Robust Multipath-Assisted SLAM with Unknown Process Noise and Clutter Intensity

Zesheng Dan, Baowang Lian * and Chengkai Tang

School of Electronics and Information, Northwestern Polytechnical University, Xi’an 710072, China; danzesheng@mail.nwpu.edu.cn (Z.D.); cktang@nwpu.edu.cn (C.T.)
* Correspondence: bwlian@nwpu.edu.cn

Abstract: In multipath-assisted simultaneous localization and mapping (SLAM), the geometric association of specular multipath components based on radio signals with environmental features is used to simultaneously localize user equipment and map the environment. We must contend with two notable model parameter uncertainties in multipath-assisted SLAM: process noise and clutter intensity. Knowledge of these two parameters is critically important to multipath-assisted SLAM, the uncertainty of which will seriously affect the SLAM accuracy. Conventional multipath-assisted SLAM algorithms generally regard these model parameters as fixed and known, which cannot meet the challenges presented in complicated environments. We address this challenge by improving the belief propagation (BP)-based SLAM algorithm and proposing a robust multipath-assisted SLAM algorithm that can accommodate model mismatch in process noise and clutter intensity. Specifically, we describe the evolution of the process noise variance and clutter intensity via Markov chain models and integrate them into the factor graph representing the Bayesian model of the multipath-assisted SLAM. Then, the BP message passing algorithm is leveraged to calculate the marginal posterior distributions of the user equipment, environmental features and unknown model parameters to achieve the goals of simultaneous localization and mapping, as well as adaptively learning the process noise variance and clutter intensity. Finally, the simulation results demonstrate that the proposed approach is robust against the uncertainty of the process noise and clutter intensity and shows excellent performances in challenging indoor environments.

Keywords: multipath-assisted; robust SLAM; factor graph; message passing

1. Introduction

Simultaneous localization and mapping (SLAM) [1,2] aims to estimate the time-varying position of the mobile agent when it is moving in an unknown environment and build a map of the surroundings from measurements provided by sensors. SLAM is significant in many fields, such as robotics [3], autonomous driving [4], and situation awareness [5]. In multipath-assisted methods [6–10], the relation between multipath components (MPCs) of radio signals and local geometry is used to turn the multipath propagation from an error source to an advantage that can improve the positioning accuracy. Recently, multipath-assisted SLAM, a combination of the two, has attracted considerable attention from scholars and has been applied in the field of indoor positioning [11–16].

Instead of mapping the environment in conventional SLAMs, the map in multipath-assisted SLAM is represented by an unknown number of features with unknown spatial positions. Features denote the physical transmitters (PTs) and virtual transmitters (VTs) in the environment. Radio signals are transmitted from base stations (PTs) to the user equipment (UE). Specular MPCs in the multipath radio channel are described by VTs, which are mirror images of the PTs. Due to the introduction of VTs, specular MPCs can...
be interpreted as line-of-sight (LOS) signals from the VTs, which can increase the number of transmitters involved in positioning and even realize localization in the case of insufficient PTs. Multipath-assisted SLAM detects the VTs associated with the PTs and estimates their positions as well as the trajectory of the UE. Multipath-assisted SLAM can cope with harsh multipath channel conditions and realize positioning and mapping without knowing the prior position of the PTs or without having a sufficient number of PTs, which greatly improves the accuracy and robustness of indoor positioning [17–21].

