

# A New Motion Model Selection Approach for Multi-Model Particle Filters

Murat Barkan UCAR<sup>1</sup>, Derya YILMAZ<sup>2</sup>

<sup>1</sup> Havelsan Technology Radar Inc., Çankırı Road 7. Km, 06750 Akyurt-Ankara, Turkey

<sup>2</sup> Dept. of Electrical and Electronics Engineering, University of Başkent, Eskişehir Road, 20. Km, 06790 Ankara, Turkey

murat.ucar@htr.com.tr, derya@baskent.edu.tr

Submitted March 26, 2019 / Accepted October 15, 2019

**Abstract.** *One of the important factors in real-time tracking of the moving radar targets is the speed of the algorithm. In the multi-model particle filters (MMPFs) which is frequently preferred tracking of such targets, the numbers of particles and motion models are important parameters determining the speed of the filter. Reducing the number of particles and/or the model transitions processes as much as possible will facilitate real-time tracking of moving targets by accelerating the algorithm. In this study, for reducing the time cost of the MMPF, a new approach called weighted statistical model selection (WSMS) which reduces the number of model estimation calculations is proposed. A new basic MMPF algorithm that allows the use of the WSMS approach is also constituted. In order to evaluate the success of the WSMS; the MMPFs integrated with the WSMS, are simulated for different noise variances, particle numbers, and scenarios. The simulation results are compared based on processing time and prediction error criterions. The results demonstrate that the WSMS approach increases the speed of the algorithm by reducing the processing time at high rates without any change in the prediction error and, thus it can be used in real-time tracking of the moving targets.*

## Keywords

Multiple maneuvering targets tracking, time cost, target tracking algorithm, model selection

## 1. Introduction

The radar systems have different infrastructures and a wide range of applications that focus on the target detecting, positioning, tracking, etc. [1–3]. Therefore, the approaches used in the development of these systems are quite varied. In this context, tracking of moving targets is one of the important issues based on the estimation of the target location by using noise-based measurements. The possible maneuver of the target is tried to be estimated by using a system (motion) model and the measured values from the target. In the maneuvering target tracking problem that includes a nonlinear structure, the Kalman Filter (KF)

which tries to resolve the problem by linearizing it, and the improved versions of this filter have been frequently used approaches [4–6]. Since the estimation performance of KF is weak for the nonlinear systems having non-Gaussian distribution, the use of particle filters (PFs), which is also known as the Monte Carlo (MC) filter approach, has come into prominence in the solution of these problems. In PF, nonlinear structures in the kinematic variables of the system and/or measurements can be better estimated by using the samples (particles) from the state space [7–10].

For tracking a moving target, it is very important to identify all possible maneuvers of the target. However, there is no single model representing all possible movements of a maneuvering target. In this case, multiple model-based approaches offering different models for each maneuver are used. Multi-model particle filters (MMPFs) are frequently used for this purpose [11–14]. The working principle of the MMPFs is generally to run all particles of the PF for each step in each model. However, this procedure reduces the algorithm speed by increasing the computational cost of the processing unit. For real-time tracking of targets, it is important that the algorithm is fast. Reducing the number of particles and/or models used in the estimation plays an active role in increasing speed.

In recent years, there have been several studies trying to reduce the complexity of the processor or to increase the processing speed of MMPFs. Yang and Zhao added the probabilistic data relationship structure to the MMPF to overcome the measurement-related uncertainty and the increment in the process complexity [15]. The proposed approach is compared to a traditional PF for very low noise intensities (0.0001, 0.001 and 0.005) and 2000 particles. According to their results, the error performance of the proposed algorithm is better within all noises and, the processing time is halved for 0.005 noise level. Another PF is developed by adding a gray estimation algorithm to standard PF [16]. This algorithm does not require a model approach and predicts the system state based on past measurements. The results of simulation given for 1000 particles showed that the error performance of the proposed PF is approximately similar to that of MMPF and it uses about 10% less time than MMPF. Hong et al. proposed a MMPF structure consisting of a statistical model for maneuvering

motion [17]. It is stated that, for the 0.01 noise level and 1000 particles, the presented algorithm has similar error to traditional MMPF and lower processing time at resampling step. In another study, a novel MMPF based on composite sampling is presented [18]. The results of this study showed that the processing time performance of the proposed filter is similar to the single model PF.

In this study, a new model determination approach called the weighted statistical model selection (WSMS) algorithm is proposed for MMPFs. The proposed method determines the motion model to be used in the next estimation step by using the statistical information obtained from previous model estimates. In this approach, the model calculation process is made for a specified model instead of all models so the processing time of MMPFs can be effectively reduced. In the simulations, three MMPFs are tested on various conditions. The WSMS is integrated into two of these filters. The results are interpreted based on performance criterions.

## 2. Particle Filter

The PF is used for estimation of posterior density function for target state in non-linear and non-Gaussian filter problems. By using the sequential MC estimation method in the PFs, it is tried to determine the posterior distribution using a large number of weighted particles that are samples taken from the state space. The particles are weighted according to the dynamics of the target and, the particles with high probability assumed to be closer to the object remain more in the system [9], [19].

For PF, the state and the observation (measurement) equations are given in (1) and (2), respectively. In (1),  $\mathbf{x}_k$  is state,  $\mathbf{u}_k$  is processing noise and  $f_k(\cdot)$  is system transition function. In (2),  $\mathbf{y}_k$  is observation vector,  $\mathbf{v}_k$  observation noise,  $h_k(\cdot)$  is observation function.  $f_k(\cdot)$  and  $h_k(\cdot)$  are non-linear functions.  $k$  is the time variable in here.

