



# Article Performance Study of Distance-Weighting Approach with Loopy Sum-Product Algorithm for Multi-Object Tracking in Clutter

Pranav U. Damale <sup>1,\*</sup>, Edwin K. P. Chong <sup>1</sup> and Tian J. Ma <sup>2</sup>

- <sup>1</sup> Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA; edwin.chong@colostate.edu
- <sup>2</sup> Sandia National Laboratories, Albuquerque, NM 87123, USA; tma@sandia.gov
- Correspondence: pdamale@rams.colostate.edu; Tel.: +1-970-214-4768

Abstract: In this paper, we explore the performance of the distance-weighting probabilistic data association (DWPDA) approach in conjunction with the loopy sum-product algorithm (LSPA) for tracking multiple objects in clutter. First, we discuss the problem of data association (DA), which is to infer the correspondence between targets and measurements. DA plays an important role when tracking multiple targets using measurements of uncertain origin. Second, we describe three methods of data association: probabilistic data association (PDA), joint probabilistic data association (JPDA), and LSPA. We then apply these three DA methods for tracking multiple crossing targets in cluttered environments, e.g., radar detection with false alarms and missed detections. We are interested in two performance metrics: tracking accuracy and computation time. LSPA is known to be superior to PDA in terms of the former and to dominate JPDA in terms of the latter. Last, we consider an additional DA method that is a modification of PDA by incorporating a weighting scheme based on distances between position estimates and measurements. This distance-weighting approach, when combined with PDA, has been shown to enhance the tracking accuracy of PDA without significant change in the computation burden. Since PDA constitutes a crucial building block of LSPA, we hypothesize that DWPDA, when integrated with LSPA, would perform better under the two performance metrics above. Contrary to expectations, the distance-weighting approach does not enhance the performance of LSPA, whether in terms of tracking accuracy or computation time.

**Keywords:** multiobject tracking; high-clutter tolerant multiobject tracking; probabilistic data association (PDA); joint probabilistic data association (JPDA); loopy sum-product algorithm (LSPA)

# 1. Introduction

Since probabilistic data association (PDA) was introduced for tracking in a cluttered environment [1], engineers have been trying to optimize the data association (DA) technique for implementing in a Kalman filter (KF) [2] tracker. With the inception of the sum-product algorithm (SPA) [3], a new graphical approach for KF tracking was developed that is more efficient in dealing with the complex problem of multiple target tracking (MTT). To develop a robust algorithm that is also scalable for tracking multiple objects in clutter, in this paper we examine the performance of the distance-weighting probabilistic data association (DWPDA) [4] in conjunction with the loopy sum-product algorithm (LSPA) [5].

The problem of DA is finding the correspondence between the targets and the measurements of uncertain origins. There are several approaches to tackle this problem, such as the nearest neighbor (NN), PDA, and joint probabilistic data association (JPDA) [1,6,7]. The NN approach is one of the easiest and uses at any time only the nearest measurement to the predicted measurement as if it were the one originated from the target of interest [6]. The NN approach is suboptimal and can only work well in case of widely spaced targets, accurate measurements, and few false alarms. Finding some association between all the



Citation: Damale, P.U.; Chong, E.K.P.; Ma, T.J. Performance Study of Distance-Weighting Approach with Loopy Sum-Product Algorithm for Multi-Object Tracking in Clutter. *Sensors* **2021**, *21*, 2544. https:// doi.org/10.3390/s21072544

Academic Editor: Taek Lyul Song

Received: 18 February 2021 Accepted: 30 March 2021 Published: 5 April 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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 (https:// creativecommons.org/licenses/by/ 4.0/). targets and measurements is computationally expensive. Therefore a gate is formed around the predicted measurement based on some predefined threshold called the validation region. The PDA algorithm obtains the probability of each measurement lying inside the validation region as being the correct measurement and updates the state estimates according to an appropriately modified tracking filter, called PDAF [1]. One major drawback of the PDA algorithm is that it treats every measurement inside the validation region of a target of interest as if it were originated from the said target or as a Poisson-distributed false alarm. PDA does not take into account the possibility that a validated measurement for one target might be a measurement originated from another nearby target. The JPDA algorithm improves this deficiency of the PDA algorithm by computing the association probabilities for the validated measurements from the joint likelihood functions corresponding to all feasible joint events such that no more than one measurement originates from each target [7]. However, with an increasing number of targets and/or clutter, the JPDA algorithm becomes impractical for real-time applications due to its combinatorial complexity because it considers all feasible joint events of measurements to targets to obtain the joint association probabilities [8].

PDA and JPDA are zero-scan algorithms, meaning that all hypotheses are combined after computation of the probabilities, for each target at each time step [1,7]. Alternatively, multiple hypothesis tracking (MHT) is a deferred decision logic algorithm. In the case of a conflicting target-measurement association, the MHT algorithm formulates alternative data association hypotheses instead of choosing the best-combined hypothesis. The ambiguities between the alternative hypotheses are resolved using the measurements that arrive later into the future [9]. JPDA is shown to produce reasonable results compared to the computationally expensive multi-scan MHT algorithm [7–10]. Both the PDA and the JPDA algorithms utilize known targets for forming validation gates in the measurement space to compute the posterior probabilities and are therefore categorized as target-oriented approaches. There exist measurement-oriented algorithms, such as the one described in [11] and others, where hypotheses are formed for each measurement to have originated from a known target, a new target, or clutter. Both sets of algorithms have shown to yield an equivalent expression for posterior probabilities with appropriate assumptions [7,11,12].

As we realize that calculating DA probabilities for MTT is a process that involves a complicated global function of many variables that can be broken down into a product of local functions each of which depends on a subset of the variables, we look for alternate methods that can exploit this trait. SPA is one such method that operates in a factor graph [13] and attempts to compute, either exactly or approximately, various marginal functions associated with the global function [3]. SPA operates by passing messages, called beliefs, between the nodes of a factor graph, i.e., variables and local functions, that involve summations and products of factors. Implementation of a factor graph for the MTT DA problem requires a loopy sum-product solution which is neither guaranteed to converge nor produce the correct marginal functions if convergence occurs. A simultaneous target-oriented and measurement-oriented factor graph formulation of the DA problem has been shown that is guaranteed to converge and results in accurate association probabilities [5,14]. A simplified implementation of LSPA results in a significant reduction in computational complexity without loss of accuracy [14,15], which makes LSPA more appealing than PDA or JPDA for DA.

In recent years, the use of LSPA for DA in MTT has been gaining traction. A beliefpropagation approach to multi-target multi-sensor tracking is proposed in [16] by formulating a detailed factor graph where every single target and data-association variable is modeled as an individual node. To tackle the problem of tracking an unknown number of targets, where targets appear and disappear, an LSPA-based method is proposed in [17] that creates augmented target states for keeping track of existing and non-existing targets. A more comprehensive derivation of the message-passing algorithm for multi-sensor multi-target tracking is given in [18,19]. An extension of the LSPA-based MTT framework for target estimation is given in [20] that exploits additional target information provided by a classifier. All these efforts highlight the potential of SPA for solving the complicated problem of DA in MTT.

The data association probabilities obtained using PDA are a crucial part of the beliefs being passed between nodes during the convergence of LSPA. Modifying the DA probabilities according to a weighting scheme based on distances between the predicted and validated measurements has been shown to enhance the tracking accuracy of PDAF while tracking a single target in a densely cluttered environment [4]. In this paper, we want to explore whether this distance-weighting approach for PDA, when integrated with LSPA, would improve the performance of LSPA even further. To evaluate this possibility, we formulate a distance-weighting LSPA (DWLSPA) and compare its performance in terms of tracking accuracy and computation time against DWPDA, JPDA, and LSPA for tracking multiple targets crossing at a small angle in different density clutter. Our results turn out to be contrary to expectations.