The data association (DA) between measurements and PTs or VTs is unknown, and measurements will be affected by noise, false alarms and missed detections, which makes the multipath-assisted SLAM complex. A multipath-assisted SLAM algorithm called Channel-SLAM was proposed in [22]. Channel-SLAM is based on a recursive Bayesian filter and can realize SLAM only with the prior position of the PT and the initial position and orientation of the UE. However, Channel-SLAM does not consider the DA uncertainty. Channel-SLAM is extended in [23] by adding a multiple hypothesis tracking (MHT)-based DA algorithm, which enables Channel-SLAM to associate the reappearing MPCs with corresponding VTs. Simulation results demonstrate that the improved Channel-SLAM algorithm has higher positioning reliability and lower computational complexity than the original method. A multipath-assisted SLAM algorithm called BP-SLAM based on belief propagation (BP) [24] was proposed in [25]. The approach jointly realizes the probabilistic DA and the estimation of UE positions and feature positions in the environment map. Simulation results demonstrate that BP-SLAM has lower complexity and higher scalability than other state-of-the-art SLAM algorithms and has higher positioning and mapping accuracy in the case of measurements containing false alarms and missed detections.

Furthermore, unknown and time-varying model parameters also complicate multipath-assisted SLAM. For example, the detection probability describes the possibility of PTs or VTs being detected by the UE. In practice, due to the influence of electromagnetic interference and surrounding environments, the detection probability is usually variable. However, the influence of the detection probability was not considered or simply considered to be fixed and known in [22–25]. Adaptive algorithms of the time-varying detection probability-based BP-SLAM algorithm were proposed in [26–28]. In [26], the detection probability was considered to be a discrete random variable with the value taking from a finite set. Then, the evolution of the detection probability was integrated into the multipath-assisted SLAM Bayesian model as an independent variable node. Thus, the algorithm can realize the online adaptive adjustment of the detection probability. In [27,28], the detection probability was expressed as a function of the MPC amplitude, and improved BP-SLAM algorithms were proposed, which leveraged MPC amplitude information to adaptively adjust the detection probability.

In multipath-assisted SLAM, we have to address the uncertainty of two important model parameters, process noise and clutter intensity, in addition to the detection probability. The change in process noise is mainly reflected in the change in UE dynamic models. Maneuvering or sudden acceleration and turning may cause process model uncertainties in UEs. Clutter represents spurious measurements that do not come from any transmitter (PTs or VTs). Clutter intensity is used to describe the amount of clutter in the environment. To the best of the authors’ knowledge, none of the multipath-assisted SLAM algorithms consider the uncertainty of process noise and clutter intensity at present. However, in the field of multitarget tracking (MTT) [29], which is closely related to feature-based SLAM, some algorithms are proposed with the condition that these two model parameters are unknown. In [30,31], clutter was regarded as measurements generated by incorrect targets. By distinguishing between correct and incorrect targets, the probability hypothesis density (PHD) filter, the cardinalized PHD (CPHD) filter, and the multi-Bernoulli filter were developed to independently estimate the clutter intensity in the environment. The conventional PHD filter and CPHD filter were extended in [32] to include estimation of the time-varying target birth intensity. In [33], an adaptive MTT
algorithm based on the factor graph and the sum product algorithm [34] was proposed. The algorithm can be applied to complex scenes where the target dynamic model and detection probability change at the same time.

Most of the existing multipath-assisted SLAM algorithms regard the process noise and clutter intensity as known and fixed. However, in practical applications, the dynamic model of UEs cannot remain invariable all the time, and the process noise of UEs may change at any time. Clutter environments are often complicated due to the influence of electromagnetic interference, signal scattering and diffraction; hence, it is difficult to match the assumption of known process noise and clutter intensity with reality. Significant mismatches in the process and clutter model parameters seriously affect the SLAM accuracy, and even the estimation results diverge. Therefore, it is of critical importance to study the robust multipath-assisted SLAM algorithm when the process and clutter model parameters are unknown. Based on the framework of the BP-SLAM algorithm, in this paper, we propose a robust multipath-assisted SLAM algorithm, which can address the uncertainties of process and clutter model parameters in multipath-assisted SLAM. First, the UE state is extended to an augmented state including the process noise, and the evolution of the augmented state is described by new prediction and update rules compared to the conventional BP-SLAM algorithm. Second, we add a new variable to represent the clutter intensity in the Bayesian model of BP-SLAM and describe the evolution of the clutter intensity via a Markov chain model. Finally, we integrate the augmented state of the UE and the new variable node into a factor graph representing the Bayesian model of BP-SLAM and use the BP message passing algorithm to calculate marginal posterior distributions of the process noise, the clutter intensity, the UE and other features to realize the online adjustment of the process noise and the clutter intensity in multipath-assisted SLAM. The simulation results demonstrate that the proposed algorithm can adaptively learn the process noise and clutter intensity for SLAM, improving the robustness of the multipath-assisted SLAM system.

