

Article

# A Novel FEM Based T-S Fuzzy Particle Filtering for Bearings-Only Maneuvering Target Tracking

Xiaoli Wang <sup>1,2</sup>, Liangqun Li <sup>1,2,\*</sup> and Weixin Xie <sup>1,2</sup>

<sup>1</sup> Automatic Target Recognition (ATR) Key Laboratory, Shenzhen University, Shenzhen 518060, China; xlwang@szu.edu.cn (X.W.); wxxie@szu.edu.cn (W.X.)

<sup>2</sup> Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, China

\* Correspondence: lqli@szu.edu.cn; Tel.: +86-0755-26732055

Received: 30 March 2019; Accepted: 10 May 2019; Published: 13 May 2019



**Abstract:** In this paper, we propose a novel fuzzy expectation maximization (FEM) based Takagi-Sugeno (T-S) fuzzy particle filtering (FEMTS-PF) algorithm for a passive sensor system. In order to incorporate target spatial-temporal information into particle filtering, we introduce a T-S fuzzy modeling algorithm, in which an improved FEM approach is proposed to adaptively identify the premise parameters, and the model probability is adjusted by the premise membership functions. In the proposed FEM, the fuzzy parameter is derived by the fuzzy C-regressive model clustering method based on entropy and spatial-temporal characteristics, which can avoid the subjective influence caused by the artificial setting of the initial value when compared to the traditional FEM. Furthermore, using the proposed T-S fuzzy model, the algorithm samples particles, which can effectively reduce the particle degradation phenomenon and the parallel filtering, can realize the real-time performance of the algorithm. Finally, the results of the proposed algorithm are evaluated and compared to several existing filtering algorithms through a series of Monte Carlo simulations. The simulation results demonstrate that the proposed algorithm is more precise, robust and that it even has a faster convergence rate than the interacting multiple model unscented Kalman filter (IMMUKF), interacting multiple model extended Kalman filter (IMMEKF) and interacting multiple model Rao-Blackwellized particle filter (IMMRBPF).

**Keywords:** maneuvering target tracking; particle filtering; T-S fuzzy modeling; fuzzy expectation maximization

## 1. Introduction

The motion state (position, velocity, acceleration, etc.) estimation of the target in the system refers to the measurement information obtained by the measuring device, and establishes a reasonable and accurate dynamic model by using modern signal processing techniques such as a stochastic process, estimation and detection theory, and a filtering algorithm. When dealing with linear systems, the Kalman filter [1] theory is the optimal linear Bayesian estimation algorithm. Subsequently, some extended Kalman filters (EKF) [2,3] were proposed to further apply the Kalman filter (KF) theory to the nonlinear system. The basic idea of the EKF is to linearize the nonlinear system and then perform KF, so EKF is a suboptimal filter. Then, a second-order generalized Kalman filter [4] method was proposed and applied to further improve the estimation performance of KF for nonlinear systems. It took into account the second-order terms of the Taylor series expansion. Therefore, the estimation error caused by linearization was reduced, and the filtering precision of nonlinear systems was improved, but the computational complexity was greatly increased. Therefore, it was not widely used in practice. Otherwise, with the aggravation of the nonlinearity of the dynamic system, the performance of EKF

decreases sharply. In order to solve this problem, Julier and Uhlmann proposed the unscented Kalman filter (UKF) [5] algorithm. Unlike the EKF, the UKF [6–8] approximates the distribution of state random variables by selecting a small number of sample points, using the nonlinear model directly. However, since EKF and UKF have no way of getting rid of the system's Gaussian constraints, the processing effect on non-Gaussian systems is not good.

In order to deal with the problem of state estimation in the case of nonlinear and non-Gaussian systems, particle filtering (PF) [9] had been proposed. PF is a non-parametric Monte Carlo simulation method used to achieve a recursive Bayesian filtering. With the development of computing ability and statistical theory, the PF algorithm has been developed rapidly. Nowadays, the algorithm has been successfully applied in many fields. Isard et al. [10] introduced PF into the target tracking of image sequences, which made PF become a hot topic in the fields of target tracking, machine learning and robot localization. Then, scholars put forward many improved algorithms based on the PF algorithm, such as Unscented PF (UPF) [11], Rao-Blackwellized PF (RBPF) [12], Auxiliary PF (APF) [13], Regular PF (RPF) [14], MCMC PF [15] and Gaussian PF (GPF) [16,17].

Another significant feature in the nonlinear non-Gaussian system is the uncertainty of its motion pattern, so the traditional single-model method has difficulty achieving a good tracking performance. In view of this, some scholars have applied multiple PFs or a PF and multiple model method [18,19] to the system, and have obtained a good performance. Bando et al. [20] proposed a switching particle filter, which allowed robust and accurate visual tracking under typical circumstances of real-time visual tracking. This scheme switched two complementary sampling algorithms, condensation and auxiliary particle filter, in an on-line fashion based on the confidence of the filtered state of the visual target. Meshgi et al. [21] proposed an occlusion-aware particle filter framework that employs a probabilistic model, with a latent variable representing an occlusion flag, which prevented the loss of the target through the prediction of emerging occlusions, updated the target template by shifting relevant information, expanded the search area for an occluded target, and granted a quick recovery of the target after an occlusion. Martino et al. [22] introduced two novel Markov Chain Monte Carlo (MCMC) techniques based on group importance sampling, where the information contained in different sets of weighted samples was compressed by using only one (properly selected, however), particle, and one suitable weight. Otherwise, an interacting multiple model Bernoulli PF (IMMBPF) [23] algorithm for maneuvering target tracking was simply combined with the particle implementation of IMM and PF. The introduction of model information into the particle sampling process will lead to a reduction in the number of particles used to approach the current real state of the model, and the particles will interact with each other in each recursive model, which had the disadvantage of too much computation. To improve the effectiveness of single sampling particles in IMMBPF for real target states and a model approximation, Yang et al. [24] proposed an improved multiple model Bernoulli particle filter (MMBPF), in which the number of particles in each model was pre-selected. Furthermore, the particles in the model did not need to interact with each other, which reduced the computational load. The model probability was calculated from the model likelihood function, and the particle degradation of the small probability model was avoided without changing the Markov property of the model. Additionally, to meet the requirements of modern radar maneuvering target tracking systems and to remedy the defects of an interacting multiple model based on PF, a non-interacting multiple model (NIMM) and an enhanced particle swarm optimized particle filter (EPSO-PF) were proposed in [25]. NIMM was used to figure out the index of particles to avoid the high computing complexity resulting from particle interaction, and EPSO-PF not only improves the equation of a particle update through the rules through which individuals develop a group understanding but it also enhances particle diversity and accuracy through the small variation probability of the superior velocity. Additionally, the random assignment of an inferior velocity was capable of upgrading the filter efficiency. Instead of resorting to model selection, Urteaga et al. [26] fused the information from the considered models within the proposed SMC method, and achieved the goal by dynamically adjusting the resampling step according to the posterior predictive power of each model, which was

updated sequentially as we observed more data. Martino et al. [27] designed an interacting parallel sequential Monte Carlo scheme for inference in state space models and in an online model selection. The parallel particle filters collaborated to provide a global efficient estimate of the hidden states and an approximation of the probability of the models, given the received measurements. For a static parameter estimation of the model, Carvalho et al. [28] extended existing particle methods by incorporating the estimation of static parameters via a fully-adapted filter that utilizes conditional sufficient statistics for parameters and states as particles. For the sake of a better tracking performance, it may be necessary to use a large set of models, but this will inevitably increase the computational complexity, which is one of the drawbacks of the multi-model filtering method. At the same time, unnecessary competition from too many models may result in a decline in performance. Therefore, it is of great value and practical significance to seek a more effective modeling approach.

As a mathematical tool to deal with the fuzzy phenomenon, fuzzy mathematics [29] can not only solve the uncertainty caused by the randomness of the system, but can also deal with the fuzzy uncertainty caused by the uncertainty of the extension of the concept of the system. It can be used to model qualitative, fuzzy or uncertain ones in the form of natural language. Fuzzy mathematics can describe different uncertain information in simple fuzzy language. Recently, the study of fuzzy particle filtering has become one of the research hotspots of complex nonlinear non-Gaussian systems. Widynski et al. [30] proposed a particle filtering algorithm with integrated fuzzy spatial information, which introduced the target spatial information through the fuzzy probability and improved the accuracy of the sampling particle. Li et al. [31,32] proposed a fuzzy orthogonal particle filter, which approximated the predicted probability density function and the posterior probability density function by introducing a set of positive intersection probabilities based on the Gauss-Hermite rule. The advantage of a fuzzy logic particle filter is that it does not need to know the statistical model of the process in advance. In addition, it does not need any maneuvering detector, even when tracking high-performance targets, so the computational complexity is low, and appropriate fuzzy overlapping sets will be closer to the real motion model. The theory of fuzzy models is a general concept proposed in the last decade. Among the various types of fuzzy models, there is a very important T-S [33,34] fuzzy model. Due to its special rules of consequence-structure and its success in function approximation, it has been widely used recently. Therefore, this paper constructs a general T-S fuzzy model framework based on spatial-temporal semantic information, and uses multiple linear models to get a more accurate target motion model, which makes the state estimation of the particle filtering algorithm more accurate.

