, Andhra University. Visakhapatnam, India

Abstract— In the Track Before Detection (TBD) approach, a target is monitored in an environment with a low signal-to-noise ratio before it is officially identified as a target. TBD is used in various areas such as air traffic control, maritime monitoring, tracking of underwater autonomous vehicles (UAVs), and detecting small objects in remote sensing. A key technique used in TBD is the particle filter (PF). In situations that are nonlinear and non-Gaussian, the PF is a leading method for detecting targets. However, the conventional Particle Filter algorithm suffers from a loss of particle diversity due to degradation and resampling processes, which prevents the particle set from accurately representing the true state probability density function. The Particle Swarm Optimization (PSO) algorithm can help reduce particle degradation in PF, but its effectiveness is heavily influenced by the variance of measurement noise and it is prone to getting stuck in local optima, limiting its filtering precision. To address these issues, a hybrid method combining the Genetic Algorithm (GA) with PSO is proposed. By combining PSO's fast convergence with GA's strong global search abilities, this approach improves particle diversity, preserves the effectiveness of high-quality particles, and enhances overall accuracy. GA employs global search operations such as selection, crossover, and mutation. In this paper, we propose a selection method called the clustered sorted weighting lookahead particle filter. This algorithm divides the particle set into several clusters using the k-means clustering algorithm. Within each cluster, particles located in high-importance regions are identified and replicated to replace those in low-importance regions. After this replacement, the clusters are merged using cluster weights. This approach integrates clustering with selective likelihood-based resampling to preserve multiple posterior modes, maintain particle diversity, and enhance root mean square error. The tracking performance of the proposed method is assessed using two nonlinear models.

Keywords— Particle filter, K-means Clustering, Track before detection (TBD), RMSE.

1. INTRODUCTION

One signal processing method for tracking and detecting faint targets is the track-before-detection (TBD) approach. TBD is applied in sonar, radar, and image tracking systems. It processes raw sensor data by accumulating information over multiple frames to track the target. The primary techniques used in TBD include dynamic programming, particle filtering, and velocity filtering. Particle filters are particularly employed for tracking targets in nonlinear and non-Gaussian environments.

The Bayesian state estimator is commonly used to estimate nonlinear and non-Gaussian models. One type of Bayesian state estimator is the particle filter (PF). The PF performs recursive estimation by generating a set of weighted particles. These particles are used to approximate the posterior distribution of the state space. The PF carries out this procedure in two steps. First, it propagates the particles through a proposal distribution and updates their weights. If the proposal distribution is poorly chosen, a single particle may dominate the weight distribution. To address this issue, the second step involves a resampling process, where particles with high weights replace those with low weights. This resampling significantly enhances the accuracy of the posterior approximation.

In the literature, various resampling techniques have been proposed. One such stochastic resampling method [1] is multinomial resampling. This approach involves selecting particles randomly based on a uniform distribution, where the sum of the chosen particles corresponds to the cumulative sum of the particles' normalized weights. However, this method tends to have high variance due to sampling over the interval (0,1). To reduce the variance among particles, methods like systematic resampling and stratified resampling are used.

The residual resampling technique enhances performance by first deterministically assigning copies of particles based on the integer portion of their weights, and then randomly sampling the remaining fractional part. Additionally, various (partially) deterministic resampling methods have been suggested. For example, Bolić et al. [8] proposed threshold-based methods that deterministically duplicate particles with high weights, while particles with lower weights are resampled in a stochastic manner.

The Metropolis resampler is a parallel resampling method used in particle filters. It enables parallel implementation of particle filtering by minimizing communication between particles. However, this benefit comes at the cost of a slight decrease in particle accuracy. Combining Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) with the Particle Filter significantly improves the performance of particle filtering. PSO enhances the filter's ability to exploit by guiding particles toward areas in the state space with high probability, leading to quicker convergence. Meanwhile, GA contributes strong exploration capabilities through genetic operations like selection, crossover, and mutation, which help maintain particle diversity and prevent premature convergence. As a result, the hybrid PSO-GA approach effectively balances global exploration and local exploitation, improving estimation accuracy and robustness in nonlinear and non-Gaussian systems. Additionally, the optimized particle distribution reduces particle degeneracy, allowing the filter to achieve reliable tracking performance with fewer particles.