The remainder of this paper is organized as follows: Section 2 provides the environment map and the signal model. Section 3 describes the system model with unknown process noise and clutter intensity. Moreover, our robust multipath-assisted SLAM algorithm is proposed in Section 4. Then, the simulation results are reported in Section 5. Finally, Section 6 contains the conclusion of the paper.

2. Environment Map and Signal Model

Multipath-assisted SLAM combines the geometric parameters (delays, angles of arrival (AOAs) and angles of departure (AODs)) of the LOS signal in received radio signals with the geometric parameters of specular MPCs to localize the UE and build the environment map. These parameters are obtained according to the relationship between the positions of the UE and the PTs or VTs. The VT positions are mirror images of the PT positions with respect to the reflection surface. From the UE side, the reflected and virtual propagation paths originating from the VTs have the same AOA and delay; hence, the reflected path can be described as a LOS path between the VT and the UE. Regardless of how the UE moves, the VTs will remain static as long as the PTs are static. As an example, Figure 1 depicts an indoor environment map containing two PTs and some of the corresponding VTs. The red solid circle and the blue solid box denote PT1 and PT2 at \( a^{(1)} \) and \( a^{(2)} \), respectively. The gray lines represent the doors, windows and walls in the room. They all have smooth surfaces that can reflect the radio signals from the PTs. The red hollow circles and blue hollow boxes represent the corresponding VTs, which are mirror images of PT1 and PT2. The green dot is the estimated position of the UE at \( u^{(1)} \), while the magenta line denotes the real trajectory of the UE.
Figure 1. Example of an environment map.

We consider an environment map with one UE and J PTs. The UE position is unknown and time-varying, which is denoted by $u_n \in \mathbb{R}^2$, while the PT positions $a_{i,j}^{(l)} \in \mathbb{R}^2, j \in \{1,...,J\}$ may be unknown, where the number of PT $j$ is supposed to be known. Assume that there are $L_{n}^{(l)} - 1$ VTs associated with the $j$th PT and that their positions are unknown and represented by $a_{i,j}^{(l)} \in \mathbb{R}^2, j \in \{2,...,L_{n}^{(l)}\}$. PTs and VTs are features of the environmental map, and their number is unknown and varies with the UE position. At each discrete time $n$, the radio signal $s(t)$ is transmitted from the $j$th PT to the mobile UE, and the received signal of the UE can be expressed as [17]

$$s_{RX,n}(t) = \sum_{l=1}^{L_{n}^{(l)}} g_{i,n}^{(l)} s(t - \tau_{i,n}^{(l)}) + g_{n}^{(l)}(t) + \omega_{\text{GWN}}(t) \tag{1}$$

The first term on the right in Equation (1) describes $L_{n}^{(l)}$ specular MPCs with complex amplitude $g_{i,n}^{(l)}$ and time delay $\tau_{i,n}^{(l)}$, where $l \in \{1,...,L_{n}^{(l)}\}$. These MPCs correspond to the features (PTs and VTs) of the environment. The delay $\tau_{i,n}^{(l)}$ corresponds to the distance between the UE and the $j$th PT ($l = 1$) or the associated VTs ($l \in \{2,...,L_{n}^{(l)}\}$); thus, we have $\tau_{i,n}^{(l)} = \frac{\|u_n - a_{i,j}^{(l)}\|}{c}$, where $c$ is the speed of light. Furthermore, $g_{n}^{(l)}(t)$ in (1) denotes the diffuse MPCs, which is regarded as interference. Finally, $\omega_{\text{GWN}}(t)$ represents additive white Gaussian noise.