In this paper, for the nonlinear non-Gaussian problem in the passive sensor system, a T-S fuzzy modeling particle filtering algorithm, based on improved fuzzy expectation maximization, is proposed. The main contributions are as follows: (1) A T-S fuzzy model, based on spatial-temporal information, is proposed for the uncertain modeling of a target dynamic model, in which spatial-temporal feature information is represented by multiple semantic fuzzy sets. Then, the general T-S fuzzy model framework is constructed, approximating the dynamic model with a high precision. (2) An improved fuzzy expectation maximization method with fuzzy C-regressive model clustering based on entropy and integrated spatial-temporal information is proposed for the premise parameter identification in the T-S fuzzy model. In addition, the model probability is adjusted adaptively through the premise membership functions. (3) The importance density function is constructed by using the proposed T-S fuzzy model, which contains abundant prior knowledge and the latest measurement information; thus, it can effectively approximate the true posterior probability density function and improve the diversity of particles.

The rest of this paper is organized as follows. Section 2 presents the proposed T-S fuzzy modeling particle filtering algorithm. Section 3 describes the simulation results that compare the performances of all of the algorithms. Finally, some conclusions of the proposed algorithm are given in Section 4.

## 2. The Proposed Algorithm

According to the constrained Bayesian principle [35], the nonlinear discrete system model is considered.

$$x_k = f_k(x_{k-1}) + e_{k-1} \quad (1)$$

$$\theta_k = h_{\theta_k}(x_k) \quad (2)$$

$$\theta_k = h_{\theta_k}(x_k) \quad (3)$$

where  $k \in N$  denotes the discrete time,  $f_k$ ,  $h_k$  and  $h_{\theta_k}$  denote some appropriate nonlinear functions, and  $x_k$  is a state vector.  $z_k$  is a measurement vector.  $e_k$  is the process noise, with zero mean and covariance  $Q_{e_k}$ , and  $v_k$  is the measurement noise, with zero mean and covariance  $R_{v_k}$ .  $\theta_k$  denotes the spatial-temporal feature information.

As is well known, the spatial relationship cannot be directly applied to the particle filtering algorithm. In this paper, the target feature information is used to construct the importance density function through the proposed T-S fuzzy model, so that the spatial relation is introduced into the particle filtering algorithm. In the particle filtering framework, the probability density function estimation is divided into two phases: time update and state update. The prediction state density is calculated from the prior probability density function (PDF) by using the Chapman-Kolmogorov equation:

$$p(x_k|z_{1:k-1}, \theta_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1}, \theta_{1:k-1})dx_{k-1} \quad (4)$$

where  $p(x_k|x_{k-1})$  is the priori transfer density function, and the measurement update is computed by a Bayesian formula:

$$p(x_k|z_{1:k}, \theta_{1:k}) = \frac{p(z_k|x_k)p(\theta_k|x_k)p(x_k|z_{1:k-1}, \theta_{1:k-1})}{p(z_k|z_{1:k-1}, \theta_k)p(\theta_k|\theta_{1:k-1})} \quad (5)$$

where  $p(z_k|x_k)$  is the likelihood function, and  $p(\theta_k|x_k)$  is the characteristic likelihood function.

Suppose that  $\{x_{k,j}\}_{j=1}^M$  represents the particles at time  $k$ , where  $M$  is the number of particles. Under the constraint of spatial-temporal feature  $\theta_{1:k}^m$ ,  $m = 1, \dots, G$ , and  $\sum_j^M \omega_{k,j} = 1$ ,  $\omega_{k,j}$  denotes the weight of the particle; consequently,

$$\omega_{k,j} \propto \omega_{k-1,j} \frac{p(z_k|x_{k,j})p(\theta_k|x_k)p(x_{k,j}|x_{k-1,j})}{q(x_{k,j}|x_{k-1,j}, z_k)} \quad (6)$$

The  $p(\theta_k|x_k)$  not only contains abundant prior knowledge, but also incorporates a higher-level spatial-temporal feature as well as measurement information. If we combine it with a prior probability density function  $p(x_{k,j}|x_{k-1,j})$  to form an importance density function  $q(x_{k,j}|x_{k-1,j}, z_k)$ , it will reduce the degradation of the particles.

$$q(x_{k,j}|x_{k-1,j}, z_k) = p(\theta_k|x_k)p(x_{k,j}|x_{k-1,j}) \quad (7)$$

### 2.1. Construction of Importance Density Function

The selection of an importance density function is a very important step. Traditional particle filtering algorithms usually use priori probability as the importance density function, but generally the prior probability does not fully consider the real-time effects of the current measurement. It is easy to cause particle degradation. In order to solve this problem, according to the principle of constrained particle filtering, the importance density function is constructed, as in Equation (7), not only reducing the particle degradation phenomenon, but also improving the stability of the system. The block diagram of the T-S fuzzy modeling method is shown in Figure 1.

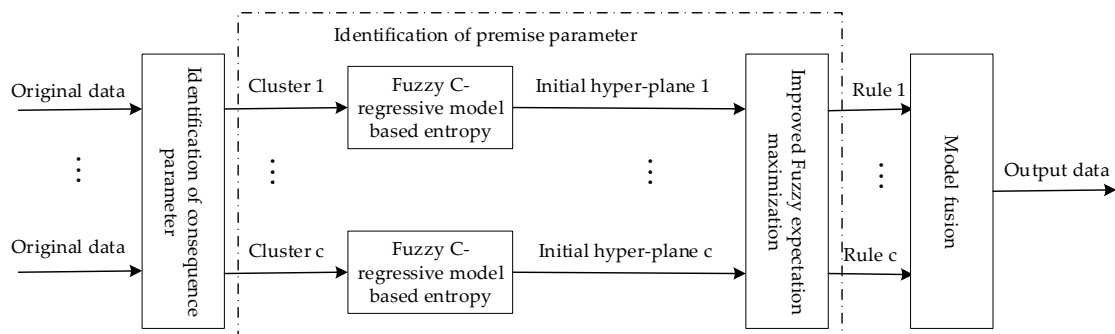


Figure 1. Block diagram of the T-S fuzzy modeling method.

### 2.1.1. T-S Fuzzy Semantic Modeling

A detailed T-S fuzzy model can be found in references [36,37], and it is briefly described in this section. In general, the T-S fuzzy model can be described by  $N_f$  fuzzy linear models:

Model  $i$ : IF  $\theta_k^1$  is  $A_k^{i,1}$ ,  $\theta_k^2$  is  $A_k^{i,2}$ , ...,  $\theta_k^G$  is  $A_k^{i,G}$ , then:

$$x_k^i = \Phi_{k-1}^i x_{k-1}^i + e_{k-1}^i \tag{8}$$

$$z_k^i = H_k^i x_k^i + v_k^i, i = 1, 2, \dots, N_f \tag{9}$$

where  $\theta_k = [\theta_k^1, \theta_k^2, \dots, \theta_k^G]$  denotes the premise parameters of the model,  $A_k^{i,G}$  denotes the fuzzy set of the  $G$ th premise parameter in the model  $i$ , and  $\Phi_{k-1}^i$  and  $H_k^i$  denote the state transition matrix and the measurement matrix, respectively. The consequent part is iteratively updated by the strong tracking algorithm [38], so the global fuzzy model can be represented as follows:

$$x_k = \sum_{i=1}^{N_f} \mu_k^i(\theta_k) (\Phi_{k-1}^i x_{k-1}^i + e_{k-1}^i) \tag{10}$$

$$z_k = \sum_{i=1}^{N_f} \mu_k^i(\theta_k) H_k^i x_k^i + v_k \tag{11}$$

where  $\mu_k^i(\theta_k)$  denotes the model probability, which is calculated as follows:

$$\mu_k^i(\theta_k) = \frac{\bar{\mu}_k^i(\theta_k)}{\sum_{j=1}^{N_f} \bar{\mu}_k^j(\theta_k)}, \bar{\mu}_k^i(\theta_k) = \prod_{m=1}^G p_{A_k^{i,m}}(\theta_k^m) \tag{12}$$

where  $p_{A_k^{i,m}}(\theta_k^m) \in [0, 1]$  denotes the membership function of the premise parameter  $\theta_k^m$  belonging to model set  $A_k^{i,m}$  in the fuzzy linear model  $i$ .

In general, the fuzzy membership function of the model sets  $A_k^{i,m}$  is designed as the following Gaussian type function:

$$p_{A_k^{i,m}}(\theta_k^m) = \exp\left\{-\frac{1}{2}\left(\frac{\theta_k^m - \tau_k^{i,m}}{\sigma_k^{i,m}}\right)^2\right\} \tag{13}$$

where  $\tau_k^{i,m}$  and  $\sigma_k^{i,m}$  denote the mean and standard deviation of the membership function of the premise parameter  $m$  in the model  $i$ , respectively.

### 2.1.2. Premise Parameter Identification Based on Improved Fuzzy Expectation Maximization

Inspired by the Gaussian mixed model (GMM) [39], in which the expectation maximization (EM) algorithm was used to fit its parameters, in the process of constructing the T-S model we use the EM algorithm, a general method for a maximum likelihood estimation of the model parameters, to identify the premise parameters. According to the idea of the EM in the GMM, the likelihood function of the parameter model is constructed as follows:

$$L(\theta_k) = \sum_{l=1}^C \log \sum_{i=1}^{N_f} \pi_k^i p_{A_k^{i,m}}(\theta_k^m) \quad (14)$$

where  $\pi_k^i$  is a hidden feature, and  $p_{A_k^{i,m}}(\theta_k^m)$  is defined in (13).

Given the appropriate initial assumptions, the noise samples and normal samples participate in the iterative process equally, which will undoubtedly have a negative impact on the accuracy and convergence rate of the parameter estimation. To solve this problem, the knowledge of the fuzzy theory was introduced in the iterative process of the EM algorithm, and a fuzzy expectation maximization (FEM) algorithm [40] was proposed. The fuzzy theory was introduced into the EM algorithm to reduce the influence of noise by making different samples play different roles in the iterative process. Simulations showed that this limitation can better realize the parameter estimation function of the EM algorithm and accelerate the convergence speed of the algorithm.