In this paper, we introduce a Clustered Sorted Weighting Lookahead Particle Filter (CSLAPF) as a selection mechanism within Genetic Algorithms (GA). The CSLAPF operates with high precision. The algorithm divides the set of particles into clusters. Within each cluster, the lookahead set is sorted, and a few particles with large weights are chosen by comparing their weights against a threshold. These high-weight particles replace those with low weights. This replacement strategy eliminates the need for resampling particles. After replication within each cluster, the clusters are merged, and the weights are normalized. Although clustering results in a slight increase in computational cost compared to standard particle filters, it significantly enhances estimation accuracy, convergence rate in multimodal situations.

The rest of the paper is organized as follows: Section II introduces the PF notations. In Section III, we propose a hybrid PSO-GA based clustered sorted weighting lookahead PF. Section IV presents the evaluation results for the two nonlinear methods. Finally, Section V provides the conclusion.

II. PARTICLE FILTERING

The particle filter is a Bayesian estimator that tracks hidden dynamic targets in nonlinear and non-Gaussian environments by utilizing a collection of particles with associated discrete weights. It is composed of two models: the state transition model and the observation model. These two models are regarded as

$$\begin{aligned} x_t/x_{t-1} &\sim p(x_t/x_{t-1}, \theta) \\ y_t/x_{t-1} &\sim p(y_t/x_t, \theta) \end{aligned} \quad (1)$$

Where $t = 1, \dots, T$, x_t/x_{t-1} is the state transition model with initial distribution $p(x_0)$. The equations (1) can be alternatively represented by state probability distribution $p(x_t/x_{t-1})$

and observation probability distribution $p(y_t/x_t)$. The noises in equation (1) and (2) may not be gaussian.

The main aim of the PF is sequentially estimating the predictive distribution $p(x_t/y_{1:t-1})$ and filtering distribution $p(x_t/y_{1:t})$. The filtering distribution function expressed in terms of posterior pdf at time $t-1$ is $p(x_{t-1}/y_{1:t-1})$ as

$$p(x_t/y_{1:t}) \propto \int p(y_t/x_t)p(x_t/x_{t-1})p(x_{t-1}/y_{1:t-1}) dx_{t-1} \quad (2)$$

Analytically implementing the above update (2) is not possible, except few cases. The PF can perform the approximation of the continuous distribution by set of biased particles.

The posterior pdf $p(x_{t-1}/y_{1:t-1})$ is approximated as $\{x_{t-1}^i, w_{t-1}^i\}_{i=1}^N$ where x_t^i is the particle and their corresponding weight are w_t^i and number of particles are N ,

$$p(x_t/y_{1:t-1}) \approx \sum_{i=1}^N w_{t-1}^{(i)} \delta(x_{t-1} - x_{t-1}^{(i)}) \quad (3)$$

Where $\delta(\cdot)$ represents the Dirac delta function. The above equation (2) can be simplified by using the above approximation as

$$p(x_t/y_{1:t}) \propto p(y_t/x_t) \sum_{i=1}^N w_{t-1}^{(i)} p(x_t/x_{t-1}^{(i)}) \quad (4)$$

Using the proposal distribution $q(\cdot)$, the particle filter (PF) creates new particles during the sequential importance sampling (SIS) step based on the previous particles.

$$\bar{x}_t^i \sim q(x_t/x_{t-1}^{(i)}, y_t) \quad (5)$$

$$\bar{w}_t^i = w_{t-1}^{(i)} \frac{p(y_t/\bar{x}_t^i)p(\bar{x}_t^i/x_{t-1}^{(i)})}{q(\bar{x}_t^i/x_{t-1}^{(i)}, y_t)}, i=1,2, \dots, N \quad (6)$$

Once the particle weights are normalized, this collection of weights represents the posterior probability density function (pdf) at time t . After several iterations, the majority of the weights become concentrated on a small number of particles, while the rest have negligible weights. This issue is known as particle degeneracy. If observation processing continues under these conditions, the variance among the weights increases, resulting in a poor approximation of the posterior pdf. To prevent this, particle resampling is used. During resampling, particles with negligible weights are replaced by those with larger weights. Following resampling, the posterior pdf is represented as

$$p(x_t/y_{1:t}) \approx \sum_{i=1}^N \bar{w}_t^i \delta(x_{t-1} - \bar{x}_t^i) = \sum_{i=1}^N \frac{n_t^i}{N} \delta(x_{t-1} - \bar{x}_t^i) \quad (7)$$

Here, n_t^i it indicates the number of times the i th particle is replicated. The resampling step introduces additional issues, such as sample impoverishment and decreased speed. To improve the tracking accuracy and decrease the problem of sample impoverishment clustered weight sorted lookahead PF was proposed.

