

Article

## A Particle Filter Track-Before-Detect Algorithm Based on Hybrid Differential Evolution

Chaozhu Zhang, Lin Li \* and Yu Wang

Department of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China; E-Mails: zhangchaozhu@hrbeu.edu.cn (C.Z.); wangyu1256@hrbeu.edu.cn (Y.W.)

\* Author to whom correspondence should be addressed; E-Mail: lilin\_2005@163.com; Tel.: +86-150-4562-2396; Fax: +86-451-8258-9811.

Academic Editor: Stefano Mariani

*Received: 19 August 2015 / Accepted: 27 October 2015 / Published: 3 November 2015*

---

**Abstract:** In this paper, we address the problem of detecting and tracking targets with a low signal-to-noise ratio (SNR) by exploiting hybrid differential evolution (HDE) in the particle filter track-before-detect (PF-TBD) context. Firstly, we introduce the Bayesian PF-TBD method and its weaknesses. Secondly, the HDE algorithm is regarded as a novel particle updating strategy, which is proposed to optimize the performance of the PF-TBD algorithm. Thirdly, we combine the systematic resampling approach to enhance the performance of the proposed algorithm. Then, an improved PF-TBD algorithm based on the HDE method is proposed. Experiment results indicate that the proposed method has better performance in detecting and tracking than previous algorithms when the targets have a low SNR.

**Keywords:** track-before-detect; particle filter; hybrid differential evolution

---

### 1. Introduction

The traditional target detection approach is called track-after-detect (TAD); it involves the application of a detection threshold at every scan, which leads to a loss of information, as some sensor responses may fail to pass the detection threshold. Additionally, TAD methods cannot detect the targets with a low signal-to-noise (SNR) under 10 dB [1]. An alternative approach is called track-before-detect (TBD) [2],

which jointly processes the raw sensor data (infrared sensor, radar, passive sonar, *etc.*) without a threshold and jointly declares the presence of a target and its track. This is acceptable if the SNR is low.

A large number of TBD algorithms have been developed in the past several decades including direct maximum likelihood [3], the Hough transform [4], dynamic programming [5,6], and so on. These methods generally require discretization of the state space and are very computationally intensive. Recently, the popularity of the particle filter (PF) method has increased due to its flexibility to handle cases where the dynamic and observation models are non-linear and/or non-Gaussian [7]. Particle filter track-before-detect (PF-TBD) was first proposed by Salmond [8], in parallel with the work by Boers [9]. These algorithms have since been extended in [10–12], which form a single target filter for two targets. The literature [10] presents a performance comparison of three particle filters using several different particle proposal densities designed for a single target. In [11], a new model for solving the problem of detecting a single target arrival and tracking its state in a TBD context is presented. The work in [12] shows a modeling setup and algorithm for a multiple target recursive Bayesian TBD application in a radar context. Phase information is used to improve the PF-TBD algorithm in the literature [13]. In [14], the PF is replaced by a cost-reference particle filter (CRPF). However, these proposals have some disadvantages in terms of computational complexity.

In this paper, an improved PF-TBD algorithm is proposed, which exploits a novel particle updating strategy that regards the particle updating process as an optimization problem using the hybrid differential evolution (HDE) algorithm [15,16] before the resampling process. It can be drawn from this strategy that we can reduce effectively the particle impoverishment and optimize the performance of weak target detection. Simulation results are presented to demonstrate the validity of the algorithm.

The remainder of this paper is organized as follows. The TBD system model is introduced in Section 2. In Section 3, first of all, we introduce the Bayesian PF-TBD approach and its weaknesses. Then, the HDE algorithm is given, and an improved PF-TBD algorithm is proposed. We evaluate the performance of the proposed algorithm in Section 4. At last, conclusions and pointers for future research are presented in Section 5.

## 2. TBD System Setup

In this section, we will describe the models that will be used in the TBD application, the target dynamics models and the measurement models.

### 2.1. Target Dynamic Model

The target state is represented by a five-dimensional state vector  $\mathbf{x}_k = [x_k \dot{x}_k y_k \dot{y}_k I_k]^T$ , where  $(x_k, y_k)$ ,  $(\dot{x}_k, \dot{y}_k)$  and  $I_k$  represent the position and velocity in the  $x$  and  $y$  dimensions and the intensity of the target at time  $k$ , respectively.  $k$  is the time index. A constant-velocity (CV) model is widely used as an example.

The discrete-time system dynamics of this model is of the form:

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

where  $\mathbf{w}_k$  is the zero mean Gaussian noise process with covariance  $\mathbf{Q}_{CV}$  and  $\mathbf{F}_{CV}$  is the process transition matrix, which are defined by:

$$\mathbf{F}_{CV} = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\mathbf{Q}_{CV} = \begin{bmatrix} q_1 T_{CV}^3 / 3 & q_1 T_{CV}^2 / 2 & 0 & 0 & 0 \\ q_1 T_{CV}^2 / 2 & q_1 T_{CV} & 0 & 0 & 0 \\ 0 & 0 & q_1 T_{CV}^3 / 3 & q_1 T_{CV}^2 / 2 & 0 \\ 0 & 0 & q_1 T_{CV}^2 / 2 & q_1 T_{CV} & 0 \\ 0 & 0 & 0 & 0 & q_2 T_{CV} \end{bmatrix} \tag{3}$$

where  $T_{CV}$  is the sampling period,  $q_1$  and  $q_2$  are the power spectral density of the acceleration noise and the noise in the rate of change of target return intensity, respectively.

The presence of a target in the data is denoted by a target existence variable  $E_k$ , a target existing in the data,  $E_k = 1$ , otherwise  $E_k = 0$ . The probability of target existence is modeled as a two-state Markov chain, the probability transition matrix of which is defined by:

$$\mathbf{\Pi} = \begin{bmatrix} 1 - P_d & P_b \\ P_d & 1 - P_b \end{bmatrix} \tag{4}$$

where  $P_b$  and  $P_d$  denote the probability of target birth and death, respectively.

### 2.2. Measurement Model

Generally, we use the results of the sensor signal processing, which are a series of images, and the signal is assumed to be two-dimensional images [8–10], each frame consisting of  $n$  cells in the  $x$  dimension and  $m$  cells in the  $y$  dimension. The measurement at time  $k$  is the intensity of each cell in the image, which is given by  $z_k = \{z_k^{(i,j)} : i = 1, \dots, n; j = 1, \dots, m\}$ . As in Chapter 11 of [7], for each cell, the measurement model can be defined as follows:

$$z_k^{(i,j)} = \begin{cases} h_k^{(i,j)}(\mathbf{x}_k) + v_k^{(i,j)} & E_k = 1 \\ v_k^{(i,j)} & E_k = 0 \end{cases} \tag{5}$$

where  $h_k^{(i,j)}(\mathbf{x}_k)$  is a contribution from the intensity of the target in the cell  $(i, j)$  and  $v_k^{(i,j)}$  is the measurement noise in the cell, which is a known Gaussian distribution with zero mean, variance  $\sigma^2$ . In this work, we adopt the sensor point spread function, a point target of intensity  $I_k$  at position  $(x_k, y_k)$ ; the contribution to cell  $(i, j)$  can be written as:

$$h_k^{(i,j)}(\mathbf{x}_k) = \frac{\Delta_x \Delta_y I_k}{2\pi \Sigma^2} \exp\left(-\frac{(x_k - i\Delta_x)^2 + (y_k - i\Delta_y)^2}{2\Sigma^2}\right) \tag{6}$$

Here, the parameter  $\Sigma$  represents the extent of blurring and  $\Delta_x$  and  $\Delta_y$  denote the size of a cell in the  $x$  and  $y$  dimensions, respectively. Then, the set of complete measurements up to time  $k$  is denoted by  $\mathbf{Z} = \{z_1, \dots, z_k\}$ .