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{u}_k) \quad (1)$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \quad (2)$$

In order to approach the posterior distribution in the PF, the state density at  $k$  is expressed by  $M$  particles weighted by  $w$ . Weights indicate the importance of the particles. A random measurement with  $M$  particles at  $k$  is defined as in (3), where  $\mathbf{x}_k^{(m)}$  is the  $m$ -th particle at  $k$ ,  $\mathbf{x}_{1:k}^{(m)}$  are the particles from time 1 to  $k$ ,  $w_k^{(m)}$  is the weight of the particle at  $k$ .

$$\{\mathbf{x}_{1:k}^{(m)}, w_k^{(m)}\}_{m=1}^M \quad (3)$$

The probability distribution function according to the observation values is as in (4). Here,  $M$  denotes the total number of particles (randomly generated and weighted),  $w$  is the weight of particle,  $\mathbf{x}$  is the particle position,  $\mathbf{z}$  is the observation value and the  $\delta$  indicates the Dirac function.

$$p(\mathbf{x}_{1:k} | \mathbf{z}_{1:k}) \cong \sum_{m=1}^M w_k^{(m)} \delta(\mathbf{x}_{1:k} - \mathbf{x}_{1:k}^{(m)}) \quad (4)$$

There are three basic steps in the PF: Formation of particles (initialization), weighting (calculation/updating and standardization) and resampling.

Formation of particles: The process of forming particles is made by using the importance density function  $q$  (5). Here,  $\mathbf{x}$  and  $\mathbf{z}$  represent state and observation values, respectively, and  $m$  is the number of particles at the  $k$  state

$$\mathbf{x}_k^{(m)} \cong q(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}, \mathbf{z}_{1:k}). \quad (5)$$

Weighting: The calculation and standardization of the weights are made by importance sampling (6) that allows Bayesian filtering with the MC method. The distribution parameters are estimated by using the randomly selected particles of the posterior distribution [8, 9, 19]. The normalization is made by (7)

$$w_k^{(m)} \cong \frac{P(\mathbf{x}_{1:k}^{(m)} | \mathbf{z}_{1:k})}{q(\mathbf{x}_{1:k}^{(m)} | \mathbf{z}_{1:k})}, \quad (6)$$

$$w_k^{*(m)} = \frac{w_k^{(m)}}{\sum_{i=1}^M w_k^{(m)}}. \quad (7)$$

Resampling: Over time, the weights of selected particles increase and the others' weights approach to zero. In this case, the representation of the posterior distribution with a small sample set reduces the estimation accuracy. To prevent this situation named deterioration phenomenon, resampling is done. In resampling, the low weighted particles are discarded from the sample set, and high weighted samples are duplicated in resampling. This process causes sample impoverishment problem and it can be prevented by adding noise samples that have small variance Gaussian distribution to new samples [8], [19]. The most commonly used resampling methods are systematic and sequential importance resampling (SIR). In this study, because of easy application, SIR method is used [20–22].

## 3. The Motion Models

In order to be able to follow the moving target in the best way, it is necessary to model the target movement and to obtain maximum information from the observations. The problem of moving target tracking is defined by the state-space model consisting of two parts: the motion model (8) and the observation model (9) [20]

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k^u \mathbf{u}_k + \mathbf{G}_k w n_k, \quad (8)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k. \quad (9)$$

In (8),  $\mathbf{F}_k$  is state transition matrix,  $\mathbf{x}_k$  is state (prediction) vector,  $\mathbf{G}_k^u$  is control input matrix,  $\mathbf{u}_k$  is control input vector, the  $w n_k$  is process noise, and the  $\mathbf{G}_k$  is process noise

matrix. In (9),  $\mathbf{H}_k$  refers to the observation matrix,  $\mathbf{x}_k$  is the state vector, and  $\mathbf{v}_k$  is the observation noise. The movement of the target can be expressed by various models. The models used in this study are explained below.

Constant velocity (CV): This model indicates that the target moves at a constant velocity on a straight line (10).  $\mathbf{F}_k$  and  $\mathbf{G}_k$  matrices are given by (11) and (12), respectively.  $\Delta t$  is the difference between the time  $t_{(k+1)}$  and  $t_k$  [23].

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k w n_k, \quad (10)$$

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} I & \Delta t \cdot I \\ 0 & I \end{bmatrix}, \quad (11)$$

$$\mathbf{G}_k = \begin{bmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} = \begin{bmatrix} (\Delta t^2/2) \cdot I \\ \Delta t \cdot I \end{bmatrix}. \quad (12)$$

Constant acceleration (CA): In this model (13), the target moves with a constant acceleration and the acceleration change is expressed by noise [20].  $\mathbf{F}_k$  is given by (14).

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + w n_k, \quad (13)$$

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

Coordinated turn (CT): In CT model (15), the target is assumed to move with constant speed, constant altitude and constant angular rotation ratio ( $\omega$ ) [20], [23]. The state transition matrix is given by (16).

$$\mathbf{x}_{k+1} = \mathbf{F}_k(\omega) \mathbf{x}_k + w n_k, \quad (15)$$

$$\mathbf{F}_k(\omega) = \begin{bmatrix} 1 & \sin(\omega T)/\omega & 0 & -[1 - \cos(\omega T)]/\omega \\ 0 & \cos(\omega T) & 0 & -\sin(\omega T) \\ 0 & [1 - \cos(\omega T)]/\omega & 1 & \sin(\omega T)/\omega \\ 0 & \sin(\omega T) & 0 & \cos(\omega T) \end{bmatrix}. \quad (16)$$

## 4. Multi Model Particle Filter

All possible maneuvers of the target should be introduced to the MMPF for correct tracking. The MMPF works by calculating all models introduced to the system for the  $k$  step and predicting the target state for the  $k + 1$  step. In this study, three MMPFs are evaluated to reveal the performance of the WSMS approach. The common feature of these MMPFs is that the calculations are made for all models ( $X$  model) and  $N$  particles (for each model) at each step. For this reason,  $X \times N$  particles exist in the system during the calculation phase. After calculation, the number of particles is reduced back to  $N$  for the next step.