III. PROPOSED METHOD

In this paper, we proposed a hybrid PSO-GA based particle filter method with GA selection method of clustered sorted weighting lookahead particle filter (PSO-GA-CSLAPF). In the CSLAPF method particles are divided into clusters and then sorted, weighting according to their weights. The particles at a time $t-1$ is considered as x_{t-1}^i and their corresponding weights are w_{t-1}^i for $i=1,2 \dots N$. These particles are propagated through the nonlinear state model

$$v_t^i = f(x_t^i) + e_t, \quad e_t \sim N(0, \tau^2) \quad (8)$$

Where $f(\cdot)$ Is the nonlinear model, and e_t is the state transition noise with mean zero and variance τ^2 . The particles are weighted as

$$a_t^i = w_{t-1}^i p(y_t/x_t^i) \quad (9)$$

The particles at time t are $\{v_t^i, a_t^i\}_{i=1}^N$. The PSO method adaptively directs particles toward areas with a high likelihood. It preserves particle diversity, moves particles in a purposeful way instead of randomly, and speeds up convergence to precisely estimate the state. The particles' positions and velocities are updated according to the following equations.

$$v_i = wv_i + c_1 r_1 (p_i^{\text{best}} - x_i) + c_2 r_2 (g^{\text{best}} - x_i) \quad (10)$$

$$x_i = x_i + v_i \quad (11)$$

In this context, c_1, c_2 are the acceleration coefficients and the random numbers r_1, r_2 are drawn from a uniform distribution $U(0,1)$. The initial value g^{best} is taken from the measurement and p_i^{best} is initially set to zero. With each successive iteration, the position and velocity of every particle are updated.

After PSO step the GA is applied to the particles. The first step in the GA is selection. During selection the clustered sorted weighting lookahead particle filtering method is applied. In this method the particles are divided into K clusters using K-means clustering algorithm. K-Mean clustering algorithm used in PFs to group particles representing similar state hypotheses, allowing efficient approximation of the posterior distribution by cluster centroids and enabling multi-modal tracking and particle reduction. In this algorithm the particle set $X = \{x_1, \dots, x_N\}$ are divided into K clusters. The K-mean clustering algorithm described as follows

- Initially K particles considered as centroids (nearest mean).
 - The particles which are nearer to the centroid is formed as one cluster
- $$C_i = \{x: \|x - \mu_i\| \leq \|x - \mu_j\|, \forall j\} \quad (12)$$
- After the assignment of particles to each cluster, recalculate the centroid of each cluster by averaging the particles weights on it.
 - The step 2 and 3 are repeated until the centroid does not change.
 - The final clusters are $C_1, C_2, \dots, C_k$ and their centroids are $\mu_1, \mu_2, \dots, \mu_k$.

Finally, clusters $C_1, C_2, \dots, C_k$ is generated. After formation of clusters calculate the mixture of weights

$$\pi_k = \sum_{i \in C_k} a_t^i \quad (13)$$

After calculation π_k , with in each cluster the sorted weighting is applied. Consider a cluster C_k containing L particles, where the number of particles L varies from cluster to cluster. The particles within each cluster are arranged in order based on their weights. After sorting, the particles are listed in decreasing order of weight. Calculate the ratio of the sorted weights

$$\eta_t^i = \frac{\bar{w}_t^i}{\bar{w}_t^1}, \quad i=1, 2, \dots, L \quad (14)$$

Where $\bar{w}_t^i$ are the sorted weights located within the cluster. The sorted weight ratios are compared against a certain threshold value. Particles with ratios $\eta_t^i \geq \text{Thresh}$ exceeding this threshold are classified as high-importance region particles, while the others are considered low-importance region particles. The set of indices corresponding to weights greater than the threshold are

$$\psi_t = \{j=1, \dots, P, \eta_t^j \geq \text{Thresh}\} \quad (15)$$

Therefore, within a cluster of L particles, P particles have a normalized weight ratio that exceeds the threshold value, while the remaining $L-P$ particles fall into the low importance

region. The particles in this low importance region are removed and replaced by particles from the high importance region. The indices of these particles are provided by

$$\{o(i)_{i=1,\dots,L}\} = \left\{ \underbrace{\psi_t, \dots, \psi_t}_{\lfloor \frac{L}{P} \rfloor \text{ times}}, \psi_t(1, \dots, L - P \lfloor \frac{L}{P} \rfloor) \right\} \quad (16)$$