Under the assumptions of Gaussian measurement noise, which is independent of cell-to-cell, the likelihood function in the presence and absence of a target can be written as:

$$p(z_k^{(i,j)} | x_k, E_k = 1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{[z_k^{(i,j)} - h^{(i,j)}(x_k)]^2}{2\sigma^2}\right) \tag{7}$$

$$p(z_k^{(i,j)} | E_k = 0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{[z_k^{(i,j)}]^2}{2\sigma^2}\right) \tag{8}$$

Then:

$$p(z_k | x_k, E_k) = \prod_{i=1}^n \prod_{j=1}^m p(z_k^{(i,j)} | x_k, E_k) \tag{9}$$

The likelihood ratio for the cell is:

$$l(z_k | x_k, E_k) = \begin{cases} \frac{p(z_k | x_k, E_k = 1)}{p(z_k | E_k = 0)} & E_k = 1 \\ 1 & E_k = 0 \end{cases} \tag{10}$$

where:

$$l(z_k | x_k, E_k = 1) = \prod_{i=1}^n \prod_{j=1}^m \exp\left(-\frac{h^{(i,j)}(x_k)[h^{(i,j)}(x_k) - 2z_k^{(i,j)}]}{2\sigma^2}\right) \tag{11}$$

### 3. Hybrid Differential Evolution Particle Filter Track-Before-Detect Algorithm

In this section, we give a derivation of the PF-based TBD algorithm and analyze the weaknesses of the PF-TBD, and then, an improved algorithm based on HDE is proposed.

#### 3.1. Bayesian Particle Filter Track-Before-Detect

The main problem of TBD can be formulated in the framework of recursive Bayesian estimation in different expansion forms. The target state is not defined when the target does not exist in the data, where  $E_k=0$ . Then, target detection and tracking can be implemented based on the posterior density  $p(x_k, E_k | Z_k)$  when the target does exist, where  $E_k = 1$ .

$$\begin{aligned} p(x_k, E_k = 1 | Z_k) &= \frac{p(z_k | x_k, E_k = 1)p(x_k, E_k = 1 | Z_{k-1})}{p(z_k | Z_{k-1})} \\ &\propto l(z_k | x_k, E_k = 1)p(x_k, E_k = 1 | Z_{k-1}) \end{aligned} \tag{12}$$

The predicted target state can be written in terms of the target and existence at the previous time, giving:

$$\begin{aligned}
 p(x_k, E_k = 1 / Z_{k-1}) &= \int [p(x_k, E_k = 1 / x_{k-1}, E_{k-1} = 1) p(x_{k-1}, E_{k-1} = 1 / Z_{k-1}) \\
 &\quad + p(x_k, E_k = 1, E_{k-1} = 0 / Z_{k-1})] dx_{k-1} \\
 &= \int p(x_k / x_{k-1}, E_k = 1, E_{k-1} = 1) p(E_k = 1 / E_{k-1} = 1) \\
 &\quad \cdot p(x_{k-1}, E_{k-1} = 1 / Z_{k-1}) dx_{k-1} \\
 &\quad + p(x_k / x_{k-1}, E_k = 1, E_{k-1} = 1, Z_{k-1}) p(E_k = 1 / E_{k-1} = 0) \\
 &= \int p(x_k / x_{k-1}, E_k = 1, E_{k-1} = 1) (1 - P_d) p(x_{k-1}, E_{k-1} = 1 / Z_{k-1}) dx_{k-1} \\
 &\quad + p_b(x_k) P_b
 \end{aligned} \tag{13}$$

where  $p_b(x_k)$  is the prior density of a target when it has appeared in the data between time  $k-1$  and  $k$ . The algorithm proceeds by constructing two sets of particles. The first set, the birth particles, estimate  $p(x_k / E_k = 1, E_{k-1} = 0, Z_k)$ , that is the case where the target did not exist in the data at time  $k-1$ , but does at time  $k$ . The second set, the continuing particles, estimates  $p(x_k / E_k = 1, E_{k-1} = 1, Z_k)$ , which is the case where the target has continued to exist in the data  $k-1$  to  $k$ . Both sets of particles require a proposal distribution. For the continuing particles, the target dynamics are used in Equation (4). For the birth particles, a uniform distribution can be used. Then, the former is approximated using interacting sets of particles, and the calculation of the latter is based on the particle weights in these two sets.

Additionally, the estimate of the probability of existence can be calculated by particle weights in these two sets. State estimation can be obtained after resampling from these combined sets. There are three main resampling methods: multinomial resampling, residual resampling and systematic resampling [17].

To a certain extent, the resampling process is introduced to improve the degradation phenomenon of the particles. However, it also brought some problems. Particles that have higher weights will be reproduced many times. Additionally, those that have lower weights will fade away. Then, all particles will collapse to a small area, which will lead to the sample point sets not being sufficient after several iterations. The sample point sets are described the a posteriori probability density function. In the PF-TBD algorithm, if the particles collapse into a small area at time  $k$ , most of the particles, after the resampling process, are “the continuing particles”. This leads to these particles, which are produced by the state transition equation at time  $k + 1$ , also collapsing into a small area. Moreover, if the area deviates from the true location, the target will be lost.

### 3.2. Improved PF-TBD-Based on the HDE Algorithm

In this paper, to improve the particle impoverishment problem, produced by the resampling process, a novel particle updating strategy is proposed before the resampling process to overcome the problem. We regard the particle updating process as an optimization problem using the HDE algorithm to increase the diversity of particles and to avoid the particles collapsing into a small area.

The HDE algorithm based on the simulated annealing (SA) method is utilized for the search capability of the SA to enhance the convergence capability of differential evolution (DE) in later periods and to improve the robustness of the DE algorithm [15,16]. It uses the Metropolis criterion of the SA algorithm to replace the section step of the DE algorithm. It seems that the computation of the HDE step is more complicated than DE. However, the HDE algorithm can achieve better results than DE with much lower iterations [15,16]. The HDE algorithm saves time to make up for its complexity.

Therefore, we choose the better optimized performance of HDE to obtain sampling particles that conform more to the true state distribution.

### 3.2.1. Hybrid Differential Evolution Algorithm

The HDE algorithm relies on the initial population generation, mutation, recombination and the new selection to probe the search space through iterative progress until the termination criteria are met.