The main contribution of this performance study is to explore the idea of modifying one of the building blocks for a state-of-the-art data-association algorithm [15] for multitarget tracking and to compare the tracking accuracy of the modified algorithm to that of the original algorithm. The distance-weighting modification analyzed in this paper is based on a recent successful implementation of a similar modification to one of the earliest data-association frameworks for tracking targets in cluttered environments [4]. In doing so, we develop the mathematical formulation for each data-association filter being considered and evaluate its performance in terms of tracking accuracy and computation time over a wide range of easy-to-replicate multi-object-tracking scenarios.

We introduce our notations and the target tracking system in Section 2. In Section 3, we describe the problem of PDA, DWPDA, JPDA, and LSPA. We present simulation results for these methods and our analysis in Section 4. In Section 5, we conclude the paper with our observations and final remarks regarding the performance of these methods.

## 2. Target Tracking Dynamic System Model and Assumptions

We describe the classic data association problem in which a single sensor surveils a large number of targets. The number of targets under surveillance is assumed to be known and is denoted by  $N_T$ . The measurements are comprised of possible target detections and false alarms. A target is detected with a known probability of detection  $P_D$  and is independent of time. The false alarms, modeled according to the Poisson point process with a known spatial density  $\lambda$ , are uniformly distributed in the measurement space. A validation region, with the threshold  $\gamma$  corresponding to certain gate probability  $P_G$ , is set up at every sampling time around the predicted measurement and possibly several measurements fall in it. Each algorithm differs in how these measurements are used (or not) in the estimation of the state of the target. We assume that each target can generate at most one measurement and each measurement can have only one source.

We denote by  $x_i(k)$ ,  $i \in \{1, ..., N_T\}$ , the state of *i*-th target of dimension  $n_x$  at time k. The complete set of target states at time k is denoted by  $X(k) = (x_1(k), ..., x_{N_T}(k))$ . At time k, the total number of measurements is denoted by M(k). We denote by  $z_j(k)$ ,  $j \in \{1, ..., M(k)\}$ , the value of *j*-th measurement of dimension  $n_z$ . The complete set of measurements at time k is denoted by  $Z(k) = (z_1(k), ..., z_{M(k)}(k))$ , and the complete set of measurements up to and including time k is denoted by  $Z^k = (Z(1), ..., Z(k))$ .

The state and measurement equations are assumed linear with additive zero-mean white noise with known covariances. The state of target t evolves in time according to the equation

$$x_t(k+1) = F(k)x_t(k) + G(k)u(k),$$
(1)

and the true measurement for target *t* is given by

$$z_t(k) = H(k)x_t(k) + w(k)$$
(2)

where u(k) and w(k) are zero-mean mutually independent white Gaussian noise sequences with known covariances Q(k) and R(k), respectively. Functions F(k), G(k), and H(k)are known matrices for state transition, noise gain, and sensor, respectively. The past information (through time k - 1) about the target t is assumed to be known and summarized approximately by the Gaussian posterior

$$p[x_t(k-1)|Z^{k-1}] = \mathcal{N}[x_t(k-1); \hat{x}_t(k-1|k-1), P_t(k-1|k-1)]$$
(3)

where  $\hat{x}_t(k-1|k-1)$  and  $P_t(k-1|k-1)$  are the state estimate and covariance for target *t*.

## 3. Algorithm Description

# 3.1. Probabilistic Data Association Filter

The PDA algorithm calculates the association probabilities for each validated measurement at the current time for the target of interest. PDA assumes that all the validated measurements are generated by either the target of interest or clutter. The association probabilities are used for calculating the mean squared error (MSE) estimate and covariance of the target's state. An appropriately modified KF, called PDAF, is used to account for the uncertainty of origins for the validated measurements while estimating the state of the target. The algorithm can be given as follows.

#### 3.1.1. Prediction

The prediction of the state and measurement of target t at time k is done as in the KF, i.e.,

$$\hat{x}_t(k|k-1) = F(k-1)\hat{x}_t(k-1|k-1)$$
(4)

$$\hat{z}_t(k|k-1) = H(k)\hat{x}_t(k|k-1).$$
(5)

The covariance of the predicted state for target *t* is

$$P_t(k|k-1) = F(k-1)P_t(k-1|k-1)F(k-1)' + G(k-1)Q(k-1)G(k-1)'.$$
 (6)

Here,  $\hat{x}_t(k-1|k-1)$  and  $P_t(k-1|k-1)$  are available from Equation (3). The innovation covariance of the target *t* (for the correct measurement) is

$$S_t(k) = H(k)P_t(k|k-1)H(k)' + R(k).$$
(7)

## 3.1.2. Measurement Validation

The validation region for target *t* at time *k* is the elliptical region given by

$$\mathcal{V}_t(k,\gamma) = \{ z \in Z(k) : [z - \hat{z}_t(k|k-1)]' S_t(k)^{-1} [z - \hat{z}_t(k|k-1)] \le \gamma \}.$$
(8)

Here,  $\gamma$  can be obtained according to

$$V(k) = c_{n_z} |\gamma S(k)|^{1/2}$$
(9)

where V(k) is the volume of the validation region given by Equation (8) and  $c_{n_z}$  is the volume of the unit hypersphere of dimension  $n_z$  [12]. The validated measurements for target *t* according to Equation (8) are

$$Z_t(k) \stackrel{\Delta}{=} \{z_i(k)\}_{i=1}^{m_t(k)} \tag{10}$$

where  $m_t(k)$  is the number of validated measurements for target *t* at time *k*.

## 3.1.3. Data Association Probabilities

The association probability  $\beta_i^t$  for each validated measurement of target t is obtained as

$$\beta_{i}^{t}(k) = \begin{cases} \frac{\mathcal{L}_{i}(k)}{1 - P_{D}P_{G} + \sum_{j=1}^{m_{t}(k)} \mathcal{L}_{j}(k)} & i = 1, \dots, m_{t}(k) \\ \frac{1 - P_{D}P_{G}}{1 - P_{D}P_{G} + \sum_{j=1}^{m_{t}(k)} \mathcal{L}_{j}(k)} & i = 0 \end{cases}$$
(11)

where i = 0 indicates probability of associating none of the validated measurements to the target. In Equation (11),  $\mathcal{L}_i(k)$  the likelihood ratio (LR) of measurement  $z_i(k)$  originating from the target t vs. from clutter and is obtained as

$$\mathcal{L}_{i}(k) \stackrel{\Delta}{=} \frac{\mathcal{N}[z_{i}(k); \hat{z}_{t}(k|k-1), S_{t}(k)]P_{D}}{\lambda}, \quad z_{i}(k) \in Z_{t}(k).$$
(12)

## 3.1.4. State Estimation

The state estimation for target *t* is done according to PDAF by

$$\hat{x}_t(k|k) = \hat{x}_t(k|k-1) + W_t(k)v_t(k)$$
(13)

where the combined innovation for target *t* is

$$v_t(k) = \sum_{i=0}^{m_t(k)} \beta_i^t(k) v_i^t(k),$$
(14)

and

$$v_i^t(k) = z_i(k) - \hat{z}_t(k|k-1)$$
(15)

is the innovation of measurement for  $z_i(k) \in Z_t(k)$ . The filter gain is calculated as

$$W_t(k) = P_t(k|k-1)H(k)'S_t(k)^{-1}.$$
(16)

The covariance estimation for target *t* associated with the updated state is

$$P_t(k|k) = \beta_0(k)P_t(k|k-1) + [1 - \beta_0^t(k)]P_t^c(k|k) + \tilde{P}_t(k)$$
(17)

where the covariance of the state updated with the correct measurement,  $P^c$ , and the innovation spread,  $\tilde{P}$ , for target *t* are given by

$$P_t^c(k|k) = P_t(k|k-1) - W_t(k)S_t(k)W_t(k)'$$
(18)

and

$$\tilde{P}_t(k) = W_t(k) \left[ \sum_{i=1}^{m_t(k)} \beta_i^t(k) v_i^t(k) v_i^t(k)' - v_t(k) v_t(k)' \right] W_t(k)',$$
(19)

respectively.