The number of indices, P, is repeated until reaching the value L. Consequently, within the cluster, the final weighted particles are $x_k^i = v_t^{o(i)}$, $w_k^i = a_t^{o(i)}$. In some cases, excessive replication may occur within individual clusters, leading to sample impoverishment. To enhance particle diversity, particles with low likelihood are replaced by replicas of particles from high-importance regions, perturbed using Gaussian jitter

$$x_k^i \leftarrow x_k^{o(i)} + \epsilon, \quad \epsilon \sim N(0, \sigma_r^2) \quad (17)$$

After the completion of sorted weighting in each cluster C_k , the particles in each cluster are combined as

$$w_t^i = \pi_k w_k^i \quad (18)$$

The normalized global weights are

$$w_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j} \quad (19)$$

Following the selection process, the next phase in the genetic algorithm is crossover. In this step, the arithmetic crossover method is used to create two parent particles from the set of particles obtained through selection. These parent particles are then combined to produce a new child particle.

$$O = (1-\alpha)P^{(1)} + \alpha P^{(2)}, \quad \alpha \sim [0,1]$$

After the crossover stage, the subsequent step in a genetic algorithm is mutation. This step is used to maintain diversity and avoid particles getting trapped in local minima by adding random noise.

The posterior at a time t can be estimated using the equation

$$\hat{x}_t^i = \sum_{i=1}^N w_t^i x_t^i \quad (20)$$

Compared to traditional particle filters, the proposed approach maintains multimodal posterior distributions, reduces sample impoverishment, and enhances estimation accuracy, all while incurring only a slight increase in computational complexity.

IV. EVALUTION STUDY

To evaluate the performance of the proposed algorithm, we examine two nonlinear models defined by

$$x_t = \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + a_t \quad (21)$$

$$y_t = \tan^{-1} x_t + e_t \quad (22)$$

Here the x_t and y_t are the state transition and observation models. The $a_t \sim N(0, \tau^2)$ and $e_t \sim N(0, \sigma^2)$ are the state transition noise and observations noises. Where τ^2 is the variance of state transition noise and σ^2 is the variance of observation noise. For simulation purpose the parameters considered are particle count $N=1000$, number of timesteps $T=50$, number of clusters $K=3$, Thresh=0.15, $\sigma^2=1$ and $\tau^2=3$. The proposed method performance is compared with some standard particle filters systematic resampling PF, residual resampling PF and sorted weighting lookahead PF(SLAPF).

Figure 1: Proposed method Tracking of true posterior

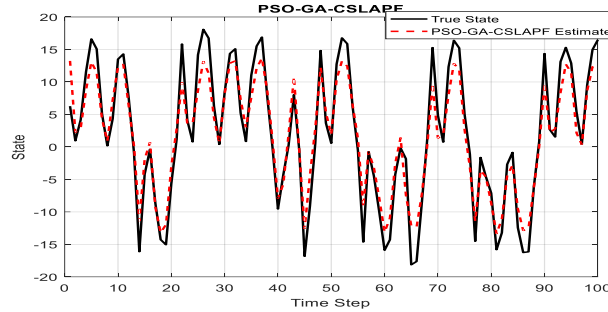


Figure 1 above illustrates the tracking performance of the proposed method. The black line indicates the true posterior, while the red line shows the posterior from our proposed method. From the figure, it is clear that our method's posterior closely follows the true posterior.

Figure 2: RMSE performance with respect to error precision

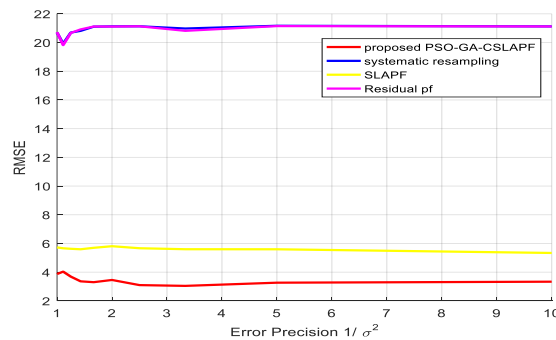


Figure 2 presents the root mean square error (RMSE) for various error precision values. The results of the proposed method are compared to those of conventional particle filter (PF) techniques. This figure clearly demonstrates that the proposed method achieves better RMSE performance at all examined precision levels.