Detailed steps are presented accordingly in the sequel.

Step 1: Creating the initial population.

The first step of HDE is to create the initial population samples (the number of generations is  $g = 0$ ) in  $n$  dimensional space as follows:

$$x_{ij}(0) = x_{ij}^L + rand_{ij}(0,1)(x_{ij}^U - x_{ij}^L) \tag{14}$$

where  $i = 1, 2, \dots, NP$ ;  $j = 1, 2, \dots, n$ , and  $NP$  is the population size.  $x_{ij}^U$  and  $x_{ij}^L$  denote the upper and lower limits of the  $j$ -th variable in the population respectively.  $rand_{ij}(0,1)$  represents a uniformly-distributed random value within  $[0, 1]$ .

Step 2: Mutation operation.

The function of mutation in HDE is to maintain the diversity of a population. A typical HDE mutation sample formulation is:

$$h_{ij}(g) = x_{ij}(g) + F_m \cdot (x_{r1j}(g) - x_{r2j}(g)) \tag{15}$$

where  $g$  represents the  $g$ -th generation, and  $h_{ij}(g)$  are the mutated vector samples.  $r1 \neq r2 \neq i$  and  $r1, r2$  are randomly-selected integers within  $NP$ ,  $r1, r2 \in \{1, 2, \dots, NP\}$ .  $F_m$  is the scaling factor.

Step 3: Crossover operation.

The basic crossover process is a discrete recombination, which employs a crossover constant  $C \in [0, 1]$  to determine if the newly-generated individual samples are to be recombined. The expression of the crossover process is given in Equation (16):

$$v_{ij}(g) = \begin{cases} h_{ij}(g) & rand(0,1) \leq C \\ x_{ij}(g) & rand(0,1) > C \end{cases} \tag{16}$$

where  $v_{ij}(g)$  are the trial vector samples.

Step 4: New selection operation.

The HDE algorithm adopts the Metropolis criterion of the SA algorithm to select whether to accept the trial vector samples. We construct the parameter about fitness function values  $\Delta f = f(v_{ij}(g)) - f(x_{ij}(g))$ , and then, we decide the trial vector samples by the Metropolis criterion.

Metropolis criterion: if  $\Delta f < 0$ , the trial vector sample is accepted. If  $\Delta f > 0$ , we calculate an acceptance probability  $P(\Delta f) = \exp(-\frac{\Delta f}{T_{SA}})$  and generate a pseudo-random number  $r$ . If  $P(\Delta f) \geq r$ , the trial vector sample is accepted; otherwise, the point is discarded.

Step 5: Cool down operation.

In this step, the cool down operation of the SA algorithm is executed. We defined  $T_{SA} = T_{SA} \cdot \rho$ ; here,  $T_{SA}$  is an annealing temperature, and  $\rho \in (0,1)$  is an annealing control parameter.

When the new population is propagated, Step 2 to Step 5 is repeated until the pre-specified temperature  $T_0$  is reached.

### 3.2.2. Proposed Method

Firstly, the particle updating process is optimized by the HDE algorithm. The importance-sampling particles are regarded as the initial population of the HDE algorithm, and the corresponding weights are treated as the fitness functions of the target vectors, respectively. The HDE resampling scheme recombines the particles by using an iterative process of mutation, crossover and the simulated annealing operator. Then, a new set of diverse particles is propagated. Secondly, the systematic resampling is chosen to be executed.

The proposed algorithm exploits a hybrid estimation technique, where the state is expanded to include the existence variable  $\mathbf{s}_k = [\mathbf{x}_k^T E_k]^T$ . Given a set of particles describing the joint posterior density at time  $k-1$ ,  $\{s_{k-1}^n, w_{k-1}^n\}_{n=1}^M$  with uniform weights,  $N$  is the particle number. Then, the proposed algorithm contains the following steps:

Step 1: Calculate the target existence variable:  $[\{E_k^n\}_{n=1}^N] = [\{E_{k-1}^n\}_{n=1}^N, \Pi]$ .

Step 2: Create a set of particles. These are two possible situations.

(i) Set the ‘birth particle’ state as a sample from the proposal density  $\mathbf{x}_k^n \sim q_b(\mathbf{x}_k / E_k^n = 1, E_{k-1}^n = 0, z_k)$ . The target’s location  $(x_k^n, y_k^n)$  is uniform over the sensor field-of-view. Its velocity and intensity are assumed to be uniform as follows  $q_b((\dot{x}_k^n, \dot{y}_k^n) / z_k) = U[v_{\min}, v_{\max}]$ ,  $q_b(I_k / z_k) = U[I_{\min}, I_{\max}]$ , where  $v_{\min}$  and  $v_{\max}$  are the minimum and the maximum of the target velocity, respectively.  $I_{\min}$  and  $I_{\max}$  are the minimum and the maximum of the target intensity, respectively.

(ii) Set the ‘survival particle’ state sampled from the proposal  $\mathbf{x}_k^n \sim q_b(\mathbf{x}_k / E_k^n = 1, E_{k-1}^n = 1, z_k)$ . That is, the ‘survival particle’  $\mathbf{x}_k$  is sampled according to the target state transition Equation (1).

(iii) For each particle, compute the un-normalized weight  $\tilde{w}_k^n$  by using the likelihood ratio  $\tilde{w}_k^n = l(z_k / x_k^n, E_k^n = 1)$ .

Step 3: Optimize the sample particles by HDE. The sampling particles are regarded as the initial population of the HDE algorithm, and the corresponding weights are regarded as the fitness functions. According to the HDE algorithm, the process is iterated until the optimal population is found or a pre-specified condition is reached, then we can get the optimal set of particles  $\{\mathbf{x}'_k, \tilde{w}'_k\}_{n=1}^N$ .

Step 4: Normalize the weights of each particle:

$$w'_k = \tilde{w}'_k / \sum_{n=1}^N \tilde{w}'_k = l(z_k | x'_k, E_k^n = 1) / \sum_{n=1}^N (l(z_k | x'_k, E_k^n = 1)).$$

Step 5: Resample the particles:  $[\{x_k^n, 1/N\}_{n=1}^N] = \text{systematic-resample}[\{x_k'^n, w_k'^n\}_{n=1}^N]$ .

Step 6: Output the state estimation: an estimate of the target state can be made from the set of particles resulting from the time algorithm above.

The posterior density of target existence at time  $k$  is  $P_k \stackrel{\Delta}{=} P\{E_k = 1 / Z_k\}$ , and then, the probability of existence is given by  $\hat{P}_k = \sum_{n=1}^N E_k^n / N, 0 \leq \hat{P}_k \leq 1$ . Additionally, the target state can be estimated by

$$\hat{x}_k = \sum_{n=1}^N (x_k^n E_k^n) / \sum_{n=1}^N E_k^n .$$

### 4. Simulation Results

We use the following simulated measurement data to test the performance of the proposed algorithm, which is widely used in the current study and some representative PF-TBD algorithms [2–6]. Meanwhile, the CV (constant velocity) and CT (coordinate turn) motion models are used for testing.