**Theorem 1 (Jensen inequality [41]):** let  $f(x)$  be a convex function, for the random variable  $X$ , then

$$f(EX) \geq E[f(X)]$$

If and only if  $X = EX$  (That is,  $X$  is constant), equal sign holds by probability 1, where  $E[\cdot]$  is the expectation operation. For example, the logarithmic function is an upper convex function, then

$$\log(EX) \geq E[\log(X)]$$

In accordance with the traditional FEM, the non-negative fuzzy parameter  $u_{k,l}^i$  is introduced into Equation (14), where  $u_{k,l}^i$  is the fuzzy membership degree between the  $l$ th measurement and the  $i$ th rule at time  $k$ , which satisfies  $\sum_{i=1}^{N_f} u_{k,l}^i = 1$ ,  $k = 1, 2, \dots, N$ , and the Jensen inequality (Theorem 1) is used to obtain the likelihood function of the proposed FEM approach:

$$\begin{aligned} L(\tau_k^{i,m}, \sigma_k^{i,m}) &= \sum_{l=1}^C \log \sum_{i=1}^{N_f} u_{k,l}^i \frac{\pi_k^i p_{A_k^{i,m}}(\theta_k^m)}{u_{k,l}^i} \\ &= \sum_{l=1}^C \log E_{u_{k,l}^i} \frac{\pi_k^i p_{A_k^{i,m}}(\theta_k^m)}{u_{k,l}^i} \geq \sum_{l=1}^C E_{u_{k,l}^i} \left[ \log \frac{\pi_k^i p_{A_k^{i,m}}(\theta_k^m)}{u_{k,l}^i} \right] \end{aligned} \quad (15)$$

In the EM algorithm, the right side of the inequality in Equation (15) is the lower bound that needs to be optimized:

$$B(\tau_k^{i,m}, \sigma_k^{i,m}) = \sum_{l=1}^C E_{u_{k,l}^i} \left[ \log \frac{\pi_k^i p_{A_k^{i,m}}(\theta_k^m)}{u_{k,l}^i} \right] = \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log \frac{\pi_k^i p_{A_k^{i,m}}(\theta_k^m)}{u_{k,l}^i} \quad (16)$$

Consider the constraint  $\sum_{i=1}^{N_f} u_{k,l}^i = 1$ ,  $\sum_{i=1}^{N_f} \pi_k^i = 1$ , and introduce the Lagrangian term to obtain the constrained optimization likelihood objective function:

$$B(\tau_k^{i,m}, \sigma_k^{i,m}) = \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log \pi_k^i + \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log p_{A_k^{i,m}}(\theta_k^m) - \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log u_{k,l}^i - \lambda_1 \left( \sum_{i=1}^{N_f} \pi_k^i - 1 \right) - \lambda_2 \left( \sum_{i=1}^{N_f} u_{k,l}^i - 1 \right) \quad (17)$$

It is known from Equation (13) that the membership function of the premise parameter obeys the Gaussian distribution.

$$B(\tau_k^{i,m}, \sigma_k^{i,m}) = -\frac{1}{2} \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log((2\pi)^d |\sigma_k^{i,m}|) - \frac{1}{2} \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i (\theta_k^m - \tau_k^{i,m})^T (\sigma_k^{i,m})^{-1} (\theta_k^m - \tau_k^{i,m}) + \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log \pi_k^i - \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log u_{k,l}^i - \lambda_1 \left( \sum_{i=1}^{N_f} \pi_k^i - 1 \right) - \lambda_2 \left( \sum_{i=1}^{N_f} u_{k,l}^i - 1 \right) \quad (18)$$

According to the gradient descent method, the mean and standard deviation of the premise membership functions are obtained (The specific derivation process is shown in Appendix A).

$$\tau_k^{i,m} = \frac{\sum_{l=1}^C u_{k,l}^i \theta_k^m}{\sum_{l=1}^C u_{k,l}^i} \quad (19)$$

$$\sigma_k^{i,m} = \frac{\sum_{l=1}^C u_{k,l}^i (\theta_k^m - \tau_k^{i,m}) (\theta_k^m - \tau_k^{i,m})^T}{\sum_{l=1}^C u_{k,l}^i} \quad (20)$$

It can be seen from Equations (19) and (20) that the identification of the premise parameters is closely related to the non-negative fuzzy parameters  $u_{k,l}^i$ . Therefore, the design of the  $u_{k,l}^i$  is the key problem to be solved in the next step. In the traditional method, the fuzzy parameter was set by manual initialization, which will have certain errors and subjective factors. To avoid the influence, we use the fuzzy C-recessive model (FCRM) clustering algorithm, based on spatial-temporal information and entropy adjustment to obtain the fuzzy parameter. Therefore, the premise parameter membership functions can fully reflect the motion information of the target, and avoid the unnecessary influence caused by the artificial setting of the initial value. Suppose that  $z_k = \{z_{k,l}\}_{l=1}^C$  is a measurement set and  $\hat{z}_k = \{\hat{z}_k^i\}_{i=1}^{N_f}$  is a predictive measurement set,  $z_{k,l}$  denotes the  $l$ th measurement, and  $\hat{z}_k^i$  denotes the predictive measurement based on the  $i$ th fuzzy rule at time  $k$ . On the basis of the traditional FCRM [33], the weighted entropy is introduced to balance the membership degree, which is called the entropy adjustment method. Meanwhile, the target feature information reflects the target motion trend in real time. Therefore, combining the spatial constraint information  $\theta_k$ , the objective function of the entropy adjustment method is defined as follows:

$$Y(u_{k,l}^i) = -\sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \ln(u_{k,l}^i) - \beta \cdot \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right] + \sum_{l=1}^C \lambda_k \left( \sum_{i=1}^{N_f} u_{k,l}^i - 1 \right) \quad (21)$$

where  $\lambda_k$  is the Lagrange multiplier vector,  $\beta$  is a constant,  $\omega_k^{i,m}$  is the weight of the feature  $m$  in the model  $i$ .  $(D_{k,l}^i)^2$  denotes the dissimilarity measure function between the  $l$ th measurement and the output predictive measurement of the  $i$ th fuzzy rule, which is defined as:

$$(D_{k,l}^i)^2 = \frac{1}{p_i(z_{k,l}|\hat{x}_k^i)} \quad (22)$$

where  $p(z_{k,l}|\hat{x}_k^i)$  is called the likelihood function of the measurement  $z_{k,l}$  given the target state  $\hat{x}_k^i$ .

According to the Lagrangian multiplier, the update of the fuzzy membership degree  $u_{k,l}^i$  between the  $l$ th measurement and the  $i$ th fuzzy model is obtained (The specific derivation process is shown in Appendix B).

$$u_{k,l}^i = \frac{\exp\left(-\beta\left[(D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m\right]\right)}{\sum_{q=1}^{N_f} \left(\exp\left(-\beta\left[(D_{k,l}^q)^2 + \sum_{m=1}^G \omega_k^{q,m} \theta_k^m\right]\right)\right)} \quad (23)$$

The improved fuzzy expectation maximization algorithm used in the identification of the premise parameters is shown in Algorithm 1.

---

#### Algorithm 1 Premise Parameter Identification-Improved Fuzzy Expectation Maximization

---

1. **Initializations:** Define the initial premise parameter  $(\tau, \sigma)^{(0)}$ , the stop criterion  $\zeta$ , the parametric model likelihood function  $L(\theta)$ , and the set  $i = 0$ .
  2. **Do**  $i = i + 1$ 
    - Calculate the non-negative fuzzy membership degree by Equation (23) and introduce it into the likelihood function  $L(\theta)$  of Equation (14), and rewrite the likelihood function into Equation (15).
    - The lower bound function  $B(\theta)$  is obtained from Jensen's inequality.
    - E-step: Calculate  $B((\tau, \sigma), (\tau, \sigma)^{(i-1)})$  by Equations (16)–(18).
    - M-step:  $(\tau, \sigma)^{(i)} = \operatorname{argmax} B((\tau, \sigma), (\tau, \sigma)^{(i-1)})$ , and the  $(\tau, \sigma)^{(i)}$  is obtained by Equations (19) and (20).
  3. **Until**  $B((\tau, \sigma)^{(i+1)}, (\tau, \sigma)^{(i)}) - B((\tau, \sigma)^{(i)}, (\tau, \sigma)^{(i-1)}) \leq \zeta$
  4. **Return**  $(\tau, \sigma) = (\tau, \sigma)^{(i+1)}$
  5. **Finish**
- 

On the basis of the model fusion method in the traditional multiple model algorithm, the premise membership function is identified by Equations (19) and (20), the model probability is obtained by Equation (12), and the  $\hat{x}_k^i$  and  $P_k^i$  are obtained by strong tracking [38]. Therefore, the state and covariance updates of the proposed T-S fuzzy model are as follows:

$$\tilde{x}_k = \sum_{i=1}^{N_f} \mu_k^i \hat{x}_k^i \quad (24)$$

$$\tilde{P}_k = \sum_{i=1}^{N_f} \mu_k^i [P_k^i + (\tilde{x}_k - \hat{x}_k^i)(\tilde{x}_k - \hat{x}_k^i)^T] \quad (25)$$

For each particle, the state and covariance are  $\tilde{x}_{k,j}$  and  $\tilde{P}_{k,j}$  on the basis of Equations (24) and (25), so the importance density function of the proposed algorithm is defined as:

$$q(x_{k,j}|x_{k-1,j}, z_k) = N(\tilde{x}_{k,j}, \tilde{P}_{k,j}) \quad (26)$$



## 2.2. Summary of the Algorithm

Based on the above analysis, the T-S fuzzy modeling particle filtering algorithm can be summarized as Algorithm 2:

---

### Algorithm 2 Fuzzy Expectation Maximization-Based T-S Fuzzy Particle Filtering Algorithm

---

1. **Initializations:** Set that the number of fuzzy rules is  $N_f$ . The particles  $(x_{0:k-1}^j)_{j=1}^M$  are drawn from the priori probability density function  $p(x_0)$ , and the number of particles is set to  $M$ .