Figure 3: RMSE performance with respect to number of particles N .

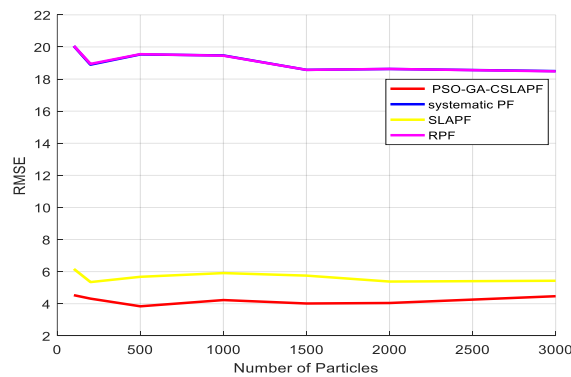


Figure 3 above illustrates the RMSE performance in relation to the number of particles. As the number of particles increases, the RMSE value changes. This figure clearly demonstrates that the proposed method achieves better RMSE performance at all examined values of N .

Figure 4 is the cumulative RMSE performance of different particle filters. The convergence rate indicates how fast the estimated states generated by the particle filter approach the actual target states as the algorithm advances over time. When the particle filter converges quickly, the estimation errors reduce rapidly, resulting in lower RMSE values at each time step. Since CRMSE is computed by summing the RMSE values throughout the entire simulation, a faster convergence rate typically leads to a smaller CRMSE. Conversely, slower convergence results in higher estimation errors during the early and middle phases of tracking, which increases the total CRMSE. From the below figure our proposed method is achieving the low CRMSE value at the middle and final stages. Because of hybrid approach the PSO algorithm the proposed method converges quickly compared to other methods.

Figure 4: Comparison of convergence rate

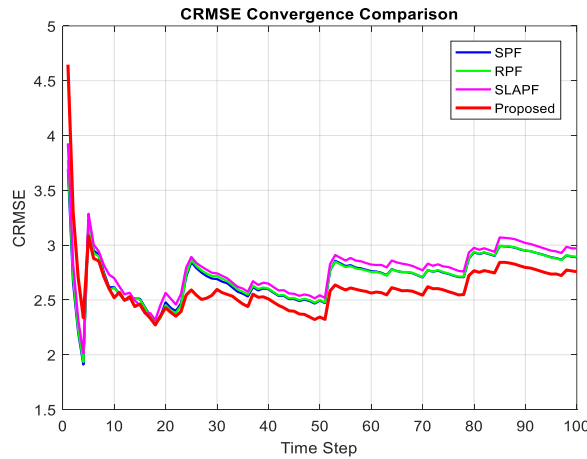
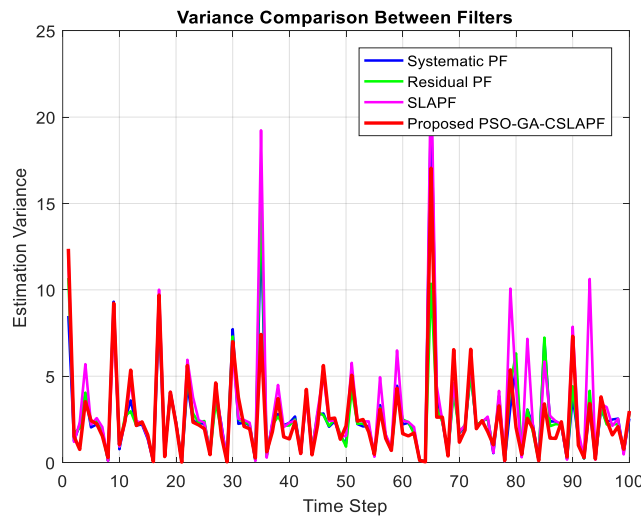


Figure 5: Comparison of Variance



The above Figure 5 represents the variance comparison of proposed method with other particle filters. Our proposed method showing almost similar variance with systematic resampling PF and residual PF.