The simulated measurement data consists of a total of 30 frames, with the target appearing in the data at Frame 7 and disappearing at Frame 22. Each frame consists of an array of  $20 \times 20$  pixels,  $n = m = 20$ ; each cell has unit dimensions  $\Delta_x = \Delta_y = 1$ . The probabilities of birth and death are set as  $P_b = P_d = 0.05$ . The standard deviation of the target spread function is  $\Sigma = 0.7$ . The target initial state was given by  $x_0 = [4.2 \ 0.45 \ 7.2 \ 0.25 \ 20]^T$ . The birth particles proposal functions  $x_k^b \sim U(0, n)$ ,  $y_k^b \sim U(0, m)$ ,  $\dot{x}_k^b \sim U(-1, 1)$ ,  $\dot{y}_k^b \sim U(-1, 1)$  and  $I_k^b \sim U(10, 30)$ . Set the observation noise variance  $\sigma^2$  to get a different SNR, which is defined by  $SNR = 10 \log\left(\frac{I^2}{\sigma^2}\right)$ . For example, if the SNR is 6 dB, the standard deviation of the background noise level in each pixel is  $\sigma = 10$  units. The HDE parameters were set according to [16].  $F_m = 0.9$ ;  $C = 0.6$ ; the annealing initial temperature  $T_{st} = 100$ ; the annealing parameter  $\rho = 0.9$ ; and the pre-specified temperature  $T0 = 5$ .

For the CV model, the transition matrix  $F_{CV}$  is given by Equation (2) and  $Q_{CV}$  by Equation (3), with  $q_1 = 0.001$ ,  $q_2 = 0.01$ ,  $T_{CV} = 1$  and the target detection threshold  $\text{thr} = 0.6$ .

For the CT model, the system dynamics model function is given by Equation (1), and the covariances of noise process  $Q_{CT}$  and the process transition matrix  $F_{CT}$  are defined by:

$$F_{CT} = \begin{bmatrix} 1 & \frac{\sin(w(k)T_{CT})}{w(k)} & 0 & -\frac{1 - \cos(w(k)T_{CT})}{w(k)} & 0 \\ 0 & \cos(w(k)T_{CT}) & 0 & -\sin(w(k)T_{CT}) & 0 \\ 0 & \frac{1 - \cos(w(k)T_{CT})}{w(k)} & 1 & \frac{\sin(w(k)T_{CT})}{w(k)} & 0 \\ 0 & \sin(w(k)T_{CT}) & 0 & \cos(w(k)T_{CT}) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$



$$Q_{CT} = \begin{bmatrix} \frac{2(w(k)T_{CT} - \sin(w(k)T_{CT})) \cdot q_1}{w(k)^3} & \frac{(1 - \cos(w(k)T_{CT})) \cdot q_1}{w(k)^2} & 0 & \frac{(w(k)T_{CT} - \sin(w(k)T_{CT})) \cdot q_1}{w(k)^2} & 0 \\ \frac{(1 - \cos(w(k)T_{CT})) \cdot q_1}{w(k)^2} & q_1 T_{CT} & \frac{-(w(k)T_{CT} - \sin(w(k)T_{CT})) \cdot q_1}{w(k)^2} & 0 & 0 \\ 0 & \frac{-(w(k)T_{CT} - \sin(w(k)T_{CT})) \cdot q_1}{w(k)^2} & \frac{2(w(k)T_{CT} - \sin(w(k)T_{CT})) \cdot q_1}{w(k)^3} & \frac{(1 - \cos(w(k)T_{CT})) \cdot q_1}{w(k)^2} & 0 \\ \frac{(w(k)T_{CT} - \sin(w(k)T_{CT})) \cdot q_1}{w(k)^2} & 0 & \frac{(1 - \cos(w(k)T_{CT})) \cdot q_1}{w(k)^2} & q_1 T_{CT} & 0 \\ 0 & 0 & 0 & 0 & q_2 T_{CT} \end{bmatrix} \quad (18)$$

where  $w(k)$  is the turn rate,  $T_{CT}$  is the sampling period and  $q_1$  and  $q_2$  are the power spectral density of the acceleration noise and the noise in the rate of change of target return intensity, respectively. Here, their values are set the same as in CV model. The turn angle is  $4^\circ$ ,  $T_{CT} = 1$ ; and the target detection threshold  $\text{thr} = 0.7$ .

For convenience, in the experimental results, the proposed algorithm is marked as HDE PF-TBD; PF-TBD with the systematic resampling method is marked as SystematicR PF-TBD; and that using the multinomial resampling method is marked MultinomialR PF-TBD. We use exactly the same set of data and the same filter parameters. The number of particles used by each algorithm was set to 6000. The algorithm performance is gauged on an average over 100 Monte Carlo simulations.

4.1. Scenario-CV

Two sampled measurement images with SNR 6 dB are shown in Figure 1 in different frames: Frames 15 and 18. This shows that the target exists at Frames 15 and 18, but it is indistinguishable from the background noise. If using the TAD algorithm, this will inevitably lead to a loss of information at low SNR.