2. **For**  $k = 1, 2, \dots$

(a) T-S fuzzy model parameter identification

- Consequence parameter identification: The strong tracking algorithm [38] is used to identify the consequence parameters.
- Premise parameter identification: As shown in Algorithm 1.

(b) Model probability update and fusion: The model probability is updated by Equation (12), and the model fusion is carried out by Equation (10).

(c) Construct the importance density function and sample: Draw particles  $\{x_{k,j}\}_{j=1}^M$  from Equation (26).

(d) Calculate and normalize the particle weight: The particle weight is calculated by Equation (6) and normalized as follows:

$$\psi_{k,j} = \omega_{k,j} / \sum_{j=1}^M \omega_{k,j}$$

(e) State and covariance estimation:

- Output state:  $\hat{x}_k = \sum_{j=1}^M \psi_{k,j} x_{k,j}$
  - Output covariance:  $P_k = \sum_{j=1}^M \psi_{k,j} [\tilde{P}_{k,j} + (\hat{x}_k - x_{k,j})(\hat{x}_k - x_{k,j})^T]$
- 

## 2.3. Discussion

**Summary:** In the design of the proposed algorithm, to improve the convergence performance of the T-S fuzzy model in a passive sensor system, and to reduce the approximation error, the FEM was introduced to identify the premise parameters that can capture the higher-order statistical parameters from a small number of samples. To identify the consequent parameters of the T-S fuzzy model, the strong tracking estimator was used. In particular, for the maneuvering target tracking, the model probability, which was closely connected to the true motion model, was adaptively updated by the premise membership functions. Moreover, the samples were drawn from the proposed T-S fuzzy model, which had abundant priori information and the latest measurement, and which can reduce the degradation of the particles.

**Comparison:** All of the samples were used to train the fuzzy model parameters in the conventional T-S fuzzy model described in [33,34], after which the trained fuzzy model was used to classify or estimate the model state. In our proposed algorithm, the parameters of the T-S fuzzy model were updated by using the recursive mechanism of the algorithm, which required the fast convergence. The simulations showed that the proposed algorithm can not only achieve fast convergence, but that it can also accurately estimate the state.

Additionally, in the traditional FEM, the fuzzy parameter was set by manual initialization, but in our proposed algorithm the fuzzy parameter was obtained through the FCRM, based on entropy and the spatial-temporal characteristic information. The simulations showed that not only can the proposed FEM realize the parameter estimation function of the EM algorithm and accelerate the convergence speed, but it can also avoid the subjective influence caused by the artificial setting of the initial value.

### 3. Simulation Results and Analysis

In order to compare the performance of the FEMTS-PF with those of IMMUKF, IMMEKF, IMMRBPF [42], fuzzy PF (FPF) and traditional FEMTS-PF, two examples are employed in this section. In Section 3.1., a bearings-only tracking example is adopted. In Section 3.2., the tracking results of a maneuvering target in a sparse environment are analyzed. 100 Monte Carlo simulations are carried out in all of the experiments, and the number of particles is 200. The proposed algorithm is implemented on Matlab R2017a on the CPU3.6 and 8 GB memory computers of Inter (R) Core (TM) i5-6500.

The root-mean square error (RMSE) is used as the performance index (PI), which is defined as:

$$RMSE = \sqrt{\frac{\sum_{k=1}^D (x_k - \hat{x}_k)^2}{D}} \tag{27}$$

where  $D$  is the number of Monte Carlo (MC) simulations.

#### 3.1. An Example of Bearings-Only Tracking (BOT)

The state and measurement equations of the target in the proposed algorithm are as follows.

$$x_k^i = \Phi_{k-1}^i(T, \omega_i)x_{k-1}^i + e_{k-1} \tag{28}$$

$$z_k^i = H_k^i x_k^i + v_k, i = 1, 2, \dots, N_f \tag{29}$$

where  $N_f$  denotes the number of fuzzy rules,  $x_k = [x_k, x'_k, y_k, y'_k]^T$  denotes the state vector,  $x_k$  denotes the  $x$ -coordinate and  $y_k$  denotes the  $y$ -coordinate of the target.  $x'_k$  and  $y'_k$  denote the corresponding velocities. The process noise  $e_k$  is assumed to be Gaussian noise with zero mean and standard deviation  $\sigma_{i,e}$  (The values are shown in Table 1). The measurement noise  $v_k$  is assumed to be non-Gaussian noise ( $R = \left[ \frac{N(0,\sigma)+N(0,\sigma)}{2}, \frac{N(0,\sigma)+N(0,\sigma)}{2} \right]^T, \sigma = 0.001$ ). The initial state describes the target's initial position and velocity. The a priori probability density of the state  $x_0$  is assumed to be  $x_0 \sim N(\hat{x}_{0|0}, \hat{P}_{0|0})$ , where  $x_{0|0} = [1 \text{ km}, 0.15 \text{ km/s}, 6 \text{ km}, 0.26 \text{ km/s}]^T$ ,  $P_{0|0} = [\sigma^2 \ 0 \ 0 \ 0; 0 \ \sigma^2 \ 0 \ 0; 0 \ 0 \ \sigma^2 \ 0; 0 \ 0 \ 0 \ \sigma^2]$ . In this paper, the innovation and the heading angle difference are selected as the semantic information, which can effectively reflect the moving state of the target. For example, the innovation can indicate the suitability of the target motion model; when it is large, the target motion model is explained to be less consistent with the current motion state, and thus we can adjust the model probability to get a more accurate model of motion. The state transition matrix  $\Phi_{k-1}^i$  is defined as follows:

$$\Phi_{k-1}^i = \begin{bmatrix} 1 & \frac{\sin \omega_i T}{\omega_i} & 0 & -\frac{1-\cos \omega_i T}{\omega_i} \\ 0 & \cos \omega_i T & 0 & -\sin \omega_i T \\ 0 & \frac{1-\cos \omega_i T}{\omega_i} & 1 & \frac{\sin \omega_i T}{\omega_i} \\ 0 & \sin \omega_i T & 0 & \cos \omega_i T \end{bmatrix}$$

**Table 1.** The  $\omega_i$  and the  $\sigma_{i,e}$  for different  $\Delta\hat{\theta}_k$  and  $\Delta v_k$ .

$\Delta\hat{\theta}_k$ (rad)	$\Delta v_k$ (km)		
	Negative Large (NL)	Small (S)	Positive Large (PL)
Small (S)	-0.0324, 0.15	0, 0.015	0.0324, 0.015
Large (L)	-0.0124, 0.005	0, 0.015	0.0124, 0.015

Table 1 shows the turning rate  $\omega_i$  and standard deviation  $\sigma_{i,e}$  of the process noise in the T-S fuzzy models;  $\omega_i = 0$  denotes the constant turn (CT) model. The process noise covariance is as follows:

$$Q_{e_{k-1}^i} = \begin{bmatrix} \frac{2(\omega_i T - \sin \omega_i T)}{\omega_i^3} & \frac{1 - \cos \omega_i T}{\omega_i^2} & 0 & \frac{\omega_i T - \cos \omega_i T}{\omega_i^2} \\ \frac{1 - \cos \omega_i T}{\omega_i^2} & T & \frac{\omega_i T - \cos \omega_i T}{\omega_i^2} & 0 \\ 0 & -\frac{\omega_i T - \cos \omega_i T}{\omega_i^2} & \frac{2(\omega_i T - \sin \omega_i T)}{\omega_i^3} & \frac{1 - \cos \omega_i T}{\omega_i^2} \\ \frac{\omega_i T - \cos \omega_i T}{\omega_i^2} & 0 & \frac{1 - \cos \omega_i T}{\omega_i^2} & T \end{bmatrix} \cdot \sigma_{i,e}^2$$

Two passive sensors are placed in the (0, 5 km) and (0, -5 km). The measurement function:

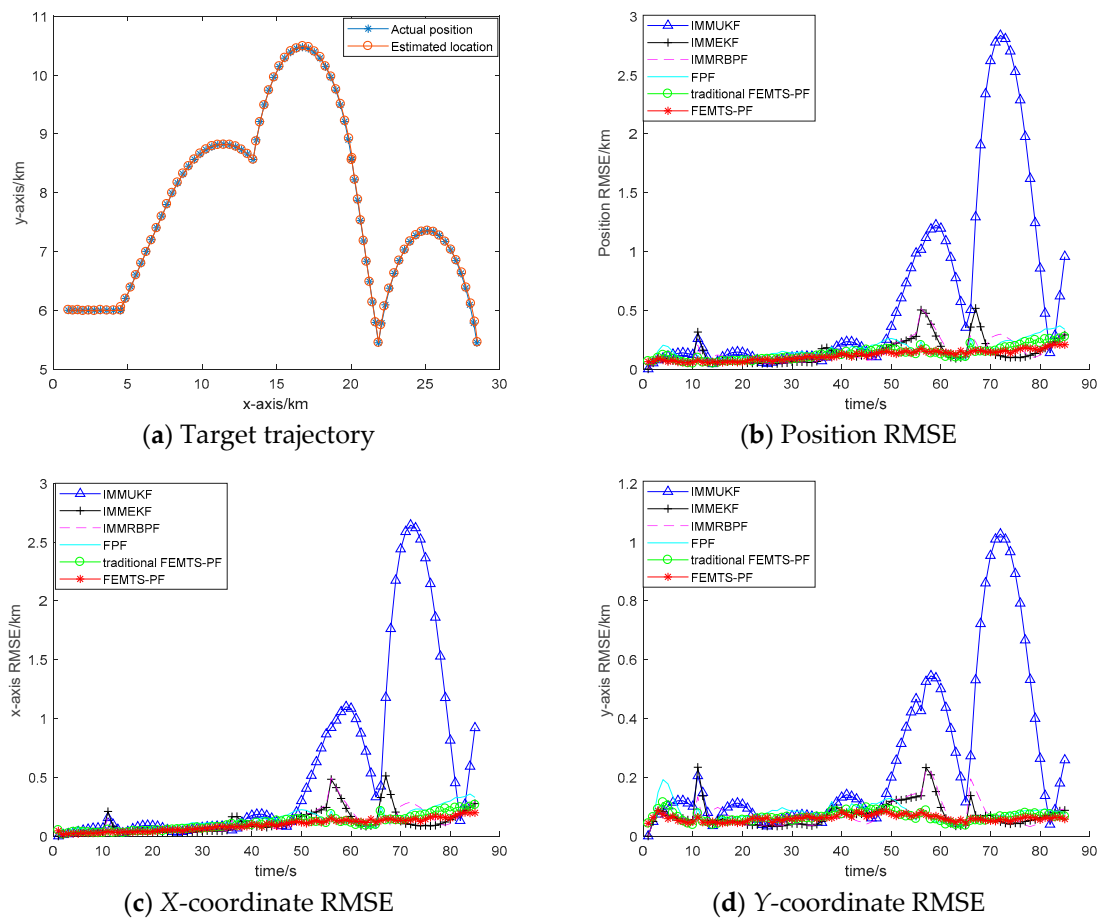
$$h(x_k) = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{y-s_{1,y}}{x-s_{1,x}}\right) \\ \arctan\left(\frac{y-s_{2,y}}{x-s_{2,x}}\right) \end{bmatrix}$$

where  $s_{i,x}, s_{i,y}$  denote the coordinate value of the sensor  $i$ .  $\beta_i$ ,  $i = 1, 2$  denotes the azimuth. Then, the Jacobi matrix for each model is:

$$H_k^i = \begin{bmatrix} -\frac{y-s_{1,y}}{(x-s_{1,x})^2 + (y-s_{1,y})^2} & 0 & \frac{x-s_{1,x}}{(x-s_{1,x})^2 + (y-s_{1,y})^2} & 0 \\ -\frac{y-s_{2,y}}{(x-s_{2,x})^2 + (y-s_{2,y})^2} & 0 & \frac{x-s_{2,x}}{(x-s_{2,x})^2 + (y-s_{2,y})^2} & 0 \end{bmatrix}$$

In order to verify the effectiveness of the proposed algorithm, the FEMTS-PF is implemented and compared with the IMMUKF, IMMEKF, IMMRBPF, fuzzy particle filtering (FPF) and traditional FEMTS-PF algorithms. Figure 2a denotes the estimated trajectory of the target motion. It can be seen from the diagram that the tracking effect of the FEMTS-PF algorithm is approximately the same as the simulation trajectory; there is no obvious loss of phenomenon. It is shown that the algorithm can deal with uncertain information efficiently in nonlinear systems, while the IMMUKF, IMMEKF and IMMRBPF algorithms have a large error when the target maneuvers; the loss phenomenon of the IMMUKF algorithm is more obvious. The main reason for this is that when the target motion model is uncertain, algorithms such as IMM cannot judge the appropriate model, but the FEMTS-PF algorithm can adaptively adjust the model probability according to the premise membership functions, which results in the posterior probability density constructed by the proposed T-S model being closer to the real posterior probability density function and in the tracking accuracy of the algorithm being improved.

Figure 2b–d describe the RMSE of the position, the  $x$ -axis and the  $y$ -axis. The initial mean of innovation (S, L) and heading angle difference (NL, S, PL) in the traditional FEMTS-PF are  $m_{k,v}^i = 0, 2*\sigma$  km and  $m_{k,\phi}^i = -2, 0, 2$  rad, respectively. In the FPF, there is no precise adjustment to the premise parameter, which is insufficient to meet the moving state of the target. Furthermore, for the traditional FEMTS-PF, the fuzzy parameter of the traditional FEM is set by the empirical value and has a certain subjectivity; thus, it results in a lower precision. It can be seen that the tracking effect of the FEMTS-PF algorithm is better than that of the other algorithms, showing a relatively stable tracking performance. The main reason for this is that the premise parameter of the T-S model is identified by the improved FEM, in which the fuzzy parameter is adaptively updated by the FCRM based entropy and spatial-temporal characteristic information, and it can help to obtain a more accurate T-S model structure. When the motion model is uncertain, by partitioning the state space into several subspaces, and because the weight of the subspaces is updated by the premise parameter membership function, which is described by the characteristic of the target, the global estimation model will be closer to the true motion model. Moreover, since the proposal distribution is chosen using the information from the previous stages, sampling is more efficient; thereby, the particle weights will have a lower variance.



**Figure 2.** Performance comparison of the IMMUKF (triangle sign ( $\Delta$ )), IMMEKF (plus sign (+)), IMMRBPF (dotted line (---)), FPF (solid line (—)), traditional FEMTS-PF (circle (o)) and proposed FEMTS-PF (star sign (\*)). (a) The target trajectory of the proposed algorithm and the actual position; (b) position root-mean-square error (RMSE); (c) X-axis RMSE; and (d) Y-axis RMSE.

Table 2 shows the statistical results of the RMSE of the proposed FEMTS-PF algorithm for different particle numbers. As can be seen from the table, the position RMSE is decreased with the increase of the number of particles before 200 particles, but the error increases as the number of particles increases after that. The main reason for this is that the sampling variance of the particles is fixed, that is, the distribution range of the particle sets is the same. When the number of particles is too large, the particles are too dense and the phenomenon of particle overlap will appear, which makes its effectiveness decline. If we increase the number of particles as the sampling variance, so that the particle set distribution is loose, the filtering result will be further improved, but this is based on sacrificial computation. Weighing the pros and cons, the number of particles is set to 200.

**Table 2.** Average RMSE for different samples of FEMTS-PF.

Particles	Position		$x$ -Coordinate		$y$ -Coordinate	
	Mean (km)	Var (km <sup>2</sup> )	Mean (km)	Var (km <sup>2</sup> )	Mean (km)	Var (km <sup>2</sup> )
25	0.1166	0.0019	0.0949	0.0026	0.0616	0.0001
50	0.1158	0.0021	0.0947	0.0027	0.0607	0.0001
100	0.1187	0.0021	0.0974	0.0028	0.0617	0.0001
200	0.1153	0.0018	0.0943	0.0025	0.0605	0.0001
500	0.1159	0.0019	0.0943	0.0025	0.0611	0.0001
1000	0.1158	0.0019	0.0946	0.0026	0.0606	0.0001

Table 3 shows the RMSE of the four algorithms with different numbers of particles. It can be seen that when the particle size is 200, the RMSE of the FEMTS-PF algorithm is about 0.1153 km, and the tracking effects are 81.17%, 18.92%, 20.15%, 29.57% and 11.24% higher than the other algorithms, respectively. The results show that the filtering effect of the FEMTS-PF algorithm is more accurate under the same conditions for the reason that the importance density function of the FEMTS-PF is constructed by the proposed T-S fuzzy model, which improves the estimation accuracy.

**Table 3.** Statistics of the position RMSE under different particle numbers.

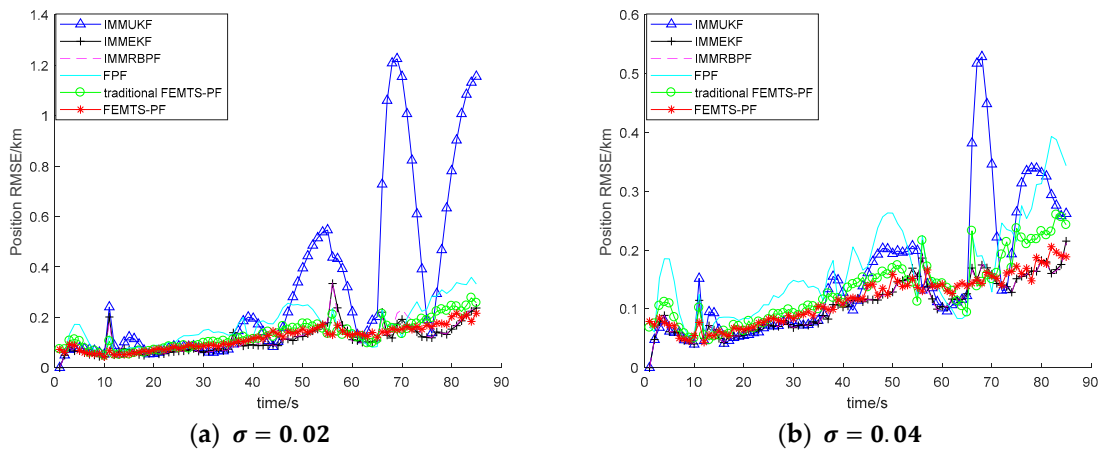
Particles	IMMUKF		IMMEKF		IMMRBPF		FPF		Traditional FEMTS-PF		FEMTS-PF	
	Mean (km)	Var (km <sup>2</sup> )	Mean (km)	Var (km <sup>2</sup> )	Mean (km)	Var (km <sup>2</sup> )	Mean (km)	Var (km <sup>2</sup> )	Mean (km)	Var (km <sup>2</sup> )	Mean (km)	Var (km <sup>2</sup> )
25					0.1405	0.0022	0.1658	0.0078	0.1296	0.0033	0.1132	0.0019
50					0.1401	0.0238	0.1663	0.0065	0.1288	0.0033	0.1158	0.0021
100	0.6122	0.6173	0.1422	0.0109	0.1411	0.0024	0.1644	0.0059	0.13	0.0032	0.1187	0.0021
200					0.1444	0.0956	0.1637	0.0066	0.1299	0.0033	0.1153	0.0018
500					0.1495	0.002	0.1638	0.0061	0.1297	0.0031	0.1159	0.0019
1000					0.1497	0.0023	0.1645	0.0063	0.1294	0.0031	0.1158	0.0019

The running time of a Monte Carlo is shown in Table 4. It can be seen that IMMEKF runs for the least time. As we know, the running time of the particle filtering algorithm is proportional to the number of particles, and the particle resampling step exists in the IMMRBPF algorithm, resulting in a large running time, which improves the estimation efficiency at the expense of the computation. However, in the FEMTS-PF algorithm, the importance density function is constructed by the T-S fuzzy model, which combines the measurement and spatial-temporal information, to effectively improve the diversity of the particles and reduce the degradation of the particles; thus, the resampling step can be omitted. Moreover, the parallel computation is used in the sampling, so the real-time of the FEMTS-PF is largely realized.