Algorithm 1 (A1): According to A1,  $X \times N$  particles at each step are weighted according to their fitness and Markov transition possibilities. In the resampling step, particles are selected according to their weight to reduce the number of particles to  $N$  again. A1 does not include a model prediction step, and with this feature, it carries the basic characteristics of a traditional MMPF [14].

Algorithm 2 (A2): In A2,  $N$  particles generated for each model propagate separately through all models for each step. At the importance step, particles are weighted and resampling is applied. Target's location is predicted with division of the sum of all models estimations by the sum of particle weights which are obtained by resampling. Thus, A2 does not need model transition probabilities for determining the model and has a better processing time performance when compared to a traditional MMPF [17].

Algorithm 3 (A3): A3 is proposed in this study and has a simple approach. A3 uses the structure of A2 up to the prediction step. The prediction errors obtained for each model at the previous step are evaluated for model selection at the current step. A3 selects the model that has the least prediction error at the previous step. When the prediction process is started, the system has total  $X \times N$  particles and  $N$  prediction values belonging to the target location. In order to determine which of the  $N$  predictions is correct, the amount of error between the prediction and the observation values at  $t - 1$  is examined. Based on the assumption that the movement will be more likely to continue with the model that has the least error, this model is used at time  $t$ . The Pseudo code for A3 can be summarized as follow:

$i = 1:N$   
 $j = 1:X$

- Draw the particles from impotence density function:  $\mathbf{x}_t^{(i)} \cong q(\mathbf{x}_t^{(i,j)} | \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_{1:t})$
- Update the weights of particles:  
 $w_t^{(i,j)} \cong \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(i,j)}) p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i,j)} | \mathbf{x}_{1:t-1}^{(i)}, \mathbf{y}_{1:t})} w_{t-1}^{(i,j)}$
- Normalize the  $X \times N$  particles' weights:  
 $w_t^{(i,j)} = \frac{w_t^{(i,j)}}{\sum_{i=1}^M w_t^{(i,j)}}$
- Resample the particles with SIR
- Take the observation value (angle and distance information from  $x$  and  $y$  coordinates) from the estimated value obtained for each model:  
 $\theta = \tan^{-1}(x/y), R = \sqrt{x^2 + y^2}$
- Calculate the error rates at time  $t - 1$  for each model
- Choose the model with the lowest error rate for  $t$

## 5. Proposed Model Selection Approach

In the MMPFs, which are frequently preferred in real-time target tracking, decreasing the processing time is an important issue. It is necessary to keep the number of particles and/or the prediction operations at the optimum. In this study, the weighted statistical model selection (WSMS) algorithm aimed to perform the model estimation process with the least operation is presented. The WSMS algorithm chooses the most appropriate motion model for the target in each step and reduces the processing time by calculation for only selected model instead of all models. We are starting from the fact that the target, which starts to act following a motion model, must continue its movement with the same model during a period of time to complete its movement. From this point of view, the possibility of continuing the last motion model of the target will be high. The previous movement dynamics of the target are evaluated statistically and included in the probabilistic evaluation for predicting the next motion model. For model selection, the error values are must be under the threshold value which is dynamically set at each time step. Considering that there are  $X$  models and  $N$  particles (for each model) in the MMPF, the total number of operations is  $X \times N$ . The number of operations can be reduced to  $N$  with WSMS. This makes the MMPF faster up to  $X$  times. The WSMS algorithm includes the following steps:

1. A window  $P$  is created and an error threshold is assigned. The minimum length of  $P$  is as the number of models defined in the system.

2. The PF is operated as much as the number of models ( $X$  times).

- a) All models are calculated in each time step.
- b) The model has the least error at  $t - 1$  is used at  $t$ .
- c) The information about the selected model for each step is stored in the  $P$  window created in point 1.

3. When the first  $X$  time step is completed, the WSMS algorithm starts to work from step  $X + 1$ , where weighting and statistical information are used together for the model selection.

- a) The probabilities of the models are determined by considering the number of usages in the  $P$ .
- b) The weights of the models within the  $P$  are assigned by decreasing so that the last selected model has the largest weight.
- c) The weights and the probabilities of the models are multiplied and the multiplication results are summed for each model.
- d) The model with the largest total value is selected and the calculation is made for only this model at the next step.

4. The information about the selected model is saved at the  $P$  and the target position is estimated by using this model.

5. The error between the actual observation and the estimated values is calculated for the current step.

6. The size of  $P$  and threshold are updated.

7. The error value is compared to the threshold.

8. As a result of this comparison;

- a) If the error value is smaller than the threshold, the algorithm goes to 3a).
- b) If the error value is higher than the threshold, the calculation for the next step is done for all models. The model that has the least error is selected and stored to the  $P$  window. The algorithm goes to 2a).