Track Before Detect Application:

Track-before-detect (TBD) involves tracking a target prior to its detection in environments with low signal-to-noise ratios. In this section, we implement our proposed method for the TBD application and assess its performance. The state space model under consideration is as follows.

$$\begin{bmatrix} x_n \\ y_n \\ \dot{x}_n \\ \dot{y}_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{n-1} \\ y_{n-1} \\ \dot{x}_{n-1} \\ \dot{y}_{n-1} \end{bmatrix} + \begin{bmatrix} v_{n,1} \\ v_{n,2} \\ v_{n,3} \\ v_{n,4} \end{bmatrix} \quad (23)$$

In this model, the target's movement is represented Cartesian coordinate system. The target's state vector, $X_n = [x_n \ y_n \ \dot{x}_n \ \dot{y}_n]^T$ which is a 4x1 matrix, includes both position and velocity components along the x and y axes. The state transition matrix is

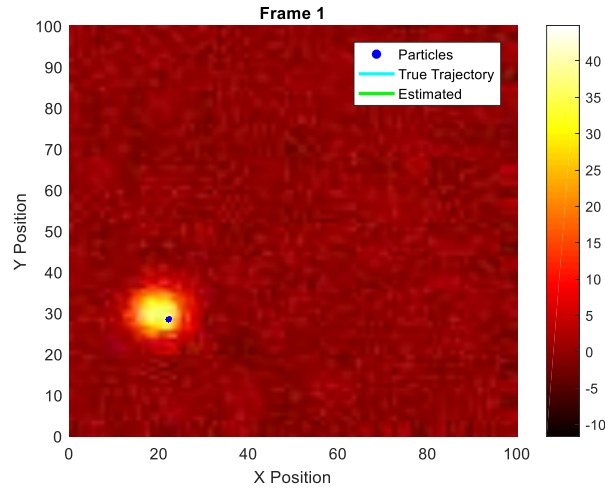
$F_n = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ of size 4x4. Where $\Delta t=1$ is considered for the simulation. State noise vector is $V_n = [v_{n,1} \ v_{n,2} \ v_{n,3} \ v_{n,4}]^T$ of size 4x1 and the Covariance matrix is $Q_n = \text{diag}[1 \ 1 \ 0.5 \ 0.5]$.

The measurement model is given by

$$Z = X_n + W \quad (24)$$

W represents the measurement noise (or observation noise), which is usually assumed to be zero-mean white Gaussian noise with a known covariance matrix $R_n = \sigma_y^2 I$.

Figure 6: Frame 1 particles are distributed throughout the surface



In this evaluation, the simulation is conducted over 100-time steps with 1000 particles ($N=1000$), and parameters set to $\Delta t=1$ and $\sigma_y^2 =$. Figure 6 above displays Frame 1, illustrating the distribution of particles across the surface used to locate the target.