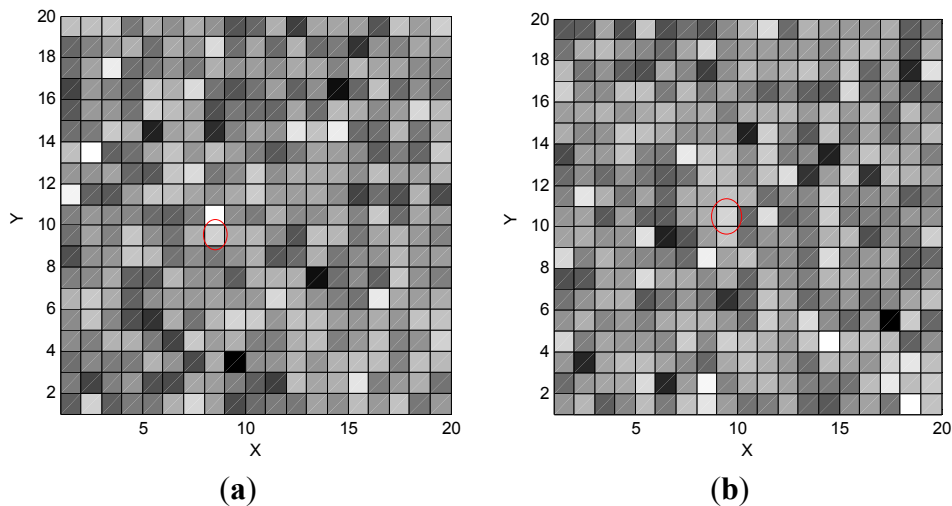


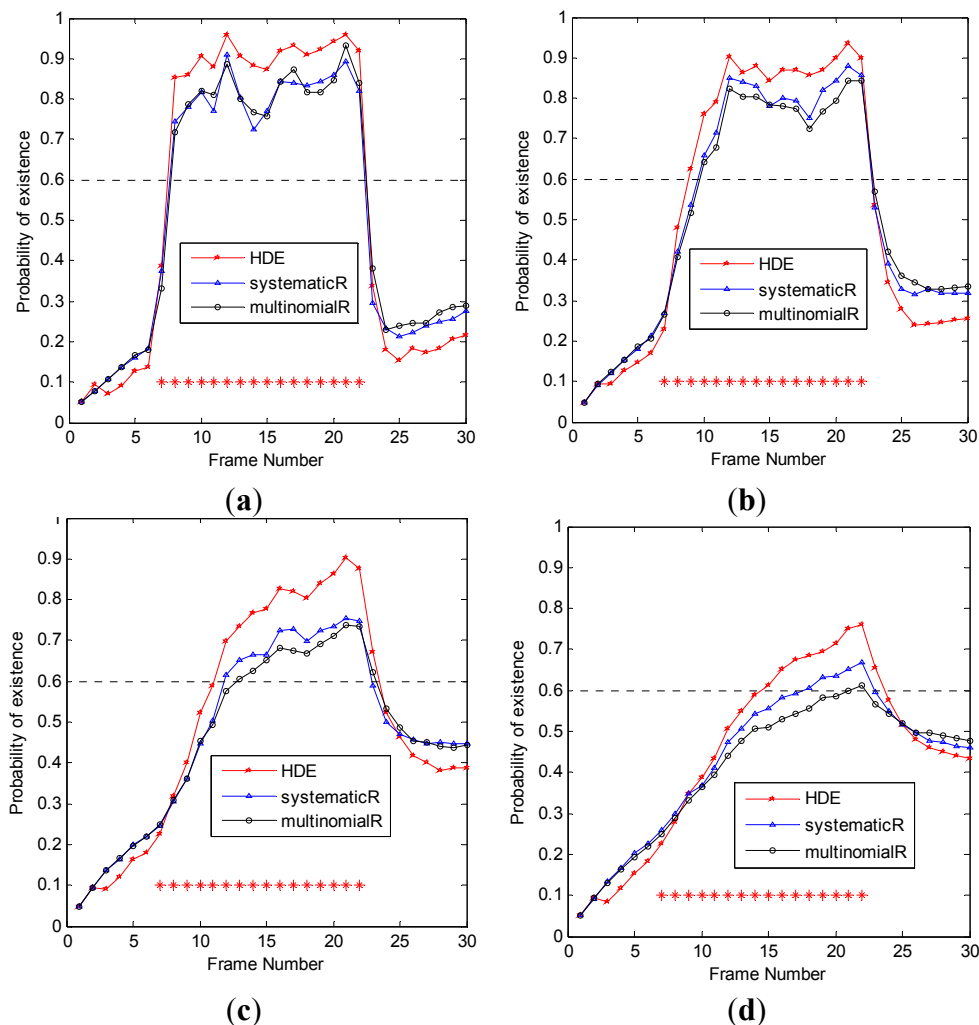
Figure 1. (a) Sampled image at Frame 15; (b) sampled image at Frame 18.

4.1.1. Detection Performance Analysis

The detection performance of the proposed algorithm (HDE PF-TBD) is evaluated in simulations with the probability of existence. The average of the probability of detection with different SNRs of the three algorithms is compared in Table 1. Figure 2 shows the results in terms of the probability of existence at different frame times.

**Table 1.** The average of the probability of detection for the constant velocity (CV) model. R, resampling; PF, particle filter; TBD, track-before-detect; HDE, hybrid differential evolution.

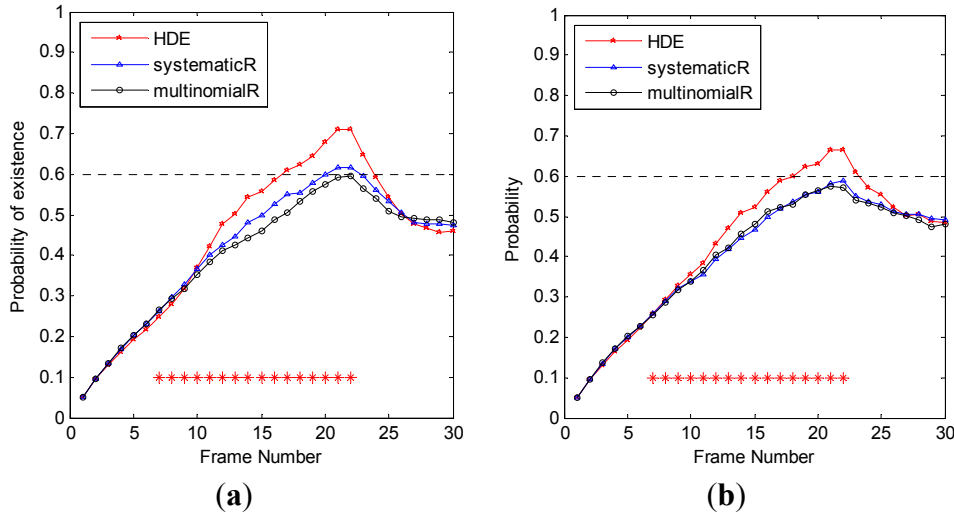
Algorithm	9 dB	6 dB	3 dB	1 dB
MultinomialR PF-TBD	0.770	0.657	0.423	0.217
SystematicR PF-TBD	0.774	0.682	0.498	0.300
HDE PF-TBD	0.875	0.765	0.628	0.395



**Figure 2.** (a) Probability of the existence at SNR = 9 dB for the CV model; (b) probability of existence at SNR= 6 dB for the CV model; (c) probability of existence at SNR = 3 dB for the CV model; (d) probability of existence at SNR = 1 dB for the CV model.

The target exists if the probability of existence is higher than 0.6, which is a tested and sensible value and is plotted as a horizontal line in Figure 2. Moreover, the asterisk signs at the bottom of the figures indicate the time of the presence of the target. It can be seen from Figure 2 that for SNR = 1, 3, 6, 9 dB, three algorithms have target presence and absence declaration delays, and each algorithm follows the true existence of the target much more closely for the higher SNR cases, with performance degrading as SNR reduces. The HDE PF-TBD algorithm provides higher target detection confidence than the other two algorithms. Furthermore, the HDE PF-TBD algorithm performs very well among the four

SNR values, in other words with less target presence and absence declaration delays than others. From Figure 2, we also can see that the probability of the existence of the proposed algorithm has a large increase when the SNR is 3 dB and 1 dB. For clarity, the data statistics of Table 1 have proven that the HDE PF-TBD has better detection sensitivity when the SNR is lower.



**Figure 3.** (a) Probability of existence at SNR = 0 dB for the CV model; (b) probability of existence at SNR = -1 dB for the CV model.

Moreover, it can be seen from Figure 3 that for SNR = 0, -1 dB, all three algorithms suffer a poor detection performance. However, compared to SystematicR PF-TBD and the MultinomialR PF-TBD, the HDE PF-TBD algorithm can detect the target at the end of the frames where the target exists. It is demonstrated that the proposed approach has a larger detectable and traceable SNR range and is capable of resolving large noise in the target. Meanwhile, it also proves that the improvements in the proposed algorithm can effectively enhance the detection sensitivity and that the proposed algorithm is more suitable for detecting weak targets with low SNR.