In the WSMS algorithm, if the threshold is greater than the current error, then the threshold is updated with current error value for the next step. If the threshold is less than the current error, then the same threshold value is used in the next step. So, the threshold is always equated to the smallest error value obtained from operations. In this case, after a while, the current error values will always be higher than the threshold; therefore, the WSMS algorithm will lose its function and all motion models will need to be calculated for all remaining steps. To avoid this situation, it is necessary to prevent the threshold dropping below a certain value, an inferior threshold is used. The too-small inferior threshold will cause the proposed approach to lose its effectiveness, and the large values will lead to decreasing in model estimation accuracy. Therefore, it is appropriate to determine the inferior threshold with an intuitive approach; it is recommended to select between 0.05 and 0.1. In the case of an interruption in the target tracking in some way, the PF cannot correctly weight the particles. In this condition, according to the proposed algorithm, because the error value will be higher than the threshold, all models need to be calculated for the next step.

The WSMS works based on the statistics of the usage of models selected in previous steps. In order to obtain a statistic, the algorithm needs to work for several times. Initially, it is suggested that the window size should be at least equal to the number of models. The initial window size does not affect the processing speed of the algorithm because WSMS is activated after the algorithm runs up to the number of models and, the window size begins to be updated. So, the initial window size will already change after a few steps. However, the initial window size affects the new window size at updating step. The window size is also updated according to the comparison between the threshold and the observed error. If the current error is smaller than the threshold, then the window is enlarged as much as "current window size/number of models", if not, then the window is reduced at the same rate. Since the window size is used to determining model probabilities, it should not be forgotten that the large selections can lead to incorrect model estimation especially for dynamic routes.

An example of the WSMS operation is given below. For a stored window with  $P = 10$  elements,  $W_P$  refers to model weights ( $W_P > \dots > W_4 > W_3 > W_2 > W_1$ ), and  $M_i$  refers to model type ( $i$ : model number). The selected models and the weights assigned by decreasing from the last to the first, are given at the first and second rows of Tab. 1, respectively.

At first, the model probabilities ( $M_P$ ) in the window are calculated. Here; the probability of  $M_2$  is 0.4, the proba-

$M_i$	$M_2$	$M_2$	$M_2$	$M_2$	$M_1$	$M_1$	$M_1$	$M_3$	$M_3$	$M_3$	?
$W_p$	1	2	3	4	5	6	7	8	9	10	-
$M_p$	0.4	0.4	0.4	0.4	0.3	0.3	0.3	0.3	0.3	0.3	-
$WM_p$	0.4	0.8	1.2	1.6	1.5	1.8	2.1	2.4	2.7	3	-

Tab. 1. Model weighting.

bility for  $M_1$  and  $M_3$  is 0.3 (Tab. 1, 3<sup>rd</sup> line). The weighted model probabilities ( $WM_p$ ) are obtained by multiplying the weights and the probabilities (Tab. 1, last line). For each model, these values are summed and the total weighted model probabilities are found as 5.4, 4 and 8.1 for  $M_1$ ,  $M_2$  and  $M_3$ , respectively. In this case, the WSMS will select  $M_3$  which has the largest total probability for the next step. Thus, instead of calculating all models at the next step, only calculation of  $M_3$  will be sufficient.

### 6. Implementation and Results

The noise in target tracking is the most important factor increasing the amount of error. Increasing the number of particles reduces the amount of error, but increases the processing time. The effects of MMPFs and WSMS approach, on the target tracking accuracy and the processing speed, are investigated for four noise levels and several particle numbers. The WSMS is integrated to two MMPFs (A2 and A3) as a model selection method. Since the structure of A1 has not model the prediction step, the WSMS cannot be integrated into A1. The simulations are performed on three scenarios (S1, S2 and S3) by using 100 MC runs for 2D tracking. S3 is more difficult and dynamic than the others. S2 and S3 have four motion models although S1 has three. The performances of MMPFs are evaluated according to the root mean square error (RMSE) and processing time. The RMSE for each simulation is given by averaging the RMSE values obtained from each MC run. In the simulations, the initial window size is set to 10 for the first two steps and the threshold is set to 0.07 for the first (number of models) steps. After these steps, the window size and error threshold are dynamically updated. The other simulation parameters for all scenarios are as follows:

- Sampling interval: 1 s.
- Total number of steps: 100.
- The noise variance:  $\sigma^2 = 0.01, 0.1, 0.15$  and  $0.2$ .
- Number of particles: 1000, 3000, 5000 and 7000.
- Models: Combination of CV, CA and CT models.
- Observer (S1, S2, S3):  $(-6, 12), (4, 10), (-3, 16)$ .
- Target maneuver area (km<sup>2</sup>) (S1, S2, S3): 196, 116, 665.

#### 6.1 The Results of Simulations

Simulations are realized for five algorithms (A1, A2, A3, A2+WSMS, A3+WSMS). The tracking results from one simulation with 100 MC runs ( $\sigma^2 = 0.15$  and 5000 parti-