Figure 7: Frame 3 the particles start tracking the target

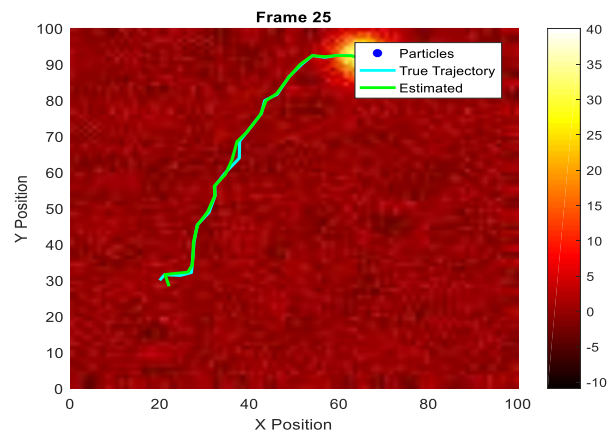


Figure 8: Frame 25 the tracking performance of particles

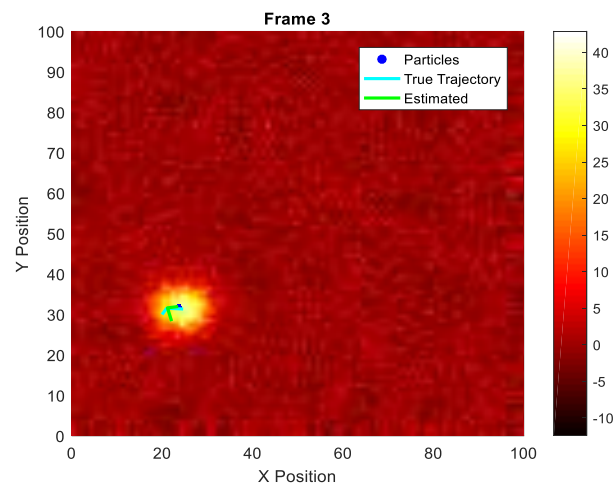


Figure 9: The tracking trajectory along X and Y directions

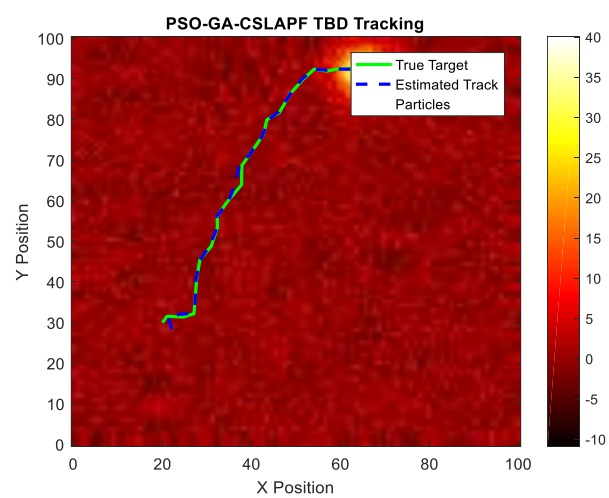


Figure 7 represents frame 5, where the particles have successfully located the target and begun tracking it. Figures 8 display frames 25, illustrating how the particles maintain tracking of the target over time following its identification.

Figure 10: The tracking RMSE of proposed method

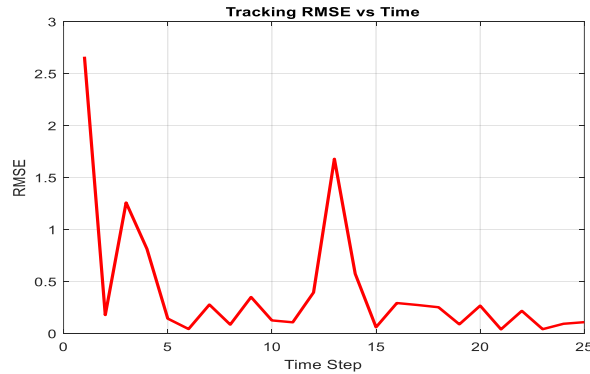


Figure 9 shows both the actual and estimated paths of the target. The strong similarity between these paths indicates that the proposed method successfully estimates and tracks the target. Figure 10 shows the RMSE performance of the proposed method

Figure 11: Probability of Target presence.

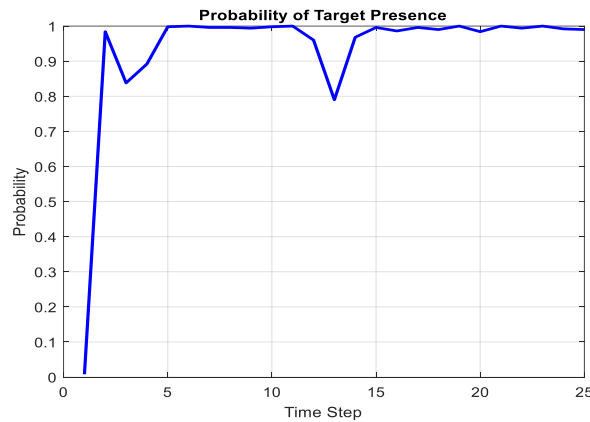


Figure 11 shows the likelihood of the target being present in the tracking environment. Until time step 3, this probability stays under 0.9, reflecting uncertainty about the target's existence. After time step 3, the probability rises above 0.9, confirming the target's detection, and the tracking process proceeds accordingly.

Figure 12 shows the RMSE results for the track-before-detect (TBD) application. The performance of the proposed method is compared with conventional particle filtering techniques, including systematic resampling PF (SRPF), residual PF (RPF), and sorted weighting lookahead PF (SLAPF). The figure demonstrates that the proposed method consistently achieves lower RMSE values compared to the standard particle filters.

Figure 13 shows a bar chart displaying the average RMSE values for all the methods evaluated. The proposed approach achieves the lowest average RMSE, further demonstrating its better estimation accuracy compared to traditional particle filtering methods.

In summary, the simulation results show that the proposed method achieves better tracking accuracy, lower RMSE and CRMSE, and greater robustness for TBD applications, significantly decreasing estimation errors compared to traditional particle filter-based methods.