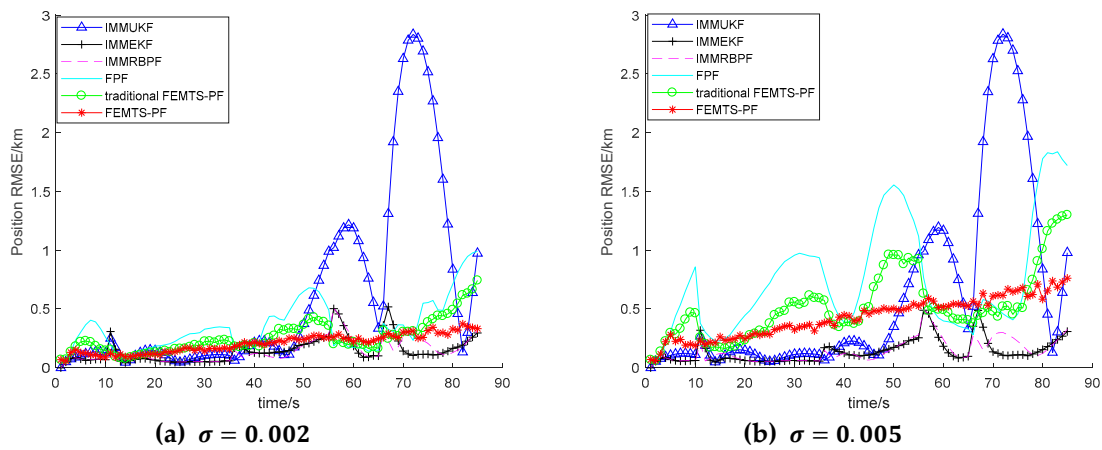
**Table 4.** Comparison of the computation time for all of the algorithms (s).

Case	IMMUKF	IMMEKF	IMMRBPF	FPF	Traditional FEMTS-PF	FEMTS-PF
<b>BOT</b>	0.0573	0.0353	2.8809	0.9275	1.0316	1.0411

Figures 3 and 4 are the RMSE of IMMUKF, IMMEKF, IMMRBPF, FPF, traditional FEMTS-PF and FEMTS-PF under different process and observation noises, respectively. The variance of the process noises and observation noises has a great influence on the filtering results. From the view of the signal-to-noise ratio (SNR), the smaller the SNR, the more likely it is that the noise will submerge the real signal, following which the filtering will fail. The larger the SNR, that is, the less noise there is, the easier it is to filter. As can be seen from Figure 3, the greater the noise, the greater the error, which is difficult to avoid. The process noise is reflected in the trust value of the model; the larger the value is, the closer the filtered value is to the measurement value, and a large amount of noise is introduced; the measurement value is a carried error compared with the real value. However, the purpose of filtering is to reduce the noise interference and make the filtering result close to the real value, and if the value of the filtered value and the real value errors are larger, the filtering accuracy is reduced.



**Figure 3.** The position RMSE for different process noises. (a) The standard deviation of the process noise is 0.02; and (b) the standard deviation of the process noise is 0.04.



**Figure 4.** The position RMSE for different measurement noises. (a) The standard deviation of the measurement noise is 0.002; and (b) the standard deviation of the measurement noise is 0.005.

As can be seen from Figure 4, the tracking effect of the algorithm decreases with the increase of the noise standard deviation. The larger the measurement noise, the slower the convergence rate of the filter output. On the whole, the FEMTS-PF algorithm is relatively good in dealing with non-Gaussian noise, which proves that the proposed algorithm can effectively solve the nonlinear non-Gaussian problem in the sophisticated system.

### 3.2. Maneuvering Target Tracking in Sparse Environment (SMTT)

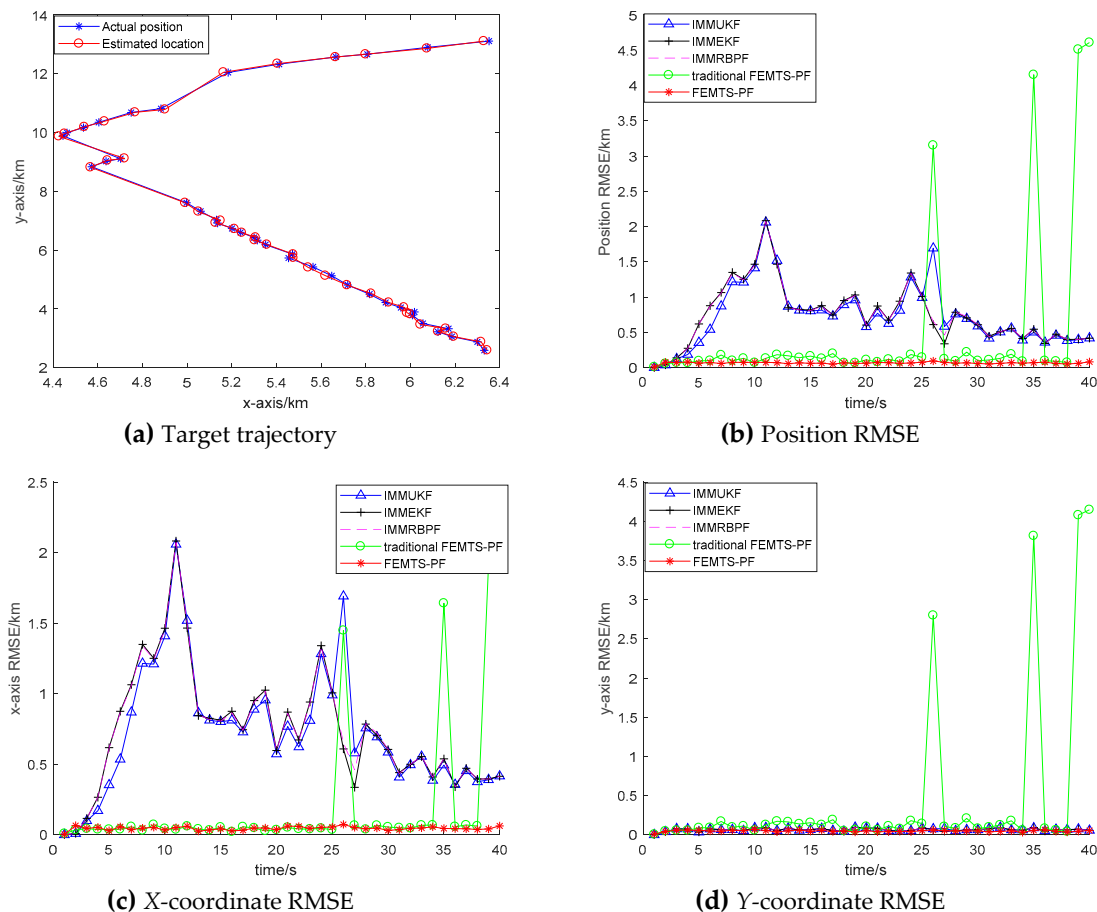
In order to further verify the effectiveness of the proposed algorithm, we implement the proposed FEMTS-PF in a sparse environment. The simulation results are compared with those of the IMMUKF, IMMEKF, IMMRBPF and traditional FEMTS-PF algorithms. For the SMTT, the data includes 40 non-periodic sampling points with a target flight time of 107 s, which includes the time interval,  $x$ -coordinate,  $y$ -coordinate and batch number of the target. The initial data is  $x_0 = [13 \text{ s}, 6.331 \text{ km}, 2.589 \text{ km}, 80]$ . The sample interval  $T$  is not a constant and is defined as follows:

$$T = t(k+1) - t(k)$$

where  $t(k+1)$  and  $t(k)$  are the time interval of  $k+1$  and  $k$ , respectively.

Figure 5a denotes the estimated trajectory of the target motion. It can be seen from the diagram that the algorithm is robust while tracking the sparse and more maneuvered radar data. Figure 5b–d

describe the RMSE of the position, the  $x$ -axis and the  $y$ -axis, respectively. It is shown that the tracking accuracy of the proposed FEMTS-PF algorithm is greatly improved. The position RMSE of the IMMUKF, IMMEKF, IMMRBPF, traditional FEMTS-PF and proposed FEMTS-PF are 0.1010, 0.0998, 0.0979, 0.0991 and 0.0688 km, respectively. The tracking effect of the proposed FEMTS-PF is 31.88%, 31.06%, 29.72% and 30.58% higher than the other algorithms.