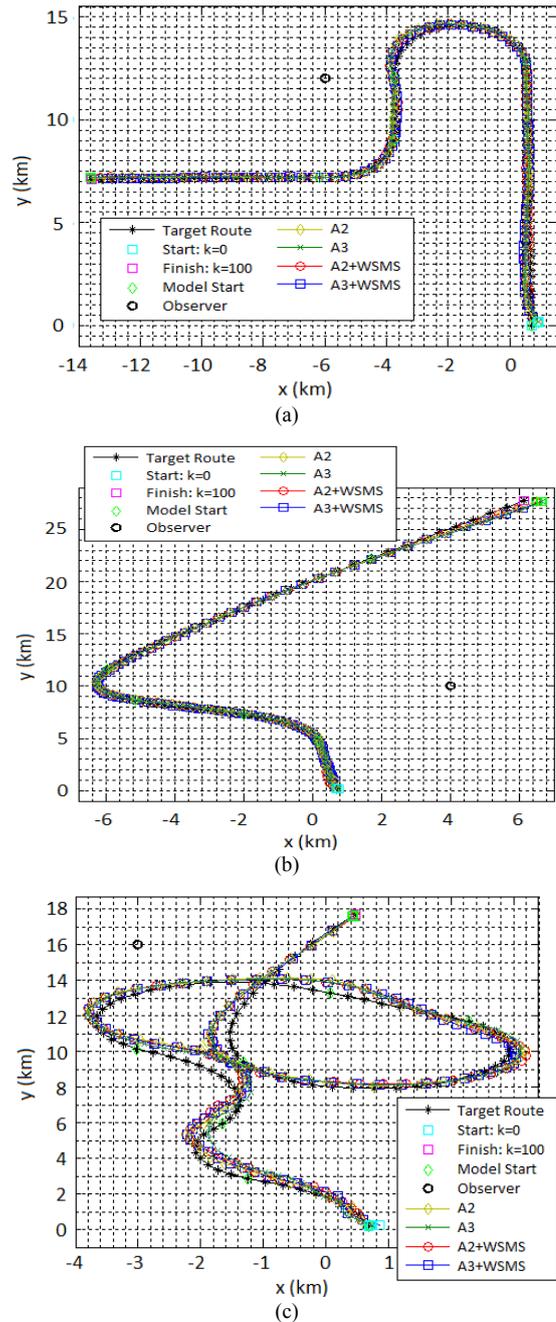


Fig. 1. The results of target tracking a) S1, b) S2, c) S3.

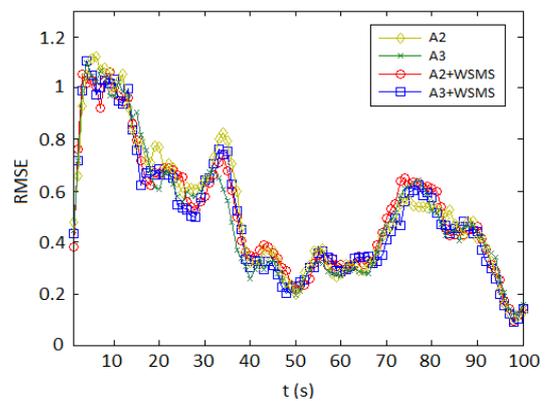


Fig. 2. RMSE for S3.

Number of Particles		$N = 1000$		$N = 3000$		$N = 5000$		$N = 7000$	
Algorithms	Noise Level	RMSE	$t$	RMSE	$t$	RMSE	$t$	RMSE	$t$
A1	$\sigma^2 = 0.01$	0.2259	1.3814	0.1012	5.8019	0.0946	12.6163	0.0928	21.9441
	$\sigma^2 = 0.1$	1.2597	1.3586	0.8736	5.8333	0.8342	12.7216	0.7214	21.9434
	$\sigma^2 = 0.15$	2.8748	1.4716	1.8862	5.6207	1.6265	12.3564	1.6175	21.1936
	$\sigma^2 = 0.2$	5.5993	1.7706	2.7044	5.5141	2.5273	12.2517	2.3059	21.0791
A2	$\sigma^2 = 0.01$	0.2541	0.6933	0.2171	1.8217	0.2115	3.0021	0.2021	4.1009
	$\sigma^2 = 0.1$	0.5311	0.6374	0.5082	1.7137	0.4839	2.8577	0.4838	3.7804
	$\sigma^2 = 0.15$	0.8901	0.6327	0.6951	1.6868	0.7010	2.7723	0.6810	3.8148
	$\sigma^2 = 0.2$	1.1428	0.5962	1.0299	1.6176	0.9910	2.6415	0.9276	3.6122
A3	$\sigma^2 = 0.01$	0.2590	0.6858	0.2109	1.8078	0.2042	2.9503	0.1944	4.3019
	$\sigma^2 = 0.1$	0.5347	0.6362	0.4977	1.7061	0.4906	2.8138	0.4764	4.0644
	$\sigma^2 = 0.15$	0.8610	0.6185	0.7039	1.6484	0.6945	2.6806	0.6999	3.6851
	$\sigma^2 = 0.2$	1.1831	0.5977	1.1602	1.5964	0.9872	2.5837	0.9647	3.5766
A2+WSMS	$\sigma^2 = 0.01$	0.2768	0.5527	0.2066	1.3597	0.2056	2.1834	0.2061	3.0242
	$\sigma^2 = 0.1$	0.5476	0.6083	0.4861	1.5182	0.4710	2.5156	0.4717	3.4567
	$\sigma^2 = 0.15$	0.8582	0.5576	0.7310	1.5189	0.6939	2.4221	0.6816	3.3481
	$\sigma^2 = 0.2$	1.2311	0.5533	0.9899	1.5168	1.0120	2.3126	0.9128	3.4886
A3+WSMS	$\sigma^2 = 0.01$	0.2605	0.5406	0.2066	1.3246	0.2091	2.0404	0.1872	3.2045
	$\sigma^2 = 0.1$	0.5182	0.5823	0.4921	1.5299	0.4816	1.7613	0.4882	2.5310
	$\sigma^2 = 0.15$	0.8843	0.5240	0.6864	1.5108	0.6760	2.4217	0.6659	3.3286
	$\sigma^2 = 0.2$	1.2023	0.5516	1.0072	1.4658	1.0456	1.7234	1.0515	3.1746

Tab. 2. Simulation results for S3.

cles) are given in Fig. 1 for three scenarios. The RMSE graphs obtained from S3 is presented in Fig. 2. The RMSE and processing time ( $t$ ) performances obtained from S3, which is the most dynamic scenario compared to the others, are presented in Tab. 2. The comparisons of the algorithms for the results in Tab. 2 are given below. The results obtained from S1 and S2, which are similar and slightly better than those of S3 in some cases, are also expressed.