Figure 12: Comparison plot of RMSE performance of Proposed method with standard PFs

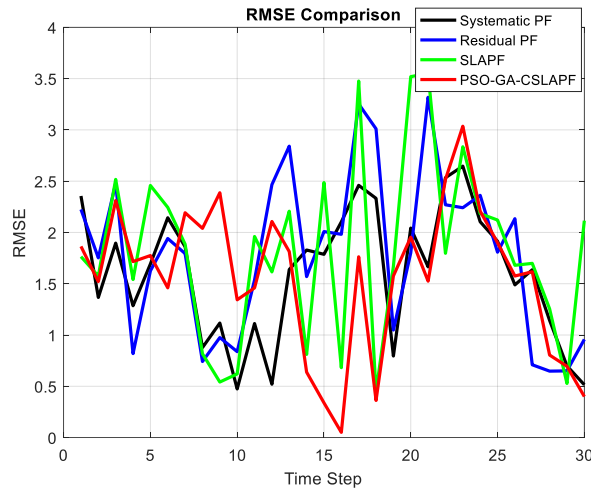
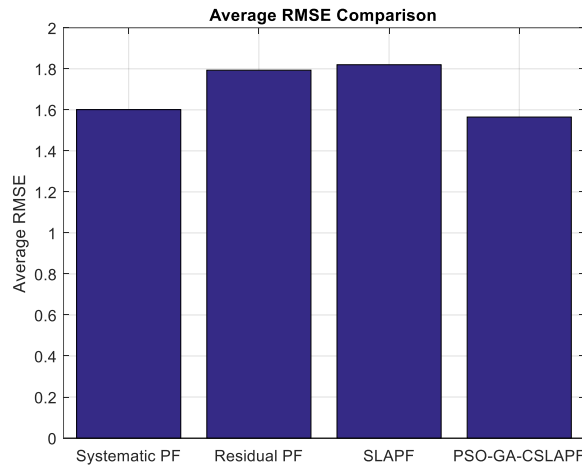


Figure 13: Average RMSE Bar Chart



V. CONCLUSION

In this paper, a hybrid particle filter particle swarm optimization and genetic algorithm based particle filter with GA selection process of a clustered sorted weighting lookahead particle filter is proposed. In the proposed algorithm, during the selection process the total particle set is first partitioned into multiple clusters using the K-means clustering algorithm. Within each cluster, particles are then sorted and weighted according to their importance weights, after which all clusters are merged to create the updated particle set. The simulation results clearly show that the proposed method greatly lowers the RMSE and cumulative RMSE throughout the filtering process, which improves particle diversity and leads to quicker convergence. Additionally, within the track-before-detect (TBD)

framework, this approach allows for earlier and more reliable target detection, enabling accurate target tracking to begin sooner. As a result, the proposed method delivers better RMSE performance and demonstrates greater robustness compared to traditional particle filter-based methods.

REFERENCES

1. Miodrug Bolić, Petar M. Djurić, and Sangjin Hong, "New Resampling Algorithms for Particle Filters". IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003.
2. Ramakrishna G, P. B. Choppala, James Stephen M and Paul D. Teal, "A sorted weighting lookahead sampling scheme for accurate and fast particle filtering," second international conference on Range Technology (ICORT), IEEE, August 2021.
3. Tiancheng Li, Miodrag Bolić, and Petar M. Djurić, "Resampling Methods for Particle Filtering (Classification, implementation, and strategies)" Signal Processing Magazine IEEE, May 2015.
4. Kitagawa G, "Monte Carlo filter and smoother for non-Gaussian non-linear state space models," Journal of Computational and Graphical Statistics, pp. 1–25, 1996.
5. J. Carpenter, P. Clifford, and P. Fearnhead, "An improved particle filter for nonlinear problems," IEEE Proc. Radar, Sonar and Navigation, vol.146, pp. 2–7, July 1999.
6. M. Bolic, Petar M. Djurić, and Sangjin Hong, "Resampling algorithms for particle filters: A computational complexity perspective," Proc. EURASIP Journal on Advances in Signal Processing, pp. 2267–2277, 2004.
7. N. Gordon, D. Salmond, and A. F. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation." IEE Proc. F, Radar Signal Process., 140:107–113, 1993
8. Kennedy J, Eberhart R C. (1995) " Particle swarm optimization", Proceedings of the 1995 IEEE International Conference on Neural Networks, 4: 1942–1948.
9. Xuerong C, Jin Y, Juan Li, Chunlei W, "Improved Genetic Algorithm to Optimize the Wi-Fi Indoor Positioning Based on Artificial Neural Network", IEEE Access. January 2020.