**Figure 5.** Performance comparison of the IMMUKF (triangle sign ( $\Delta$ )), IMMEKF (plus sign (+)), IMMRBPF (dotted line (—)), traditional FEMTS-PF (circle ( $o$ )) and FEMTS-PF (star sign (\*)). (a) The target trajectory of the proposed algorithm and the actual position; (b) position root-mean-square error (RMSE); (c) X-axis RMSE; and (d) Y-axis RMSE.

The run-time of a Monte Carlo is shown in Table 5. It can be seen that IMMEKF and IMMUKF have a great advantage in real-time performance, but compared with IMMRBPF, the operation time of the FEMTS-PF algorithm is also decreased by 70.80%. Because of the addition of the FCRM algorithm, the proposed FEMTS-PF has a longer operation time than the traditional FEMTS-PF algorithm. However, comparing the performance and computation time of all of the algorithms, the proposed FEMTS-PF algorithm has achieved a good trade-off in the sparse nonlinear non-Gaussian environment.

**Table 5.** Comparison of the computation time for all of the algorithms (s).

Case	IMMUKF	IMMEKF	IMMRBPF	Traditional FEMTS-PF	FEMTS-PF
SMTT	0.0342	0.0245	1.1600	0.2542	0.3387

#### 4. Conclusions

In this paper, a novel T-S fuzzy modeling particle filtering algorithm is proposed, in which multiple semantic fuzzy sets are used to represent the spatial-temporal characteristic information of the target, and a general framework of the T-S fuzzy model is constructed, in which the model probability is updated by the premise membership functions. The premise parameters of the T-S fuzzy model are identified by an improved fuzzy expectation maximization algorithm, in which the fuzzy parameter is obtained by a FCRM, based on entropy in order to accelerate the convergence speed and avoid the subjective influence caused by the artificial setting of the fixed value. Meanwhile, a strong tracking method is used to identify the consequence parameters. In the particle filter algorithm, the importance density function is constructed by using the proposed fuzzy model, which improves the diversity of the particles, effectively reduces the degradation of the particles, and omits the particle resampling steps. Furthermore, it improves the real-time performance of the algorithm due to the parallel computation. When the target maneuvering or the moving state of the target is uncertain, the T-S fuzzy model adaptively adjusts the optimal moving state by using the fuzzy semantic information of the innovation and heading angle difference. The results of the examples show that the proposed algorithm is more accurate and stable than the IMMUKF, IMMEKF and IMMRBPF algorithms in the passive sensor system with regards to nonlinear and non-Gaussian problems.

**Author Contributions:** Formal analysis, X.W. and L.L.; Investigation, X.W.; Methodology, X.W.; Project administration, L.L.; Resources, L.L. and W.X.; Software, W.X.; Supervision, L.L.; Writing-original draft, X.W.; Writing-review & editing, X.W. and L.L.

**Funding:** This work was supported by the National Natural Science Foundation of China (grant no. 61773267); Science & Technology Program of Shenzhen (grant no. JCYJ20170302145519524); The Postgraduate Innovation Development Fund Project of Shenzhen University (grant no. PIDFP-ZR2019010).

**Conflicts of Interest:** The authors declare no conflict of interest.

#### Appendix A

The lower bound that needs to be optimized is as follows:

$$B(\tau_k^{i,m}, \sigma_k^{i,m}) = -\frac{1}{2} \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log((2\pi)^d |\sigma_k^{i,m}|) - \frac{1}{2} \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i (\theta_k^m - \tau_k^{i,m})^T (\sigma_k^{i,m})^{-1} (\theta_k^m - \tau_k^{i,m}) \\ + \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log \pi_k^i - \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log u_{k,l}^i - \lambda_1 \left( \sum_{i=1}^{N_f} \pi_k^i - 1 \right) - \lambda_2 \left( \sum_{i=1}^{N_f} u_{k,l}^i - 1 \right)$$

Derive the  $\tau_k^{i,m}$  and make it 0:

$$\frac{\partial B}{\partial \tau_k^{i,m}} = \frac{\partial}{\partial \tau_k^{i,m}} \left( -\frac{1}{2} \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i (\theta_k^m - \tau_k^{i,m})^T (\sigma_k^{i,m})^{-1} (\theta_k^m - \tau_k^{i,m}) \right) \\ = \sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} (\theta_k^m - \tau_k^{i,m}) = 0$$

and

$$\sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} \theta_k^m = \sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} \tau_k^{i,m}$$

The  $\tau_k^{i,m}$  is obtained as:

$$\tau_k^{i,m} = \frac{\sum_{l=1}^C u_{k,l}^i \theta_k^m}{\sum_{l=1}^C u_{k,l}^i}$$



Since  $\sigma_k^{i,m}$  is a matrix variable, the partial property of the matrix variable requires the following properties:  $\frac{\partial \log|C|}{\partial C} = (C^T)^{-1}$ ,  $\frac{\partial}{\partial C}[a^T C^{-1} b] = -(C^T)^{-1} a b^T (C^T)^{-1}$ ,  $(\sigma_k^{i,m})^T = \sigma_k^{i,m}$ .

Therefore, derive the  $\sigma_k^{i,m}$  and make it 0:

$$\begin{aligned} \frac{\partial B}{\partial \sigma_k^{i,m}} &= \frac{\partial}{\partial \sigma_k^{i,m}} \left( -\frac{1}{2} \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \log(|\sigma_k^{i,m}|) - \frac{1}{2} \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i (\theta_k^m - \tau_k^{i,m})^T (\sigma_k^{i,m})^{-1} (\theta_k^m - \tau_k^{i,m}) \right) \\ &= -\frac{1}{2} \sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} + \frac{1}{2} \sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} (\theta_k^m - \tau_k^{i,m}) (\theta_k^m - \tau_k^{i,m})^T (\sigma_k^{i,m})^{-1} = 0 \end{aligned}$$

and

$$\begin{aligned} \sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} &= \sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} (\theta_k^m - \tau_k^{i,m}) (\theta_k^m - \tau_k^{i,m})^T (\sigma_k^{i,m})^{-1} \\ \sigma_k^{i,m} \sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} \sigma_k^{i,m} &= \sigma_k^{i,m} \sum_{l=1}^C u_{k,l}^i (\sigma_k^{i,m})^{-1} (\theta_k^m - \tau_k^{i,m}) (\theta_k^m - \tau_k^{i,m})^T (\sigma_k^{i,m})^{-1} \sigma_k^{i,m} \\ \sigma_k^{i,m} \sum_{l=1}^C u_{k,l}^i &= \sum_{l=1}^C u_{k,l}^i (\theta_k^m - \tau_k^{i,m}) (\theta_k^m - \tau_k^{i,m})^T \end{aligned}$$

The  $\sigma_k^{i,m}$  is obtained as:

$$\sigma_k^{i,m} = \frac{\sum_{l=1}^C u_{k,l}^i (\theta_k^m - \tau_k^{i,m}) (\theta_k^m - \tau_k^{i,m})^T}{\sum_{l=1}^C u_{k,l}^i}$$

## Appendix B

The objective function of the entropy adjustment-based FCRM method is defined as follows:

$$Y(u_{k,l}^i) = -\sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \ln(u_{k,l}^i) - \beta \cdot \sum_{l=1}^C \sum_{i=1}^{N_f} u_{k,l}^i \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right] + \sum_{l=1}^C \lambda_k \left( \sum_{i=1}^{N_f} u_{k,l}^i - 1 \right)$$

Derive the  $u_{k,l}^i$  and make it 0:

$$\frac{\partial Y}{\partial u_{k,l}^i} = -(1 + \ln u_{k,l}^i) - \beta \cdot \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right] + \lambda_k = 0$$

and

$$\begin{aligned} \ln u_{k,l}^i &= -1 - \beta \cdot \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right] + \lambda_k \\ u_{k,l}^i &= \exp\left(-1 - \beta \cdot \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right] + \lambda_k\right) \\ &= \exp(-1 + \lambda_k) \times \exp\left(-\beta \cdot \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right]\right) \end{aligned}$$

On the basis of  $\sum_{i=1}^{N_f} u_{k,l}^i = 1$ :

$$\sum_{i=1}^{N_f} u_{k,l}^i = \sum_{i=1}^c \left( \exp(-1 + \lambda_k) \times \exp\left(-\beta \cdot \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right]\right) \right) = 1$$

$$\exp(-1 + \lambda_k) = \frac{1}{\sum_{j=1}^c \exp\left(-\beta \cdot \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right]\right)}$$

The  $u_{k,l}^i$  is obtained as:

$$u_{k,l}^i = \frac{\exp\left(-\beta \left[ (D_{k,l}^i)^2 + \sum_{m=1}^G \omega_k^{i,m} \theta_k^m \right]\right)}{\sum_{q=1}^{N_f} \left( \exp\left(-\beta \left[ (D_{k,l}^q)^2 + \sum_{m=1}^G \omega_k^{q,m} \theta_k^m \right]\right) \right)}$$