In A1, as the number of particles and the levels of noise increases, the processing time and the RMSE greatly increase when compared to those in A2 and A3. It is said that A2 and A3 work more efficiently than A1. In comparison of A1 and A3, A1 has good error performance for only 0.01 noise level; in all other cases, A3 yielded better results that are ranging from 34% to 79% (S1: 44–76%, S2: 42 to 78%) for RMSE and 50% to 83% (S1: 51–83%, S2: 49 to 83%) for processing time. In A1 and A3+WSMS comparison, at the lowest noise level, A1 has better RMSE than A3+WSMS, but has a worse performance with up to 85% for processing time. For all other noises and particle numbers, A3+WSMS has a better performance than A1 at an average of 57% (S1: 60%, S2: 59%) for RMSE and 77% (S1: 77%, S2: 81%) for processing time.

In the processing time comparisons for A2 and A2+WSMS, for all particle numbers, WSMS provided a decrement of an average of 25%, 9%, 12% and 7% (S1: 40%, 24%, 15% and 14%, S2: 40%, 27%, 16% and 12%) at the 0.01, 0.1, 0.15 and 0.2 noise levels, respectively. In the RMSE comparisons, WSMS caused an insignificant increment of an average of 0.12% (S1: 0.9%, S2: 0.22%) for all cases. It can be said that the prediction error remains the same. A2 and A3 have similar responses in all cases. A3 has slightly better performance than A2, at an average of, 0.62% (S1: 0.71%, S2: 1.3%) in RMSE and 0.55% (S1: 0.45%, S2: 1.96%) in processing time.

In point of the processing times of A3 and A3+WSMS; 26%, 24%, 11% and 15% (S1: 41%, 24%, 17% and 15%, S2: 38%, 25%, 19% and 21%) improvements are achieved at an average for noise levels of 0.01, 0.1, 0.15 and 0.2, respectively, with the WSMS integration. For all cases, the RMSE improved by 0.64% (S1: 3%, S2: 0.1%), on average. It can be said that WSMS does not create any significant change in the RMSE, and it provides an increment in the processing speed performance by 19% (S1: 24%, S2: 24%), on average.

When A2+WSMS and A3+WSMS are compared for all cases, A3+WSMS has a performance increment of 0.11% (S1: 0.2%, S2: 1%) in the RMSE and 7% (S1: 1%, S2: 3%) in the processing time, on average. It is seen that these two algorithms produce similar results and A3+WSMS works slightly better than A2+WSMS.

When the results in Tab. 2 are evaluated graphically, the effect of WSMS on A2 and A3 can be better seen. The graphics do not include A1 because it has the worst values for both performances. The RMSEs and processing times for all particle numbers at the noise level  $\sigma^2 = 0.15$  are given in Fig. 3(a) and (b), respectively. Referring to Fig. 3(a), it is seen that A3+WSMS has the least error in all particles except 1000. With the WSMS, the RMSE performance of A3 improved by about 2%, but there is no significant change for A2 (0.16% increment only). A2 and A3 exhibit similar performances in RMSE, whereas A3+WSMS is 2% better than A2+WSMS. WSMS decreases the processing time by approximately 12%, 10%, 13% and 12% in A2, and 15%, 8%, 10% and 10% in A3 for 1000, 3000, 5000 and 7000 particles, respectively (Fig. 3(b)).

Figures 4(a) and (b) show the RMSEs at all noise levels for 1000 and 7000 particles, respectively. By integrating the WSMS, the RMSEs for 1000 particles are increased

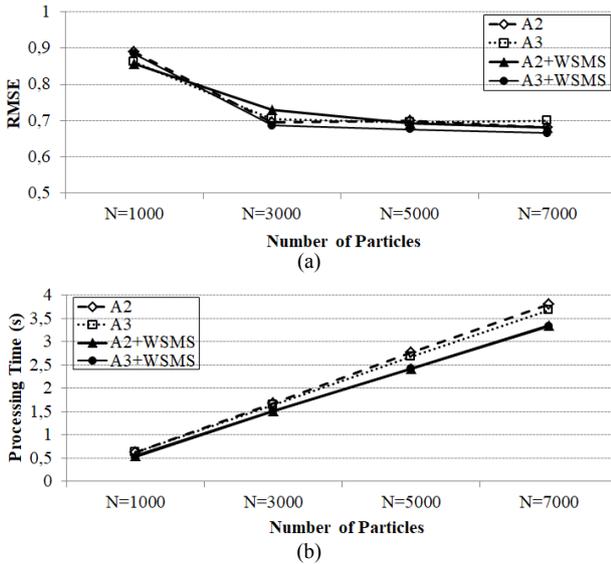


Fig. 3. Prediction error (a) and processing time (b) values for all particle numbers at noise level of  $\sigma^2 = 0.15$ .