#### 4.1.2. Tracking Performance Analysis

To measure the estimation accuracy of the proposed algorithm, we introduce the root mean squared error ( $RMSE$ ) in the position and its mean  $\overline{RMSE}$  for different SNRs. Additionally, the  $RMSE$  for  $M$  simulations with different frames at time  $k$  is defined as:

$$RMSE = \left[ \frac{1}{M} \sum_{m=1}^M \left( (\hat{x}_{k,m} - x_{k,m})^2 + (\hat{y}_{k,m} - y_{k,m})^2 \right) \right]^{1/2} \tag{19}$$

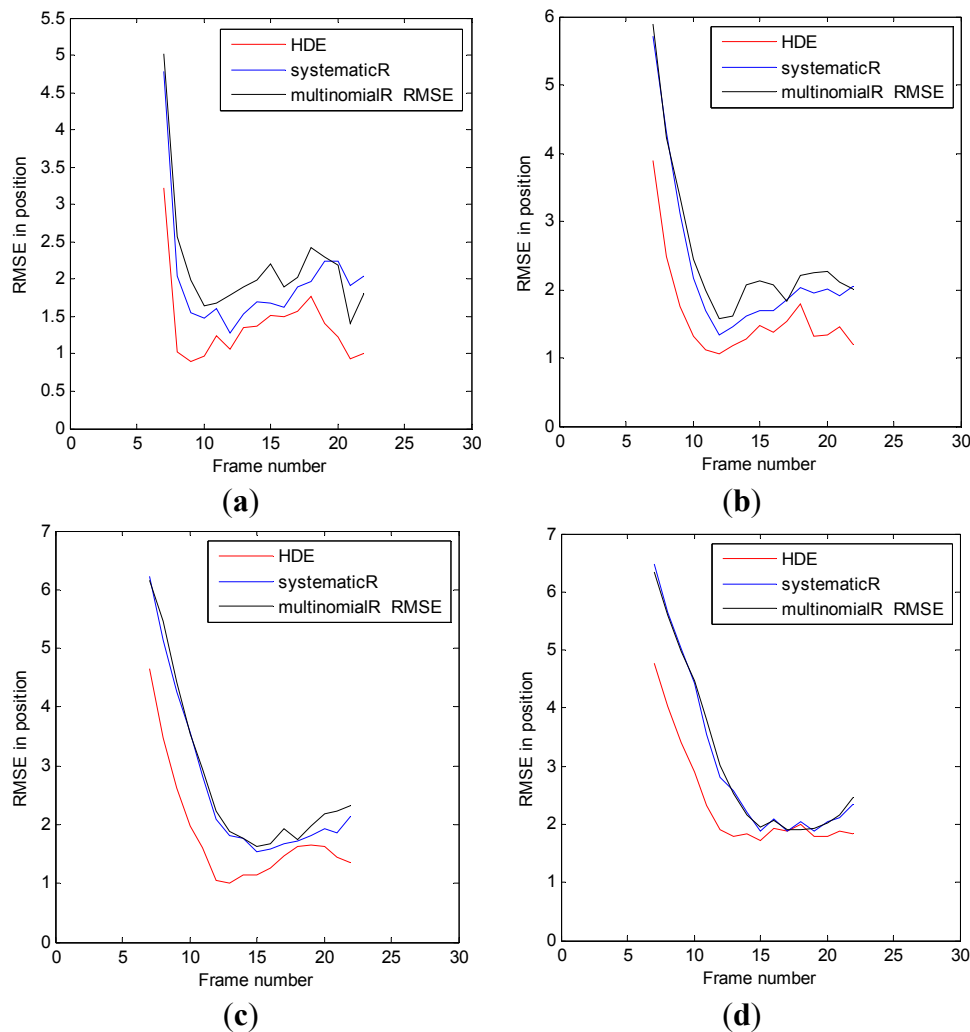
where  $x_{k,m}$  and  $y_{k,m}$  are the true values of the target state and  $\hat{x}_{k,m}$  and  $\hat{y}_{k,m}$  are defined as the estimation of the target state. We only consider the frames in which the target exists, and the period length is  $L$ . The  $\overline{RMSE}$  can be shown as:

$$\overline{RMSE} = \frac{1}{L} \sum_{i=1}^L RMSE_i \tag{20}$$

The results in terms of mean RMSE for the different SNR values of the three algorithms are compared in Table 2. Figure 4 shows the results in terms of RMSE in position with corresponding frames for different SNR values.

**Table 2.** The average of the RMSE for different SNRs for the CV model.

Algorithm	9 dB	6 dB	3 dB	1 dB
MultinomialR PF-TBD	2.1757	2.4715	2.7528	3.0814
SystematicR PF-TBD	1.9771	2.3427	2.6222	3.0622
HDE PF-TBD	1.3810	1.5971	1.8221	2.3595



**Figure 4.** (a) RMSE in position at SNR = 9 dB for the CV model; (b) RMSE in position at SNR = 6 dB for the CV model; (c) RMSE in position at SNR = 3 dB for the CV model; (d) RMSE in the position of existence at SNR = 1 dB for the CV model.

As expected, the tracking accuracy of each algorithm degrades as the SNR reduces. Meanwhile, the proposed algorithm shows a better performance on the low SNR target than the other two algorithms in Figure 4. It is can be seen that the proposed algorithm has much lower RMSE when the target began to appear. It is demonstrated that the HDE PF-TBD algorithm can more accurately estimate the initial location of the target appearance. Moreover, from Table 2, it can be clearly seen that the estimation

accuracy of the proposed algorithm is always superior to that of the other two algorithms. Although the RMSE of the proposed method approximates the other two algorithms at Frames 16 and 17 when the SNR is 1 dB, the tracking accuracy of the HDE PF-TBD algorithm is still higher than the other two algorithms in the whole frame range where the target exists. Therefore, the proposed algorithm is capable of tracking the target with low SNR, and it outperforms the SystematicR PF-TBD and the MultinomialR PF-TBD.