The estimated state in PDAF from Equation (13) is for a single target  $x_t$ . To estimate the state of every target  $x_t$ ,  $t \in \{1, ..., N_T\}$ , we need to apply PDAF to the targets one by one sequentially. This means that for each target  $x_t$ ,  $t \in \{1, ..., N_T\}$ , we perform state and measurement prediction, form a validation region around the predicted measurement and prune off potentially unrelated measurements, obtain data-association probabilities for the validated measurements, and, finally, update the state estimate according to Equation (13). The order in which PDAF is applied to the targets is irrelevant because the outcome is independent of the order.

## 3.2. Distance-Weighting Probabilistic Data Association Filter

In the PDA algorithm, the association probability  $\beta$  is calculated as the likelihood ratio of a validated measurement to have originated from a target vs. from clutter. Chen et al. [4] have pointed out that a true measurement from a target of interest is more likely to be near the target's predicted measurement. Since false alarms are uniformly distributed in the measurement space, the measurement nearest to that of the predicted measurement from a target of interest should carry more weight while calculating the association probabilities for the target. The distance weight proposed in [4] is calculated as

$$\Delta_{i}^{t}(k) = \frac{1/\delta_{i}^{t}(k)}{\sum_{i=1}^{m_{t}(k)} 1/\delta_{i}^{t}(k)}$$
(20)

where  $\delta_i^t(k)$  is the Mahalanobis distance between validated measurement *i* and predicted measurement of target *t* at time *k*. The Mahalanobis distance is calculated as the norm of the innovation squared and is given by

$$\delta_i^t(k) = v_i^t(k)' S_t(k)^{-1} v_i^t(k)$$
(21)

where  $v_i^t(k)$  is the innovation as defined in Equation (15).

The new data association probabilities for target *t* at time *k* are calculated as

$$\mathcal{B}_i^t(k) = \beta_i^t(k)\Delta_i^t(k) \quad i = 1, \dots, m_t(k)$$
(22)

and are normalized according to

$$\mathcal{B}_{i}^{t}(k) = \mathcal{B}_{i}^{t}(k) / \sum_{j=0}^{m_{t}(k)} \mathcal{B}_{i}^{t}(k) \quad i = 0, 1, \dots, m_{t}(k).$$
(23)

The data association probabilities obtained in Equation (23) are used for estimating target state as explained previously in PDAF Section 3.1.4. Similar to PDAF, the DWPDA filter is designed for tracking a single target. For the purpose of tracking multiple targets, we need to update the states of the targets one by one.

#### 3.3. Joint Probabilistic Data Association Filter

The PDA algorithm is designed for tracking a single target in clutter. Because the PDA algorithm assumes all the incorrect measurements in the validation region of a target of interest are clutter, it is susceptible to scenarios where these incorrect measurements might have originated from another nearby target. Situations may arise in MTT where the validation regions of nearby targets may overlap for several time frames in a row and cause persistent interference that can lead to track deterioration or track loss altogether with PDA. JPDA improves on PDA by calculating joint association probabilities at each time step. Marginalized association probabilities for each target can be obtained using these joint association probabilities, which are then used for estimating the state of each target.

## 3.3.1. Measurement Validation

A major difference between the PDA and the JPDA algorithm is that no individual validation gates will be assumed for the various targets. A uniform validation region for all targets is obtained by taking a union of individual validation gates. This way each measurement is assumed to be validated for each target and false alarms will be assumed to be uniformly distributed across the entire validation region. Predictions for each target are done at each time step similar to the PDA algorithm using Equations (4)–(7).

Once the validated measurements are obtained for each target using Equations (8) and (10), the combined validated measurements at time k are given according to

$$Z_{N_T}(k) = \bigcup_{i=1}^{N_T} Z_i(k).$$
 (24)

The number of combined validated measurements is  $m_{N_T}(k)$  at time *k*.

## 3.3.2. The Validation Matrix

A validation matrix  $\Omega$  of size  $m_{N_T} \times (N_T + 1)$  with binary elements is created using the combined validated measurement.

$$\Omega(k) = [\omega_{jt}] \quad j = 1, \dots, m_{N_T}(k); \quad t = 0, 1, \dots, N_T$$
(25)

where

$$\omega_{jt} = \begin{cases} 1 & \text{if } z_j(k) \in Z_t(k) \\ 0 & \text{otherwise.} \end{cases}$$
(26)

The first column of  $\Omega(k)$  corresponding to t = 0 is all unity indicating that each measurement could be a false alarm.

## 3.3.3. The Feasible Joint Events

The joint association events for time k are given by

$$\theta(k) = \bigcap_{j=1}^{m_{N_T}} \theta_{jt_j} \quad j = 1, \dots, m_{N_T}(k); \quad t = 0, 1, \dots, N_T$$
(27)

where in the event of  $\theta_{jt}$ , measurement *j* is originated from target  $t_j$ . An event matrix  $\hat{\Omega}$  consisting of the units in  $\Omega$  corresponding to the association in  $\theta$  is used to represent a joint association event  $\theta$ .

$$\hat{\Omega}(\theta(k)) = [\hat{w}_{jt}(\theta(k))] \quad j = 1, \dots, m_{N_T}(k); \quad t = 0, 1, \dots, N_T$$
(28)

where

$$\hat{\omega}_{jt}(\theta(k)) = \begin{cases} 1 & \text{if } \theta_{jt} \in \theta(k) \\ 0 & \text{otherwise.} \end{cases}$$
(29)

Not all joint association events are feasible joint events. As per our target tracking model assumptions, feasible association events are only those where no more than one measurement is associated with each target. Therefore a feasible association event is the one which satisfies the following two conditions:

1. A measurement can have only one source, i.e.,

$$\sum_{j=0}^{N_T} \hat{\omega}_{jt}(\theta(k)) = 1 \quad \forall j.$$
(30)

# 2. Each target can generate at most one measurement, i.e.,

ŧ

$$\delta_t(\theta(k)) \stackrel{\Delta}{=} \sum_{j=1}^{m_{N_T}} \hat{\omega}_{jt}(\theta(k)) \le 1 \quad t = 1, \dots, N_T.$$
(31)

 $\delta_t(\theta(k))$  in Equation (31), called the target detection indicator, indicates if a measurement is associated with target *t* in event  $\theta(k)$ . Similarly, we define the measurement association indicator  $\tau(\theta(k))$  to indicate if measurement *j* is associated with a target in event  $\theta(k)$ .

$$\tau_j(\theta(k)) \stackrel{\Delta}{=} \sum_{t=1}^{N_T} \hat{\omega}_{jt}(\theta(k)) \quad j = 1, \dots, m_{N_T}(k).$$
(32)

We can obtain the number of false alarms in event  $\theta(k)$  as

$$\phi(\theta(k)) = \sum_{j=1}^{m_{N_T}(k)} [1 - \tau_j(\theta(k))].$$
(33)