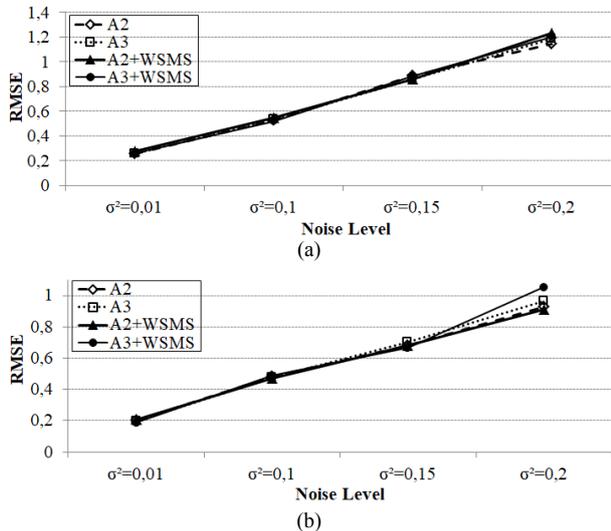


Fig. 4. RMSEs for all noise levels a)  $N = 1000$ , b)  $N = 7000$ .

by 4% in A2 and 0.5% in A3, on average (Fig. 4(a)). For 7000 particles, with the WSMS, the RMSE performance increased by 0.5% in A2, and decreased by 0.7% in A3, on average (Fig. 4(b)). As a result, the WSMS has no negative effect on the RMSE.

The processing times at all noise levels for the 1000 and 7000 particles are also given in Fig. 5(a) and (b), respectively. For both particle numbers, the WSMS increases the processing time performances of A2 and A3. With WSMS integration, for 1000 particles, processing times decrease by 20%, 5%, 12%, 7% in A2, and 21%, 9%, 15%, 8% in A3 for the noise levels of 0.01, 0.1, 0.15, 0.2, respectively (Fig. 5(a)). For 7000 particles, WSMS decreases processing times as rates of 26%, 9%, 12% and 4% in A2, and 26%, 38%, 10% and 11% in A3, for the noise variances of 0.01, 0.1, 0.15, 0.2, respectively (Fig. 5(b)). According to these results, WSMS provides significant performance improvement in processing time.

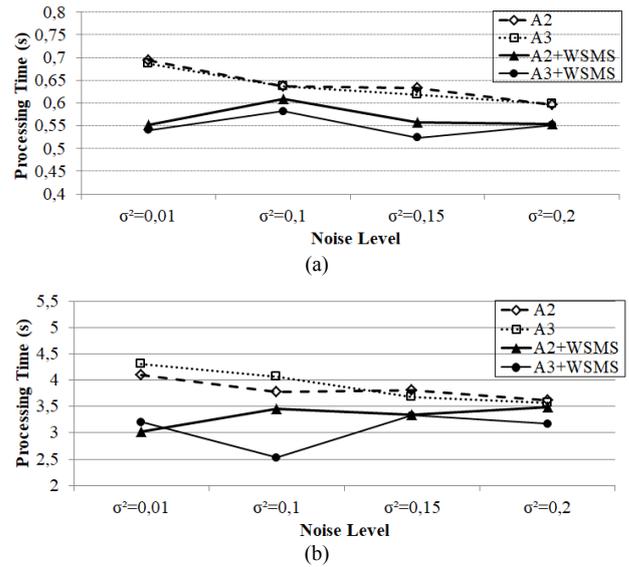


Fig. 5. Processing times for all noise levels a)  $N = 1000$ , b)  $N = 7000$ .

## 7. Discussion and Conclusion

In this study, a new motion model selection approach (WSMS), which is developed to use in the model prediction step of MMPFs, is proposed. The WSMS proceeds by selecting the most suitable model by weighting the model probabilities on a window. The calculation is only made for the selected model instead of all models in the system. Thus, the processing time is reduced by decreasing the time-consuming cost.

Looking at the results, it is seen that A1 is quite weak against noise when compared to others algorithms. In terms of processing time, the performance of A1 is the lowest in all cases. A2 and A3 demonstrated similar performances. The results show that, the WSMS is successful in reducing processing time. In A2, the WSMS decreased the processing time by up to rates of 44%, 41%, 27%, for S1, S2, S3, respectively. The using of A3 with WSMS significantly improved the processing time performance by up to rates of 51%, 45%, 38%, for S1, S2, S3, respectively. Furthermore, WSMS has no significant effect on the prediction error. As a result, A3+WSMS gives better results than A2+WSMS for time performance.

In point of the scenarios, the results obtained from S1 and S2 are similar in most cases. The results of S3 are slightly worse than those of others. Although the number of models in S2 and S3 is the same, the results of S2 are slightly better than those of S3. This situation can be interpreted that the performance of the proposed algorithm depends on the dynamic of the target motion being monitored (scenario). This also applies to all target tracking filters. It is a common result that the performance of the filter reduces as the scenario becomes difficult. In dynamic scenarios, RMSE can often fall below the threshold and, in this case, the proposed algorithm will calculate all models,

so that, the speed of the algorithm may decrease as the number of models increases. In this study, the results obtained for a challenging scenario are quite successful. As a result, the WSMS algorithm, whose activity is clearly seen on the processing time, is available in real-time tracking of moving targets.