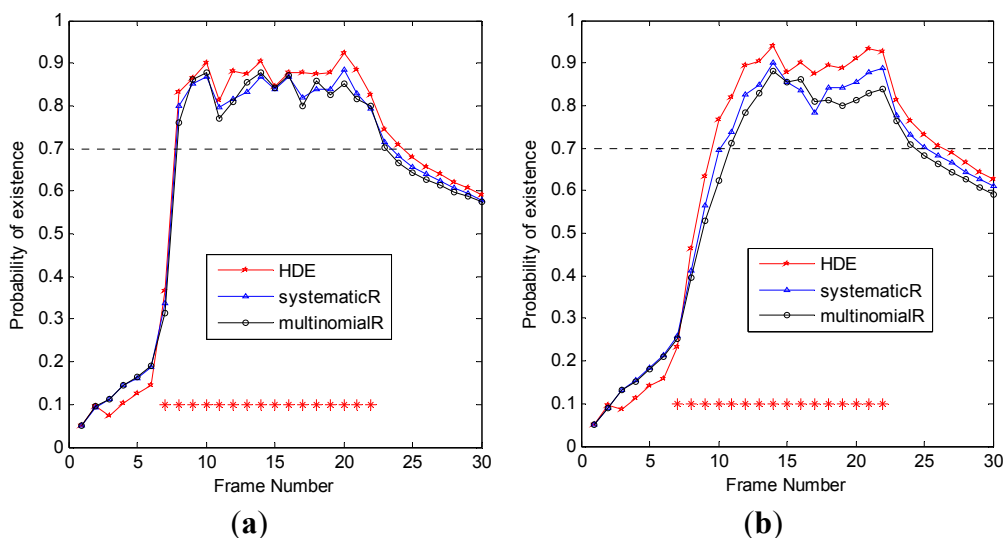
4.2. Scenario-CT

Similarly, we also analyze detection and tracking performance under the CT model. The average of the detection probability of detection with different SNRs of the three algorithms for 100 Monte Carlo simulations (MC) is compared in Table 3. Figure 5 shows the results in terms of the probability of existence in different frame times for 100 MC.

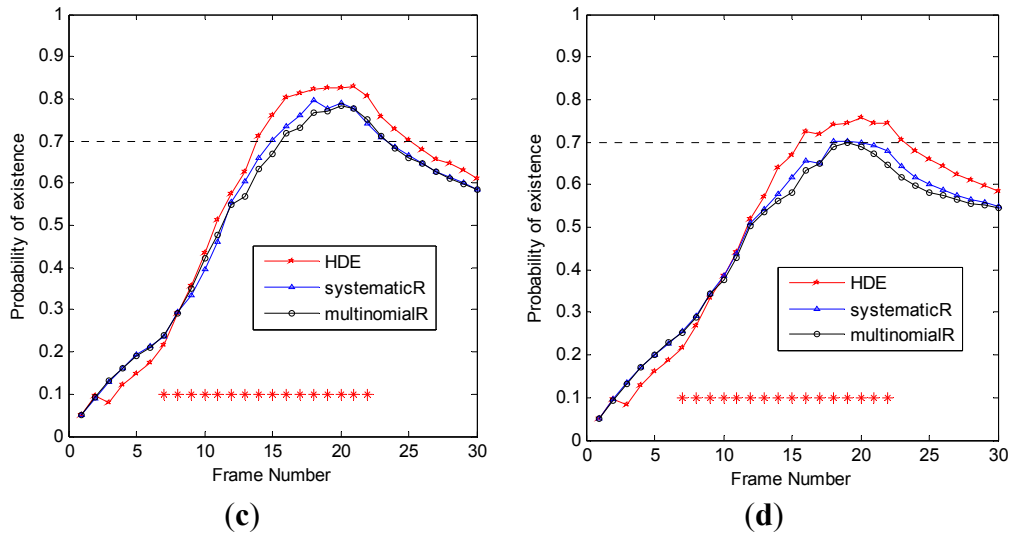
**Table 3.** The average detection probability for the CT model.

Algorithm	9 dB	6 dB	3 dB	1 dB
MultinomialR PF-TBD	0.747	0.621	0.378	0.275
SystematicR PF-TBD	0.768	0.648	0.399	0.284
HDE PF-TBD	0.827	0.743	0.501	0.367

The target exists if the probability of existence is higher than 0.7, which is plotted as a horizontal line in Figure 5. It can be seen from Figure 5 that, for SNR =1, 3, 6, 9 dB, the three algorithms also have target presence and absence declaration delays, and each algorithm follows the true existence of the target much more closely for the higher SNR cases, with performance degrading as SNR reduces under the CT model. For the CT model, the HDE PF-TBD algorithm provides higher target detection confidence than the other two algorithms. Moreover, the HDE PFTBD algorithm has less target presence delays than the others. Meanwhile, we also can see that the probability of existence of the proposed algorithm has a large increase when the SNR is 3 dB and 1 dB. For clarity, the data statistics of Table 3 have proven that the HDE PF-TBD has better detection sensitivity when the SNR is lower.



**Figure 5.** Cont.

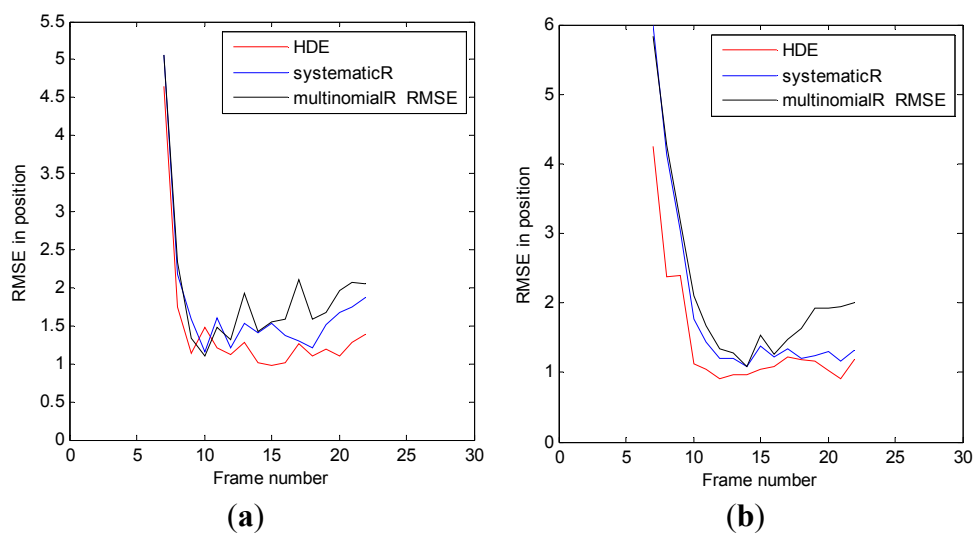


**Figure 5.** (a) Probability of existence at SNR = 9 dB for the coordinate turn (CT) model; (b) probability of existence at SNR = 6 dB for the CT model; (c) probability of existence at SNR = 3 dB for the CT model; (d) probability of existence at SNR = 1 dB for the CT model.