## 3.3.4. Joint Data Association Probabilities

The joint data association probabilities are calculated as

$$P\{\theta(k)|Z^k\} = \frac{1}{c} [Z_{m_{N_T}(k)}(k)|\theta(k), m_{N_T}(k), Z^{k-1}]P\{\theta(k)|m_{N_T}(k)\}$$
(34)

where  $P{\cdot}$  indicates the probability mass function (PMF) and *c* is the normalization constant. The marginal association probabilities are obtained from the joint DA probabilities by summing over all the joint events according to Equation (29). The marginal association probabilities at time *k* are then given as

$$\beta_{j}^{t}(k) = \sum_{\theta(k)} P\{\theta(k) | Z^{k}\} \hat{\omega}_{jt}(\theta(k)) \quad j = 1, \dots, m_{N_{T}}(k); \quad t = 0, 1, \dots, N_{T}$$
(35)

$$\beta_0^t(k) = 1 - \sum_{j=1}^{m_{N_T}(k)} \beta_j^t(k) \quad t = 0, 1, \dots, N_T$$
(36)

where the probability  $\beta_j^t$  represents that the measurement *j* is associated with target *t* and  $\beta_0^t$  indicates that none of the validated measurements is associated with target *t*. The expression for the joint association probabilities in Equation (35) can be given in terms of the variables defined in JPDAF Section 3.3.3 as

$$P\{\theta(k)|Z^k\} = \frac{1}{c_1} \prod_{j=1}^{m_{N_T}(k)} \{\lambda^{-1}[\Lambda_j^{t_j}(k)]\}^{\tau_j(\theta(k))} \prod_{t=1}^{N_T} (P_D)^{\delta_t(\theta(k))} (1-P_D)^{1-\delta_t(\theta(k))}$$
(37)

where  $\Lambda_j^{t_j}$  is the Gaussian density of measurement *j* associated with target of index  $t_j$  given by

$$\Lambda_{i}^{t_{j}}(k) = \mathcal{N}[z_{j}(k); \hat{z}_{t_{i}}(k|k-1), S_{t_{j}}(k)]$$
(38)

and  $c_1$  is the new normalization constant. The marginal association probabilities obtained in Equations (35) and (36) are used in conjunction with the state estimation equations given in PDAF Section 3.1.4 for estimating state of each target separately.

## 3.4. Loopy Sum-Product Algorithm

Graphical models can be used for representing the joint probability distributions of many variables efficiently by exploiting factorization. For SPA, factor graphs are utilized for representing the DA relations between multiple targets and their validated measurements. SPA is then conducted on the resulting loopy factor graph for obtaining the marginal DA probabilities. The algorithm for the loopy-SPA can be given as follows.

## 3.4.1. Belief Propagation in Factor Graphs

Kschischang et al. [3] demonstrated how factor graphs can be used to interpret a variety of algorithms such as the KF, the Viterbi algorithm, the Hidden Markov Model, etc. A factor graph is a standard bipartite graphical representation of a mathematical relationship between random variables and local functions. While formulating a factor graph to express the structure of the factorization of a global function of many variables into several local functions, each node represents each random variable  $n \in \mathfrak{N}$ , each factor represents each local function  $f \in \mathfrak{F}$ , and an edge connects a node n to a factor f if and only if n is an argument of f.

The SPA algorithm, also called as Belief Propagation (BP), passes messages, called beliefs, between nodes and factors in an iterative manner for conducting optimal inference on a tree-structured factor graph. We denote by  $\psi_{\cdot}(\cdot)$  the joint probability distribution function, by  $\mu_{n\to f}(x_n)$  the message sent from node  $n \in \eta_f$  to factor f, by  $\mu_{f\to n}(x_n)$  the message sent from factor f to node  $n \in \eta_f$ , by  $\eta_f \subseteq \mathfrak{N}$  the set of neighboring nodes of f, and by  $\eta_n = \{f \in \mathfrak{F} | n \in \eta_f\}$  the factors involving node n. The message computation performed by SPA can be given as

$$\mu_{n \to f}(x_n) = \prod_{\xi \in \eta_n \setminus \{f\}} \mu_{\xi \to n}(x_n) \tag{39}$$

$$\mu_{f \to n}(x_n) = \sum_{\sim \{x_n\}} \left( \psi_f(x_{\eta_f}) \prod_{\xi \in \eta_f \setminus \{n\}} \mu_{\xi \to f}(x_{\xi}) \right)$$
(40)

where  $\sim \{x_n\}$  denotes the summation over all arguments of f except  $x_n$ . For factorization involving continuous variables, the summation is replaced with an integral taken over a Lebesgue measure. The algorithm is known as sum-product because the steps involved are summations (or integrals) and products of factors and messages.

SPA extended to loopy graphs is called LSPA. LSPA simply requires repeated application of SPA until convergence occurs. Practically, this means computing messages continuously using Equations (39) and (40) until the maximum error between subsequent messages is less than a pre-set threshold. However, LSPA is neither guaranteed to converge to the right answer nor to converge at all.

## 3.4.2. Factor Graphs for Data Association

We consider the DA problem involving known and fixed  $N_T$  targets and their combined validated measurements  $Z_{N_T}(k)$  at time k obtained as explained in JPDAF Section 3.3.1. The number of combined validated measurements are  $m_{N_T}(k)$  at time k. Formulating a factor graph for the DA problem that is guaranteed to converge according to [5,14,15], we need the following two sets of association variables:

- 1. Target oriented association variable (*a*): Create an association variable  $a_i(k) \in \{0, 1, ..., m_{N_T}(k)\}$  for each target  $i \in \{1, ..., N_T\}$ . The value assigned to  $a_i(k)$  is an index to the measurement with which target *i* is hypothesized to be associated at time *k* (zero if the target is hypothesized to not have been detected). The complete set of target oriented association variables at time *k* is denoted by a(k).
- 2. Measurement oriented association variable (*b*): Create an association variable  $b_j(k) \in \{0, 1, ..., N_T\}$  for each measurement  $j \in \{1, ..., m_{N_T}(k)\}$ . The value assigned to  $b_j(k)$  is an index to the target with which measurement *j* is hypothesized to be associated at time *k* (zero if the measurement is hypothesized to be clutter). The complete set of measurement oriented association variables at time *k* is denoted by b(k).

Given one set of association variables, the other set can be perfectly reconstructed. As shown in [5,14,15], this use of seemingly redundant information while forming a factor graph leads to the remarkable result of guaranteed convergence of LSPA computed on the said factor graph. A bipartite graphical model formed using the association variables a(k) an b(k) is shown in Figure 1.



**Figure 1.** Bipartite graphical model formulation for data association at time *k*. The value assigned to  $a_i(k)$  is an index to the measurement with which target *i* is hypothesized to be associated at time *k* and the value assigned to  $b_j(k)$  is an index to the target with which measurement *j* is hypothesized to be associated at time *k*.