## References

1. Sinopoli, B.; Schenato, L.; Franceschetti, M.; Poolla, K.; Jordan, M.I.; Sastry, S.S. Kalman filtering with intermittent observations. *IEEE Trans. Autom. Control* **2004**, *1*, 1453–1464. [[CrossRef](#)]
2. Yu, J.; Chen, L. From Static to Dynamic Tag Population Estimation: An Extended Kalman Filter Perspective. *IEEE Trans. Commun.* **2016**, *64*, 4706–4719. [[CrossRef](#)]
3. Xiao, M.; Zhang, Y.; Wang, Z.; Fu, H. Augmented robust three-stage extended Kalman filter for Mars entry-phase autonomous navigation. *Int. J. Syst. Sci.* **2018**, *49*, 27–42. [[CrossRef](#)]
4. Locubiche-Serra, S.; Seco-Granados, G.; López-Salcedo, J.A. Closed-Form Approximation for the Steady-State Performance of Second-Order Kalman Filters. *IEEE Signal Process. Lett.* **2018**, *25*, 268–272. [[CrossRef](#)]
5. Julier, S.; Uhlmann, J.K. *A General Method for Approximating Nonlinear Transformations of Probability Distributions*; Technical report; Robotics Research Group, Department of Engineering Science, University of Oxford: Oxford, UK, 1996.
6. Ishihara, S.; Yamakita, M. Gain constrained robust UKF for nonlinear systems with parameter uncertainties. In Proceedings of the 2016 European Control Conference (ECC), Aalborg, Denmark, 29 June–1 July 2016; pp. 1709–1714.
7. Gao, B.; Hu, G.; Gao, S.; Zhong, Y.; Gu, C. Multi-sensor Optimal Data Fusion for INS/GNSS/CNS Integration Based on Unscented Kalman Filter. *Int. J. Control Autom. Syst.* **2018**, *16*, 129–140. [[CrossRef](#)]
8. Qi, J.; Sun, K.; Wang, J.; Liu, H. Dynamic State Estimation for Multi-Machine Power System by Unscented Kalman Filter With Enhanced Numerical Stability. *IEEE Trans. Smart Grid* **2018**, *9*, 1184–1196. [[CrossRef](#)]
9. Gordon, N.J.; Salmond, D.J.; Smith, A.F.M. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proc. F* **1993**, *140*, 107–113. [[CrossRef](#)]
10. Isard, M.; Blake, A. Condensation—Conditional Density Propagation for Visual Tracking. *Int. J. Comput. Vis.* **1998**, *29*, 5–28. [[CrossRef](#)]
11. Li, Y.; Zhuang, X.; Liu, Y. UPF Tracking Method Based on Color and SIFT Features Adaptive Fusion. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2014**, *7*, 379–390. [[CrossRef](#)]
12. Abouzahir, M.; Elouardi, A.; Bouaziz, S.; Latif, R.; Abdelouahed, T. An improved Rao-Blackwellized particle filter based-SLAM running on an OMAP embedded architecture. In Proceedings of the 2014 Second World Conference on Complex Systems (WCCS), Agadir, Morocco, 10–12 November 2014; pp. 716–721.
13. Ubeda-Medina, L.; Garcia-Fernandez, A.; Grajal, J. Adaptive auxiliary particle filter for track-before-detect with multiple targets. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 2317–2330. [[CrossRef](#)]
14. Garcia, R.V.; Silva, W.R.; Pardo, P.C.; Kuga, H.K.; Zanardi, M.C. Sequential nonlinear estimation: Regularized particle filter applied to the attitude estimation problem with real data. *Comput. Appl. Math.* **2017**, *37*, 110–121. [[CrossRef](#)]
15. Jing, L.; Vadakkepat, P. Interacting MCMC particle filter for tracking maneuvering target. *Digit. Signal Process.* **2010**, *20*, 561–574. [[CrossRef](#)]
16. Kotecha, J.H.; Djuric, P.M. Gaussian particle filtering. *IEEE Trans. Signal Process.* **2003**, *51*, 2592–2601. [[CrossRef](#)]
17. Kotecha, J.H.; Djuric, P.M. Gaussian sum particle filtering. *IEEE Trans. Signal Process.* **2003**, *51*, 2602–2612. [[CrossRef](#)]

18. Yu, M.; Gong, L.; Oh, H.; Chen, W.H.; Chambers, J. Multiple Model Ballistic Missile Tracking with State-Dependent Transitions and Gaussian Particle Filtering. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 1066–1081. [[CrossRef](#)]
19. Dhassi, Y.; Aarab, A. Visual tracking based on adaptive interacting multiple model particle filter by fusing multiples cues. *Multimed. Tools Appl.* **2018**, *77*, 26259–26292. [[CrossRef](#)]
20. Bando, T.; Shibata, T.; Doya, K.; Ishii, S. Switching particle filters for efficient visual tracking. *Robot. Auton. Syst.* **2006**, *54*, 873–884. [[CrossRef](#)]
21. Meshgi, K.; Maeda, S.I.; Oba, S.; Skibbe, H.; Li, Y.Z.; Ishii, S. An occlusion-aware particle filter tracker to handle complex and persistent occlusions. *Comput. Vis. Image Underst.* **2016**, *150*, 81–94. [[CrossRef](#)]
22. Martino, L.; Elvira, V.; Camps-Valls, G. Group Importance Sampling for Particle Filtering and MCMC. *Digit. Signal Process.* **2017**, *82*, 133–151. [[CrossRef](#)]
23. Dunne, D.; Kirubarajan, T. Multiple Model Multi-Bernoulli Filters for Manoeuvring Targets. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 2679–2692. [[CrossRef](#)]
24. Yang, F.; Zhang, W. Multiple Model Bernoulli Particle Filter for Maneuvering Target Tracking. *J. Electron. Inf. Technol.* **2017**, *39*, 634–639. (in Chinese).
25. Chen, Z.; Qu, Y.; Xi, Z.; Bo, Y.; Liu, B. Efficient Particle Swarm Optimized Particle Filter Based Improved Multiple Model Tracking Algorithm. *Comput. Intell.* **2017**, *33*, 262–279. [[CrossRef](#)]
26. Urteaga, M.; Bugallo, F.; Djuric, P.M. Sequential Monte Carlo methods under model uncertainty. In Proceedings of the 2016 IEEE Statistical Signal Processing Workshop (SSP), Palma de Mallorca, Spain, 26–29 June 2016; pp. 1–5.
27. Martino, L.; Read, J.; Elvira, V.; Louzada, F. Cooperative parallel particle filters for online model selection and applications to urban mobility. *Digit. Signal Process.* **2017**, *60*, 172–185. [[CrossRef](#)]
28. Carvalho, C.M.; Johannes, M.S.; Lopes, H.F.; Polson, N.G. Particle Learning and Smoothing. *Stat. Sci.* **2010**, *25*, 88–106. [[CrossRef](#)]
29. Hounek, L.; Cintula, P. From fuzzy logic to fuzzy mathematics: A methodological manifesto. *Fuzzy Sets Syst.* **2006**, *157*, 642–646.
30. Widynski, N.; Dubuisson, S.; Bloch, I. Integration of Fuzzy Spatial Information in Tracking Based on Particle Filtering. *IEEE Trans. Syst. Man Cybern. Part B* **2011**, *41*, 635–649. [[CrossRef](#)] [[PubMed](#)]
31. Li, L.; Li, C.; Cao, W.; Liu, Z.X. Fuzzy Quadrature Particle Filter for Maneuvering Target Tracking. *Int. J. Fuzzy Syst.* **2016**, *18*, 647–658. [[CrossRef](#)]
32. Li, L.; Xie, W.; Liu, Z. A novel quadrature particle filtering based on fuzzy c-means clustering. *Knowl. Based Syst.* **2016**, *106*, 105–115.
33. Chang, C.W.; Tao, C.W. A Novel Approach to Implement Takagi-Sugeno Fuzzy Models. *IEEE Trans. Cybern.* **2017**, *47*, 2353–2361. [[CrossRef](#)] [[PubMed](#)]
34. Xie, X.; Lin, L.; Zhong, S. Process Takagi-Sugeno model: A novel approach for handling continuous input and output functions and its application to time series prediction. *Knowl. Based Syst.* **2014**, *63*, 46–58. [[CrossRef](#)]
35. Papi, F.; Podt, M.; Boers, Y.; Battistello, G.; Ulmke, M. On constraints exploitation for particle filtering based target tracking. In Proceedings of the 2012 15th International Conference on Information Fusion, Singapore, 9–12 July 2012; pp. 455–462.
36. Li, L.-Q.; Wang, X.-L.; Xie, W.-X.; Liu, Z.-X. A novel recursive T-S fuzzy semantic modeling approach for discrete state-space systems. *Neurocomputing* **2019**, *340*, 222–232. [[CrossRef](#)]
37. Wang, X.; Li, L.; Xie, W. T-S Fuzzy Multiple Model Target Tracking Algorithm with UKF Parameter Identification. *J. Signal Process.* **2019**, *35*, 361–368.
38. Li, J.; Zhao, R.; Chen, J.; Zhao, C.; Zhu, Y. Target tracking algorithm based on adaptive strong tracking particle filter. *IET Sci. Meas. Technol.* **2016**, *10*, 704–710. [[CrossRef](#)]
39. Vila, J.P.; Schniter, P. Expectation-Maximization Gaussian-Mixture Approximate Message Passing. *IEEE Trans. Signal Process.* **2013**, *61*, 4658–4672. [[CrossRef](#)]
40. Prakash, R.M.; Kumari, R.S.S. Spatial Fuzzy C Means and Expectation Maximization Algorithms with Bias Correction for Segmentation of MR Brain Images. *J. Med Syst.* **2017**, *41*, 15–23. [[CrossRef](#)] [[PubMed](#)]

41. Buckley, S.M. Estimates for Operator Norms on Weighted Spaces and Reverse Jensen Inequalities. *Trans. Am. Math. Soc.* **1993**, *340*, 253–272. [[CrossRef](#)]
42. Li, L.; Xie, W.; Huang, J.; Huang, J. Multiple Model Rao–Blackwellized Particle Filter for Maneuvering Target Tracking. *Int. J. Def. Sci.* **2009**, *59*, 197–204.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).