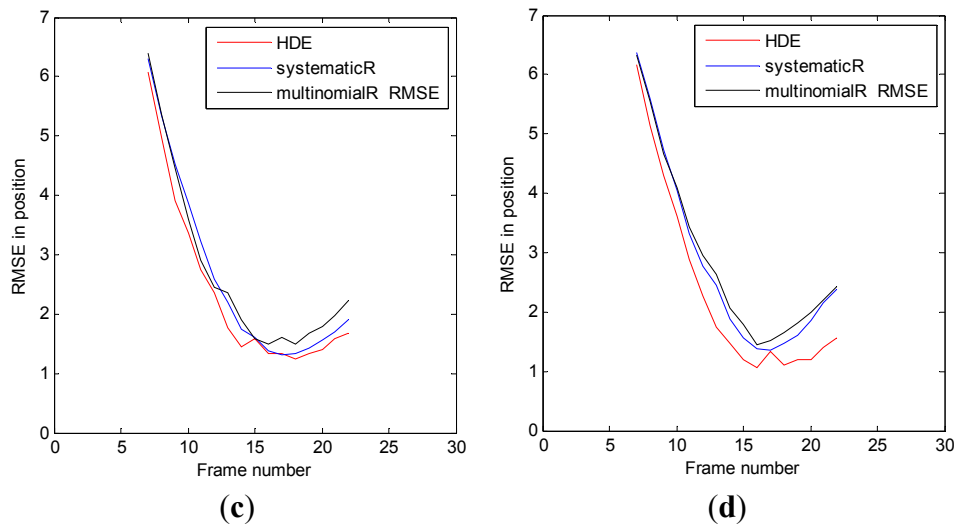
**Table 4.** The average RMSE at different SNRs for the CT model.

Algorithm	9 dB	6 dB	3 dB	1 dB
MultinomialR PF-TBD	1.910	2.1556	2.7044	2.909
SystematicR PF-TBD	1.745	1.8739	2.6258	2.81
HDE PF-TBD	1.433	1.4477	2.3836	2.3541

The results in terms of mean RMSE for the different SNR values of the three algorithms for the CT model are compared in Table 4. Figure 6 shows the results in terms of RMSE in position with corresponding frames for different SNR values for the CT model.



**Figure 6.** Cont.



**Figure 6.** (a) RMSE in position at SNR = 9 dB for the CT model; (b) RMSE in position at SNR = 6 dB for the CT model; (c) RMSE in position at SNR = 3 dB for the CT model; (d) RMSE in the position of existence at SNR = 1 dB for the CT model.

For the CT model, it can be clearly seen from Table 4 and Figure 6 that the proposed algorithm also shows a better performance on the low SNR target than the other two algorithms. Although the RMSE of the proposed method is a little higher than the other two algorithms at Frame 10 when SNR is 9 dB, the tracking accuracy of the HDE PF-TBD algorithm is still higher than other two algorithms in the whole frame range that the target exists. Therefore, as can be seen from the two aspects of detection and tracking, the proposed algorithm is also capable of tracking the target with the CT motion model and low SNR, and it outperforms SystematicR PF-TBD and MultinomialR PF-TBD.

## 5. Conclusions

In this paper, a novel particle filter track-before-detect algorithm based on hybrid differential evolution was developed. Firstly, we exploit a hybrid differential evolution as a novel particle updating strategy for overcoming the shortcoming of the particle impoverishment. Secondly, in order to reduce the influence of the particle degeneracy, the systematic resampling method is chosen for the proposed method. Finally, the proposed algorithm combines a more effective particle initiation method, which only places the particles in the highest intensity bins of the data. The proposed algorithm not only can make full use of the data information, but also decreases the particle degeneracy and enhances the diversity of particles. The detection and tracking of the performance advantage of the proposed algorithm have been demonstrated by two cases of a low signal-to-noise point target against a background of Gaussian noise with the CV and CT motion models, respectively. What is more, the proposed method is much more efficient than previous methods. Future work will continue to enhance the detection accuracy of the proposed algorithm and to explore the implementation of multiple targets within this framework.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61172159) and the Fundamental Research Funds for the Central Universities (HEUCFT1101).

## Author Contributions

The work presented here was carried out in collaboration between both authors. Chaozhu Zhang and Lin Li defined the research theme. Lin Li designed the methods and experiments, carried out the laboratory experiments, analyzed the data, interpreted the results and wrote the paper. Both authors have contributed to, seen and approved the manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Hadzagic, M.; Michalska, H.; Lefebvre, E. Track-before-detect methods in tracking low-observable targets: A survey. *Sens. Trans. Mag.* **2005**, *54*, 374–380.
2. Rutten, M.; Gordon, N.; Maskell, S. Recursive track-before-detect with target amplitude fluctuations. *IEE Proc. Radar Sonar Navig.* **2005**, *152*, 345–352.
3. Pohlig, S.C. An algorithm for detection of moving optical targets. *IEEE Trans. Aerosp. Electron. Syst.* **1989**, *25*, 56–63.
4. Carlson, B.D.; Evans, E.D.; Wilson, S.L. Search radar detection and track with the hough transform. *IEEE Trans. Aerosp. Electron. Syst.* **1994**, *30*, 102–108.
5. Huang, D.; Xue, A.; Guo, Y. Penalty dynamic programming algorithm for dim targets detection in sensor systems. *Sensors* **2012**, *12*, 5028–5046.
6. Grossi, E.; Lops, M.; Venturino, L. A Novel dynamic programming algorithm for track-before-detect in radar systems. *IEEE Trans. Signal. Process.* **2013**, *61*, 2608–2619.
7. Ristic, B.; Arulampalam, S.; Gordon, N. *Beyond the Kalman. Filter-Particle Filters for Tracking Applications*; Artech House: London, UK, 2004.
8. Salmond, D.J.; Birch, H. A Particle Filter for Track-Before-Detect. In Proceedings of the American Control Conference, Arlington, VA, USA, 25–27 June 2001; pp. 3755–3760.
9. Boers, Y.; Driessen, H. Particle Filter Based Detection for Tracking. In Proceedings of the American Control Conference, Arlington, VA, USA, 25–27 June 2001; pp. 4393–4397.
10. Davey, S.J.; Rutten, M.G. A Comparison of Three Algorithms for Tracking Dim Targets. In Proceedings of the Information, Decision and Control (IDC'07), Queensland, Australia, 12–14 February 2007; pp. 342–347.
11. Davey, S.; Rutten, M.; Cheung, B. Using phase to improve track before detect. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 832–849.
12. Boers, Y.; Driessen, J. Multitarget particle filter track before detect application. *IEE Proc. Radar Sonar Navig.* **2004**, *151*, 351–357.



13. Lepouter, A.; Rabaste, O. A Particle Filter for Target Arrival Detection and Tracking in Track-Before-Detect. In Proceedings of the Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF'12), Bonn, Germany, 4–6 September 2012; pp. 13–18.
14. Lu, J.; Shui, P.-L. Track-before-detect method based on cost-reference particle filter in non-linear dynamic systems with unknown statistics. *IET Signal. Process.* **2014**, *8*, 85–94.
15. Hu, Z.B.; Xiong, S.W. Study of hybrid differential evolution based on simulated annealing. *Comput. Eng. Des.* **2007**, *28*, 1989–1992.
16. Zhang, C.; Li, L. Hybrid differential evolution particle filter for nonlinear filtering. *Appl. Comput. Electromagn. Soc.* **2014**, *12*, 1133–1139.
17. Douc, R.; Cappe, O. Comparison of Resampling Schemes for Particle Filtering. In Proceedings of the Image and Signal Processing and Analysis, Zagreb, Croatia, 15–17 September 2005; pp. 64–69.

© 2015 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 license (<http://creativecommons.org/licenses/by/4.0/>).