Our goal is to estimate state vectors  $x_i(k)$ ,  $i \in \{1, ..., N_T\}$  from the measurement vectors Z(k) at time k. Introducing the data-association variables a(k) and b(k), a Bayesian approach to state estimation of  $x_i(k)$  produces a posterior probability density function (PDF)  $f(x_i(k), a(k), b(k)|Z(k))$ . This can be achieved by marginalizing the joint posterior PDF f(X(k), a(k), b(k)|Z(k)), where  $X(k) = (x_i(k), \ldots, x_{N_T}(k))$ . However, direct marginalization of f(X(k), a(k), b(k)|Z(k)) is not always feasible. Assuming that the posterior PDF of X(k) factorizes into a product of terms associated with each target  $x_i(k)$ , an efficient marginalization is given as

$$f(X(k), a(k), b(k)) \propto \prod_{i=1}^{N_T} \psi_i(x_i(k), a_i(k)) \prod_{j=1}^{m_{N_T}(k)} \psi_{i,j}(a_i(k), b_j(k))$$
(41)

where

$$2\psi_i(x_i(k), a_i(k)) = \begin{cases} [1 - P_D(x_i(k))f(x_i(k)|Z(k-1))] & \text{if } a_i(k) = 0\\ [P_D(x_i(k))f(z_{a_i(k)}|x_i(k))f(x_i(k)|Z(k-1))]/\lambda(z_{a_i(k)}) & \text{if } a_i(k) \neq 0 \end{cases}$$
(42)

and

$$\psi_{i,j}(a_i(k), b_j(k)) = \begin{cases} 0 & \text{if } a_i(k) = j, \ b_j(k) \neq i \text{ or } b_j(k) = i, \ a_i(k) \neq j \\ 1 & \text{otherwise.} \end{cases}$$
(43)

Here,  $\propto$  indicates equality up to a normalization factor,  $\psi_i(x_i(k), a_i(k))$  indicates the dependence of the factors  $\psi_i(\cdot)$  on Z(k),  $\psi_{i,j}(a_i(k), b_j(k))$  enforces consistency of the redundant association variables  $a_i(k)$  and  $b_j(k)$  describing the same association configuration,  $P_D(x_i)$  gives the probability of detection for target  $x_i$ , and  $\lambda(z_j)$  gives the PDF value of measurement  $z_j$  occurring as a result of false alarm with a Poisson point process.

The factorization structure in Equation (41) can be represented by a factor graph. As an example, for  $X(k) = (x_1(k), x_2(k))$  and  $Z(k) = (z_1(k), z_2(k))$ , the factor-graph representation of Equation (41) is shown in Figure 2. Since the target states  $x_i(k)$  in Figure 2 are leaf nodes, we can marginalize these and replace the factor  $\psi_i(x_i(k), a_i(k))$  with  $\psi_i(a_i(k))$ . In a factor graph, each parameter variable is represented by a variable node (circle node), and each factor is represented by a rectangle node, as shown in Figure 2. Each variable node and each factor node are connected by an edge if the variable is an argument of the factor. For each node, certain messages are calculated according to Equations (39) and (40). Message passing is started at variable nodes with only one edge (which pass a constant message) and/or factor nodes with only one edge (which pass the corresponding factor). Finally, for each variable node, a belief (posterior PDF value) is calculated as the product of all incoming messages (passed from all adjacent factor nodes) followed by a normalization. For a tree-structured factor graph, these beliefs are exactly equal to

marginal posterior PDF values. For a loopy factor graph, the beliefs are in general only approximations of the respective marginal posterior PDF values. A detailed description of the messages passed between each variable node and factor node is given in [19]. A sample MATLAB program for implementing one iteration of loopy SPA is given in [14].



**Figure 2.** Factor graph representing the factorization of the joint posterior probability density function (PDF)  $f(x_1, x_2, a_1, a_2, b_1, b_2 | z_1, z_2)$  according to Equation (41), depicted for one time step. For simplicity, the time index *k* is omitted.

# 3.4.3. Joint Data Association Probabilities

Once the combined validated measurements are obtained as given in Equation (24), the joint data association probabilities under the stated assumptions at time k are calculated as

$$p(a(k), b(k)|Z^k) = \prod_{i=1}^{N_T} \psi_i(a_i(k)) \prod_{j=1}^{m_{N_T}(k)} \psi_{i,j}(a_i(k), b_j(k))$$
(44)

where

$$\psi_{i,j}(a_i(k), b_j(k)) = \begin{cases} 0 & \text{if } a_i(k) = j, \ b_j(k) \neq i \text{ or } b_j(k) = i, \ a_i(k) \neq j \\ 1 & \text{otherwise} \end{cases}$$
(45)

enforce consistency of the redundant association variables  $a_i(k)$  and  $b_j(k)$  describing the same association configuration, and  $\psi_i(a_i(k) = 0) = 1$  and  $\psi_i(a_i(k) = j > 0)$  are the (unnormalized) single target data association probabilities obtained as explained in PDAF Section 3.1.3. Equation (44) can be expressed as the formulation of SPA for a bipartite model illustrated in Figure 1 in which all target association variables  $a_i(k)$  are connected to all measurement association variables  $b_j(k)$ . In this case, LSPA may be implemented via two half iterations, alternating between the two sets of messages  $\mu_{a_i(k) \rightarrow b_j(k)}(b_j(j))$  and  $\mu_{b_j(j) \rightarrow a_i(k)}a_i(k)$ . The message updating equations according to Equations (39) and (40) can be given as

$$\mu_{a_i(k) \to b_j(k)}(b_j(k)) = \sum_{a_i(k)} \psi_i(a_i(k))\psi_{i,j}(a_i(k), b_j(k)) \prod_{j' \neq j} \mu_{b_{j'(k)} \to a_i(k)}(a_i(k))$$
(46)

$$= \begin{cases} \psi_{i}(j) \prod_{j' \neq j} \mu_{b_{j'(k)} \to a_{i}(k)}(j) & \text{if } b_{j}(k) = i \\ \sum_{a_{i}(k) \neq j} \psi_{i}(a_{i}(k)) \prod_{j' \neq j} \mu_{b_{j'}(k) \to a_{i}(k)}(a_{i}(k)) & \text{if } b_{j}(k) \neq i \end{cases}$$
(47)

and

$$\mu_{b_j(k) \to a_i(k)}(a_i(k)) = \sum_{b_j(k)} \psi_{i,j}(a_i(k), b_j(k)) \prod_{i' \neq i} \mu_{a_{i'}(k) \to b_j(k)}(b_j(k))$$
(48)

$$=\begin{cases} \prod_{i'\neq i} \mu_{a_{i'}\rightarrow j}(i) & \text{if } a_i(k) = j\\ \sum_{b_j(k)\neq i} \prod_{i'\neq i} \mu_{a_{i'}\rightarrow j}(b_j(k)) & \text{if } a_i(k)\neq j. \end{cases}$$
(49)

These messages can be further simplified as shown in [14,15]. Upon convergence of LSPA, the approximate marginal association probabilities can be given as

$$p(a_i(k) = j | Z^k) = \frac{\psi_i(j)\mu_{b_j(k) \to a_i(k)}(a_i(k))}{\sum_{j'=0}^{m_{N_T}(k)} \psi_i(j')\mu_{b_{j'}(k) \to a_i(k)}(a_i(k))}$$
(50)

$$p(b_j(k) = i|Z^k) = \frac{\mu_{a_i(k) \to b_j(k)}(b_j(k))}{\sum_{i'=0}^{N_T} \mu_{a_{i'}(k) \to b_i(k)}(b_j(k))}$$
(51)

where  $\mu_{b_j(k)=0 \to a_i(k)}(a_i(k)) \stackrel{\Delta}{=} 1$  and  $\mu_{a_i(k)=0 \to b_j(k)}(b_j(k)) \stackrel{\Delta}{=} 1$ . The marginal association probabilities are used to estimate target states as per PDAF Section 3.1.4.