## References

- [1] ZHANG, J., JIN, T., HE, Z., ZHOU, Z. M. A centralized processing framework for foliage penetration human tracking in multistatic radar. *Radioengineering*, 2016, vol. 25, no. 1, p. 98–105. DOI: 10.13164/re.2016.0098
- [2] TUYSUZ, B. A new target radar cross section based passive radar surveillance receiver positioning algorithm on real terrain maps. *Radioengineering*, 2018, vol. 27, no. 3, p. 891–898. DOI: 10.13164/re.2018.0891
- [3] CHEN, N. Q., JI, H. B., GAO, Y. C., YANG, D. New box particle filter with improved resampling method and extended inclusion volume criteria for multi-target tracking. *Radioengineering*, 2018, vol. 27, no. 3, p. 846–855. DOI: 10.13164/re.2018.0846
- [4] KALMAN, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 1960, vol. 82 (Series D), p. 35–45. DOI: 10.1115/1.3662552
- [5] KALANE, P. Target tracking using Kalman filter. *International Journal of Science and Technology*, 2012, vol. 2, no. 2, p. 16–24. ISSN (online): 2250-141X
- [6] TOLOEI, A., NIAZI, S. State estimation for tracking problems with nonlinear Kalman filter algorithms. *International Journal of Computer Applications*, 2014, vol. 98, no. 17, p. 30–36. DOI: 10.5120/17277-7708
- [7] GORDON, N. J., SALMOND, D. J., SMITH, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE-Proceedings-F*, 1993, vol. 140, no. 2, p. 107–113. DOI: 10.1049/ip-f-2.1993.0015
- [8] DOUCET, A., GODSILL, S., ANDIEU, C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Journal of Statistics and Computing*, 2000, vol. 10, no. 3, p. 197–208. DOI: 10.1023/A:1008935410038
- [9] RISTIC, B., ARULAMPALAM, M. S., GORDON, N. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House Publishers, February 2004. ISBN: 9781580536318
- [10] SOTO, A. Self adaptive particle filter. In *Proceedings of 19th International Joint Conferences on Artificial Intelligence (IJCAI-05)*. Edinburgh (Scotland), 2005, p. 1398–1403.
- [11] BAR-SHALOM, Y., FORTMANN, T. E. *Tracking and Data Association*. Academic Press, 1988. ISBN-10: 0120797607
- [12] MCGINNITY, S., IRWIN, G. W. Multiple model bootstrap filter for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 2000, vol. 36, no. 3, p. 1006–1012. DOI: 10.1109/7.869522
- [13] BOERS, Y., DRIESSEN, J. N. Interacting multiple model particle filter. *IEE Proceedings Radar, Sonar and Navigation*, 2003, vol. 150, no. 5, p. 344–349. DOI: 10.1049/ip-rsn:20030741
- [14] YILDIRIM, B., DEMIREKLER, M. Comparison of multiple model particle filter and interacting multiple model EKF in maneuvering target tracking. In *Proceedings of IEEE 16th Signal Processing and Communications Applications Conference*. Aydin (Turkey), 2008, p. 1–4. DOI: 10.1109/SIU.2008.4632647
- [15] YANG, Z., ZHAO, X. Mixed particle filter for maneuvering target tracking in clutter. In *Proceedings of International Conferences on Computational Intelligent for Modelling, Control and Automation (CIMCA2008), Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC 2008), Innovation in Software Engineering (ISE 2008)*. Vienna (Austria), 2008, p. 557–562. DOI: 10.1109/CIMCA.2008.219
- [16] CHEN, J. F., SHI, Z. G., HONG, S. H., CHEN, K. S. Grey prediction based particle filter for maneuvering target tracking. *Progress In Electromagnetics Research*, 2009, vol. 93, p. 237–254. DOI: 10.2528/PIER09042204
- [17] HONG, S., SHI, Z., CHEN, K. Easy-hardware-implementation MPMF for maneuvering target tracking: Algorithm and architecture. *Journal of Signal Processing Systems*, 2010, vol. 61, no. 3, p. 259–269. DOI: 10.1007/s11265-010-0450-4
- [18] WANG, X., HAN, C. Z. A multiple model particle filter for maneuvering target tracking based on composite sampling. *Acta Automatica Sinica*, 2013, vol. 39, no. 7, p. 1152–1156. DOI: 10.1016/S1874-1029(13)60069-5
- [19] BOLIC, M. *Architectures for Efficient Implementation of Particle Filters*. State University of New York at Stony Brook, NY, USA, 2004. ISBN: 0-496-08666-9
- [20] RONG, L. X., JILKOV, V. P. A survey of maneuvering target tracking: Dynamic models. In *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*. Orlando (USA), 2000, p. 4048–4022. DOI: 10.1117/12.391979
- [21] YANG, X., XING, K. Joint state and parameter estimation in particle filtering and stochastic optimization. In Dritsas, I. (ed.) *Stochastic Optimization—Seeing the Optimal for the Uncertain*. IntechOpen, 2011, p. 213–224. DOI: 10.5772/14658
- [22] SAEIDI, G., MONIRI, M. R. Bearings-only tracking of manoeuvring targets using multiple model variable rate particle filter with differential evolution. *Asia Pacific Journal of Energy and Environment*, 2014, vol. 1, no. 3, p. 200–214. DOI: 10.15590/apjee/2014/v1i3/54891
- [23] TURNER, L. A., SHERLOCK, C. H. *An Introduction to Particle Filtering*. [Online] Cited 2014-12-10. Available at: <https://zapdoc.tips/an-introduction-to-particle-filtering.html>

## About the Authors ...

**Murat Barkan UÇAR** graduated with M.Sc. degree in Electrical and Electronics Engineering from Başkent University in 2015. He is currently working in the fields of defense as an engineer. His interest areas are electronics design and test systems, radar systems.

**Derya YILMAZ** received the Ph.D. degree in Electronics and Computer Technology from Gazi University in 2008. She is currently an assistant professor. Her interest areas are signal processing, chaotic analysis, data transmission.