## 3.5. Distance-Weighting Loopy Sum-Product Algorithm

Since the factor-graph representation for the MTT DA problem contains loops, and the convergence process of calculating the marginal data-association probabilities using LSPA is governed by some heuristically determined preset threshold, different initiation messages can lead to different final beliefs. These initiation messages for the convergence process happen to be the (unnormalized) single-target data-association probabilities. These probabilities directly influence the marginal data-association probabilities at the end of the convergence process and, consequently, the estimation of the target state. We would expect that a more accurate initial set of single-target probabilities leads to either more accurate final beliefs, a faster convergence to the final beliefs, or both. The distance-weightbased association probabilities have been proven to be more accurate for tracking a single target in densely cluttered environments [4]. We would expect this adjustment to the initial condition to change the calculation of the association probabilities and hence the overall tracking process in terms of tracking accuracy or computation time, and therefore worth exploring.

Here we formulate a modification for the joint data association probabilities calculated using LSPA under the stated assumptions as described in Section 3.4. The distance-weight based joint data association probabilities at time k can be given as

$$p(a(k), b(k)|Z^k) = \prod_{i=1}^{N_T} \psi_i(a_i(k)) \prod_{j=1}^{m_{N_T}(k)} \psi_{i,j}(a_i(k), b_j(k))$$
(52)

where

$$\psi_{i,j}(a_i(k), b_j(k)) = \begin{cases} 0 & a_i(k) = j, \ b_j(k) \neq i \text{ or } b_j(k) = i, \ a_i(k) \neq j \\ 1 & \text{otherwise} \end{cases}$$
(53)

enforce consistency of the redundant association variables  $a_i(k)$  and  $b_j(k)$  describing the same association configuration, and  $\psi_i(a_i(k) = 0) = 1$  and  $\psi_i(a_i(k) = j > 0) = \mathcal{B}_i^t(k)$  are the (unnormalized) distance-weight based single target data association probabilities obtained in Equation (22). The approximated marginal association probabilities are obtained from the joint DA probabilities as described in Section 3.4.

## 4. Simulation and Analysis

#### 4.1. The Dynamic Model

We assume a two-dimensional system where the target state vector consists of position and velocity in each of the two coordinates. For target *i* at time *k*, the target state  $x_i(k) = [x(k), \dot{x}(k), y(k), \dot{y}(k)]$  has four components: The first and second components are the horizontal location and velocity, respectively, while the third and fourth components are the vertical location and velocity, respectively. The system is equipped with the nearly constant velocity (NCV) model (also sometimes called the constant velocity (CV) model). The system is described by Equations (1) and (2) with

$$F(k) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(54)

$$G(k) = \begin{bmatrix} T & 0\\ 0 & T^2/2\\ 0 & T \end{bmatrix}$$
(55)

$$H(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$
(56)

$$Q(k) = \begin{bmatrix} q & 0\\ 0 & q \end{bmatrix}$$
(57)

$$R(k) = \begin{bmatrix} r & 0\\ 0 & r \end{bmatrix}$$
(58)

where T is the sampling interval.

## 4.2. Simulation Parameters

For our simulations, we set the probability of detection  $P_D = 0.9$ , the gate threshold  $\gamma = 9.21$  corresponding to gate probability  $P_G = 0.99$ , sampling interval T = 1 s, the process noise variance q = 0.05, and the measurement noise variance r = 5. To better evaluate the accuracy of different algorithms under multiple conditions, we varied the clutter density generated according a Poisson point process from  $\lambda = 1.0 \times 10^{-4}/\text{scan/m}^2$  to  $\lambda = 5.0 \times 10^{-4}/\text{scan/m}^2$ . We also varied the number of tracked targets from 1 to 6. The initial state of the first target is always set at  $x_1(1) = [100 \text{ m}, 30 \text{ m/s}, 100 \text{ m}, 30 \text{ m/s}]$ . We set the initial state of each consecutive target by  $x_i(1) = [100 \text{ m}, 30 \text{ m/s}, (100 - i \times 100 \times c(i)) \text{ m}, (30 - i \times 30 \times c(i)) \text{ m/s}]$  where *i* is the target index and c(i) is a single uniformly distributed random number in the interval (0, 1) independent of each other. To compare the performance, we performed 500 Monte Carlo simulations on MATLAB (Natick, MA, USA) [21].

## 4.3. Results and Discussion

When choosing an optimal tracking algorithm, there is typically a trade off between tracking accuracy and computation time. Maintaining a high-level accuracy in complex scenarios where multiple targets need to be tracked simultaneously and the environment is particularly noisy requires significant computation time. Traditionally, the tracking accuracy of an estimator is calculated in terms of a miss-distance, or localization error, between a reference value and its estimated value. In our context, we are also interested in evaluating missed detections and false alarms. The generalized optimal sub-pattern assignment (GOSPA) metric has been designed to reflect this performance [22]. Informally, the GOSPA metric can be defined as

$$GOSPA = \sum localization error + \frac{cutoff distance}{2} (\# of missed detections + \# of false alarms)$$

(for a precise detailed description of GOSPA, see [22]). The localization error is for pairs of true targets and target estimates that are sufficiently close. A missed detection is declared if there is no corresponding target sufficiently close to it, and a false alarm is declared if there is no corresponding true target sufficiently close to it. To evaluate the performance of the different tracking algorithms described in Section 3 in terms of the miss-distance, the GOSPA metric, and computation time, we considered multiple scenarios with varying levels of complexities.

Figures 3a,b depict two such scenarios where the trajectories of three targets are shown for clutter densities  $\lambda = 1 \times 10^{-4} / \text{m}^2$  and  $\lambda = 5 \times 10^{-4} / \text{m}^2$ , respectively. The scenario depicted in Figure 3b is more demanding because of the increased number of false alarms. Figure 4a,b compares the performance of LSPA and DWPDA in terms of tracking accuracy, calculated using the root mean square (RMS) position error, over a period of 100-time intervals for the above two scenarios, respectively. We see that the RMS position errors for LSPA stay close to 0 over the entire period while they are always increasing for DWPDA as time progresses. We see similar results while tracking six crossing targets with clutter densities  $\lambda = 1 \times 10^{-4} / \text{m}^2$  and  $\lambda = 5 \times 10^{-4} / \text{m}^2$ , as depicted in Figure 5a,b. Figure 6a,b show that the computation times for the two clutter scenarios as the number of targets increases from one to six. We see that the computation times required for LSPA are only slightly higher than those of DWPDA. Finally, Tables 1 and 2 summarize the results for the performance in terms of the GOSPA metric based on the Euclidean distance with a cutoff parameter of 30. Table 1 shows the average GOSPA errors as we increase the number of targets from 1 to 6 while keeping the clutter density constant at  $\lambda = 3 \times 10^{-4} / \text{m}^2$ . We can see that irrespective of the number of targets that are being tracked, GOSPA errors for LSPA are only a fraction of the errors for DWPDA. We see a similar pattern in Table 2 where we increase the clutter density from  $\lambda = 1 \times 10^{-4}/m^2$  to  $\lambda = 5 \times 10^{-4}/m^2$  while keeping the number of targets fixed. The poor performance of DWPDA is expected, since the algorithm is ill-equipped to deal with the problem of DA in MTT, and GOSPA appropriately penalizes any missed detections and false alarms. From the above results, it is evident that LSPA is superior to DWPDA in terms of tracking accuracy in all scenarios without trading off much computation time. This superior tracking accuracy can be attributed to the reduction in the association probabilities of false measurements in the overlapping validation regions from multiple targets and, at the same time, the increase in the association probabilities for actual target measurements. From the above results, we know that LSPA can improve the tracking performance for MTT in densely cluttered environments.



**Figure 3.** True target positions for three crossing targets with different clutter densities. (a) Clutter density  $\lambda = 1 \times 10^{-4} / \text{m}^2$ ; and (b) clutter density  $\lambda = 5 \times 10^{-4} / \text{m}^2$ .

**Table 1.** Average generalized optimal sub-pattern assignment (GOSPA) errors for tracking multiple crossing targets with clutter density  $\lambda = 3 \times 10^{-4} / m^2$ .

Number of Targets					
1	2	3	4	5	6
0.4571 0.4435 0.6751	0.9103 0.8787 2.5626	1.5033 1.5243 4.6425	1.9659 1.8601 6.1505	2.3441 2.2664 9.3851	2.9000 2.8253 11.5240
	<b>1</b> 0.4571 0.4435 0.6751 4.7454	1         2           0.4571         0.9103           0.4435         0.8787           0.6751         2.5626           4.7454         9.1838	Number           1         2         3           0.4571         0.9103         1.5033           0.4435         0.8787         1.5243           0.6751         2.5626         4.6425           4.7454         9.1838         14.2879	1         2         3         4           0.4571         0.9103         1.5033         1.9659           0.4435         0.8787         1.5243         1.8601           0.6751         2.5626         4.6425         6.1505           4.7454         9.1838         14.2879         18.5087	1         2         3         4         5           0.4571         0.9103         1.5033         1.9659         2.3441           0.4435         0.8787         1.5243         1.8601         2.2664           0.6751         2.5626         4.6425         6.1505         9.3851           4.7454         9.1838         14.2879         18.5087         23.3401

			λ		
	$1  imes 10^{-4}$	$2 imes 10^{-4}$	$3 imes 10^{-4}$	$4 imes 10^{-4}$	$5 imes 10^{-4}$
LSPA	1.4280	1.6058	1.4056	1.6628	1.7690
DWLSPA	1.3718	1.5652	1.3388	1.5289	1.6257
JPDA	3.5367	4.2030	4.3475	4.4828	5.2677
DWPDA	7.4128	11.8464	14.4591	15.0590	15.8318

Table 2. Average GOSPA errors for tracking three targets with different clutter densities.



**Figure 4.** RMS position error for three crossing targets using DWPDA and LSPA with different clutter densities. (a) Clutter density  $\lambda = 1 \times 10^{-4} / \text{m}^2$ ; and (b) clutter density  $\lambda = 5 \times 10^{-4} / \text{m}^2$ .



**Figure 5.** RMS position error for six crossing targets using DWPDA and LSPA with different clutter densities. (**a**) Clutter density  $\lambda = 1 \times 10^{-4}/\text{m}^2$ ; and (**b**) clutter density  $\lambda = 5 \times 10^{-4}/\text{m}^2$ .



**Figure 6.** Average computation time for obtaining association probabilities using DWPDA and LSPA for tracking multiple crossing targets with different clutter densities. (a) Clutter density  $\lambda = 1 \times 10^{-4}/\text{m}^2$ ; and (b) clutter density  $\lambda = 5 \times 10^{-4}/\text{m}^2$ . Error bars indicate 95% confidence intervals.

Next, we compare the performance of LSPA with JPDA, in terms of computation time and RMS position error, as the number of targets increases from one to six. Figures 7 and 8 show the results of this comparison for the two clutter scenarios. We can clearly see that the average computation time of LSPA is much smaller than that of JPDA, so much so that the LSPA computation times are not even visible in Figures 7a and 8a. However, in contrast to the comparison of LSPA and DWPDA in terms of the RMS position error, in the case of LSPA compared with JPDA, Figures 7b and 8b clearly show that there is little difference in RMS position error. The same observation applies in terms of computation time and RMS position error in Figure 9a,b respectively when we compare LSPA with JPDA in scenarios involving a fixed number of targets as the clutter density increases from  $\lambda = 1 \times 10^{-4} / m^2$ to  $\lambda = 5 \times 10^{-4}$ /m<sup>2</sup>. While there is little difference between LSPA and JPDA in terms of RMS position errors, Tables 1 and 2 show that GOSPA errors for LSPA are less than 1/3rd of GOSPA errors for JPDA across almost all scenarios. These relatively higher GOSPA errors for JPDA can be explained by a few missed detections when multiple target paths overlap. These missed detections are rightly penalized in the GOSPA metric. These results show that LSPA dominates JPDA in terms of computation time while maintaining a high level of tracking accuracy. This dramatic reduction in computation time for LSPA can be explained by the implementation of the loopy factor graph, resulting in simultaneous updating of the joint association probabilities for multiple targets during each iteration of LSPA. The results show that LSPA scales well for real-time applications involving complex tracking scenarios.



**Figure 7.** Tracking multiple crossing targets using LSPA and JPDA with clutter density  $\lambda = 1 \times 10^{-4} / \text{m}^2$ . (a) Average computation time for obtaining association probabilities; (b) average RMS position error. Error bars indicate 95% confidence intervals.



**Figure 8.** Tracking multiple crossing targets using loopy sum-product algorithm (LSPA) and joint probabilistic data association (JPDA) with clutter density  $\lambda = 5 \times 10^{-4}/\text{m}^2$ . (a) Average computation time for obtaining association probabilities; (b) average root mean square (RMS) position error. Error bars indicate 95% confidence intervals.



**Figure 9.** Tracking three crossing targets using LSPA and JPDA. (**a**) Average computation time for obtaining association probabilities; (**b**) average RMS position error. Error bars indicate 95% confidence intervals.

Finally, to see whether the integration of LSPA with the distance-weighting scheme has any effect on its performance, we compare LSPA with DWLSPA in terms of the same two metrics as above. Figures 10 and 11 show the results of this comparison for two clutter scenarios respectively, as the number of targets increases from one to six. There is little difference between LSPA and DWLSPA in terms of computation times, as shown in Figures 10a and 11a. Similarly, we can see that the RMS position errors for LSPA and DWLSPA are almost identical in Figures 10b and 11b. Figure 12a,b show that the difference between LSPA and DWLSPA remains negligible, in terms of both computation time and RMS position error, for scenarios with a fixed number of targets and varying clutter densities. Tables 1 and 2 show that GOSPA errors for LSPA and DWLSPA are comparable across all tested scenarios. This means that in addition to the localization errors, the missed detections and false alarms remain consistent between LSPA and DWLSPA, and the potential advantage of DWLSPA with the additional distance-weighting information is not apparent. Surprisingly, LSPA and DWLSPA perform equally well in every scenario in terms of both tracking accuracy and computation time. The unchanged performance of DWLSPA can be explained by the initiation of LSPA with small improvements in single-target association probabilities having insignificant effect on joint association probabilities calculated at the end of a large number of iterations. However, this unexpected lack of improvement contrasts sharply with results reported in [4] showing significant improvement when introducing distance weighting relative to PDA. This result is interesting and useful because we can see that distance weighting does not always lead to better performance. Moreover, the following important observation remains: LSPA is reliable and efficient for tracking multiple objects in cluttered environments.



**Figure 10.** Tracking multiple crossing targets using LSPA and DWLSPA with clutter density  $\lambda = 1 \times 10^{-4} / \text{m}^2$ . (a) Average computation time for obtaining association probabilities; (b) average RMS position error. Error bars indicate 95% confidence intervals.



**Figure 11.** Tracking multiple crossing targets using LSPA and DWLSPA with clutter density  $\lambda = 5 \times 10^{-4}$ /m<sup>2</sup>. (a) Average computation time for obtaining association probabilities; (b) average RMS position error. Error bars indicate 95% confidence intervals.



**Figure 12.** Tracking three crossing targets using LSPA and DWLSPA. (**a**) Average computation time for obtaining association probabilities; (**b**) average RMS position error. Error bars indicate 95% confidence intervals.

# 5. Conclusions

In this paper, we formulated a distance-weighting PDA approach for LSPA and examined its effect for tracking multiple objects in cluttered environments. It has been previously shown that a modification of PDA according to a weighting scheme based on distances between predicted and true target positions improves the tracking accuracy of PDA. LSPA is known to be better than PDA and JPDA and, since PDA constitutes a crucial building block of LSPA, we expected the integration of DWPDA with LSPA to boost the overall performance even further. We studied the performance of LSPA against DWPDA, JPDA, and DWLSPA for a wide range of tracking scenarios involving multiple targets and varying clutter densities. Our results confirm that LSPA is superior to DWPDA in terms of tracking accuracy and dominates JPDA in terms of computation time. However, contrary to expectations, we found that the distance-weighting approach, when integrated with LSPA, does not enhance the performance of LSPA in terms of either tracking accuracy or computation time. The simulation scenarios in the experiment could be made more realistic with the addition of appearing and disappearing targets and time-varying target velocities. These scenarios add extra layers of complexity to the DA without affecting the conclusions we draw in this paper. Regardless, we demonstrated the validity of LSPA having computational requirements suitable for real-time processing and accuracy of tracking multiple targets in cluttered environments.

Author Contributions: conceptualization, P.U.D., E.K.P.C., and T.J.M.; methodology, P.U.D.; software, P.U.D.; validation, P.U.D., E.K.P.C. and T.J.M.; formal analysis, P.U.D. and E.K.P.C.; investigation, P.U.D.; resources, E.K.P.C.; data curation, P.U.D.; writing—original draft preparation, P.U.D.; writing—review and editing, P.U.D. and E.K.P.C.; visualization, P.U.D. and E.K.P.C.; supervision, E.K.P.C.; project administration, E.K.P.C.; funding acquisition, E.K.P.C. All authors have read and agreed to the final revision of the manuscript.

**Funding:** This work was partially supported by Sandia National Laboratories under Purchase Order 1980525. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

## Abbreviations

The following abbreviations are used in this manuscript:

DA	Data Association
MTT	Multiple Target Tracking
KF	Linear Kalman Filter
NN	Nearest Neighbor
PDA	Probabilistic Data Association
DWPDA	Distance-Weighting Probabilistic Data Association
PDAF	Probabilistic Data Association Filter
JPDA	Joint Probabilistic Data Association
JPDAF	Joint Probabilistic Data Association Filter
MHT	Multiple Hypothesis Tracking
SPA	Sum-Product Algorithm
LSPA	Loopy Sum-Product Algorithm
DWLSPA	Distance-Weighting Loopy Sum-Product Algorithm
LR	Likelihood Ratio
PDF	Probability Distribution Function
PMF	Probability Mass Function
NCV	Nearly Constant Velocity
RMS	Root Mean Square
GOSPA	Generalized Optimal Sub-Pattern Assignment

### References

- 1. Bar-Shalom, Y.; Tse, E. Tracking in a Cluttered Environment With Probabilistic Data Association. *Automatica* **1975**, *11*, 451–460. [CrossRef]
- 2. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Technical Report; University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 1995.
- 3. Kschischang, F.R.; Brendan, J.F.; Loeliger, H.-A. Factor Graphs and the Sum-Product Algorithm. *IEEE Trans. Inf. Theory* **2001**, 47, 498–519. [CrossRef]
- 4. Chen, X.; Li, Y.; Li, Y.; Yu, J.; Li, X. A Novel Probabilistic Data Association for Target Tracking in a Cluttered Environment. *Sensors* **2016**, *16*, 2180. [CrossRef] [PubMed]
- Williams, J.L.; Lau, R.A. Data Association by Loopy Belief Propagation. In Proceedings of the 13th International Conference on Information Fusion, Edinburgh, SA, Australia, 26–29 July 2010; pp. 1–8.
- 6. Li, X.; Bar-Shalom, Y. Tracking in Clutter With Nearest Neighbor Filters: Analysis and Performance. *IEEE Trans. Aerosp. Electron. Syst.* **1996**, *32*, 995–1010.

- Fortmann, T.E.; Bar-Shalom, Y.; Scheffe, M. Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association. *IEEE J. Ocean. Eng.* 1983, *8*, 173–184. [CrossRef]
- Rezatofighi, S.H.; Milan, A.; Zhang, Z.; Shi, Q.; Dick, A.; Reid, I. Joint Probabilistic Data Association Revisited. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December, 2015; pp. 3047–3055.
- 9. Blackman, S. Multiple Hypothesis Tracking for Multiple Target Tracking. *IEEE Trans. Aerosp. Electron. Syst.* 2004, 19, 5–18. [CrossRef]
- 10. Roecker, J.A.; Phillis, G.L. Suboptimal Joint Probabilistic Data Association. *IEEE Trans. Aerosp. Electron. Syst.* **1993**, *29*, 510–517. [CrossRef]
- 11. Reid, D.B. An Algorithm for Tracking Multiple Targets. IEEE Trans. Automat. Cont. 1979, 24, 843-854. [CrossRef]
- 12. Bar-Shalom, Y.; Willett, P.; Tian, X. Tracking and Data Fusion: A Handbook of Algorithms; YBS Publishing: Storrs, CT, USA, 2011.
- 13. Wiberg, N; Loeliger, H.-A.; Kotter, R. Codes and Iterative Decoding on General Graphs. Eur. Trans. Telecomm. 1996, 6, 513–525.
- Williams, J.L.; Lau, R.A. Convergence of Loopy Belief Propagation for Data Association. In Proceedings of the Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Brisbane, QLD, Australia, 7–10 December 2010; pp. 175–180.
- 15. Williams, J.L.; Lau, R.A. Approximate Evaluation of Marginal Association Probabilities with Belief Propagation. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *4*, 2942–2959. [CrossRef]
- Meyer, F.; Braca, P.; Willett, P.; Hlawatsch, F. Scalable multitarget tracking using multiple sensors: A belief propagation approach. In Proceedings of the 18th International Conference on Information Fusion (Fusion), Washington, DC, USA, 6–9 July 2015; pp. 1778–1785.
- Meyer, F.; Braca, P.; Willett, P.; Hlawatsch, F. Tracking an unknown number of targets using multiple sensors: A belief propagation method. In Proceedings of the 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 719–726.
- 18. Meyer, F.; Braca, P.; Willett, P.; Hlawatsch, F. A scalable algorithm for tracking an unknown number of targets using multiple sensors. *IEEE Trans. Signal Process.* **2017**, *65*, 3478–3493. [CrossRef]
- 19. Meyer, F.; Kropfreiter, T.; Williams, J.L.; Lau, R.; Hlawatsch, F.; Braca, P.; Win, M.Z. Message passing algorithms for scalable multitarget tracking. *Proc. IEEE* **2018**, *106*, 221–259. [CrossRef]
- 20. Gaglione, D.; Soldi, G.; Braca, P.; De Magistris, G.; Meyer, F.; Hlawatsch, F. Classification-aided multitarget tracking using the sum-product algorithm. *IEEE Signal Process. Lett.* **2020**, *27*, 1710–1714. [CrossRef]
- 21. MATLAB. Version 9.9.0.1467703 (R2020b); MATLAB: Natick, MA, USA, 2020.
- Rahmathullah, A.S.; García-Fernández, Á.F.; Svensson, L. Generalized optimal sub-pattern assignment metric. In Proceedings of the 20th International Conference on Information Fusion (Fusion), Xi'an, China, 10–13 July 2017; pp. 1–8.