At each time $n$, the radio channel estimator [35] processes the signal $s_{RX,n}(t)$ received by the UE to obtain $M_{n}^{(l)}$ delay estimations of MPCs $\hat{\tau}_{i,n}^{(l)}, m \in M_{n}^{(l)} = \{1,...,M_{n}^{(l)}\}$. The relation between the number of MPC delay estimations $M_{n}^{(l)}$ and the number of specular MPCs $L_{n}^{(l)}$ is as follows: on the one hand, some specular MPCs are unlikely to be detected by the radio channel estimator without generating delay estimations; on the other hand, some of the delay estimations cannot correspond to any specular MPCs. Therefore, there is no exact equality between $M_{n}^{(l)}$ and $L_{n}^{(l)}$, and $M_{n}^{(l)}$ mainly depends on the UE

At each time $n$, the radio channel estimator [35] processes the signal $s_{RX,n}(t)$ received by the UE to obtain $M_{n}^{(l)}$ delay estimations of MPCs $\hat{\tau}_{i,n}^{(l)}, m \in M_{n}^{(l)} = \{1,...,M_{n}^{(l)}\}$. The relation between the number of MPC delay estimations $M_{n}^{(l)}$ and the number of specular MPCs $L_{n}^{(l)}$ is as follows: on the one hand, some specular MPCs are unlikely to be detected by the radio channel estimator without generating delay estimations; on the other hand, some of the delay estimations cannot correspond to any specular MPCs. Therefore, there is no exact equality between $M_{n}^{(l)}$ and $L_{n}^{(l)}$, and $M_{n}^{(l)}$ mainly depends on the UE
position \( u \) and the environment. The estimated distances \( z_{m,n}^{(j)}, m \in \{1, \ldots, M_n^{(j)}\} \) of the  
\( m \)th MPC of PT \( j \) are obtained by multiplying the delay estimation \( \hat{\theta}_{m,n}^{(j)} \), with the speed of  
light \( c \), which are the measurements of the proposed algorithm used in this paper. For  
convenience, we define the stacked vectors of \( z_{m,n}^{(j)}, z_{n}^{(j)} = \begin{bmatrix} z_{n1}^{(j)} & \cdots & z_{nM_n^{(j)}}^{(j)} \end{bmatrix}^T \),  
\( z_{n}^{(j)} = \begin{bmatrix} z_{11}^{(j)} & \cdots & z_{1M_n^{(j)}}^{(j)} \end{bmatrix}^T \) and the stacked vectors of \( M_n^{(j)} \),  
\( m_{n}^{(j)} = \begin{bmatrix} M_{n1}^{(j)} & \cdots & M_{nM_n^{(j)}}^{(j)} \end{bmatrix}^T \) and \( m = \begin{bmatrix} m_{1}^T & \cdots & m_{M_n^{(j)}}^T \end{bmatrix}^T \).

3. System Model with Unknown Process Noise and Clutter Intensity

3.1. UE State and PF States

The UE state at time \( n \) is \( \hat{x}_n = \begin{bmatrix} u_n^T \ v_n^T \end{bmatrix}^T \) and \( v_n \) denotes the UE velocity vector.  
Potential features (PFs) are used to describe the features in the environment map (PTs and  
VTs) because the number of VTs is unknown and time-varying. PFs are indexed via \( (j, k) \),  
where \( j \in \{1, \ldots, J\} \) and \( k \in K_n^{(j)} = \begin{bmatrix} 1, \ldots, K_n^{(j)} \end{bmatrix} \) (which implies that there are \( K_n^{(j)} \) PFs  
for each PT \( j \)). Note that the set of PFs includes the PT itself, and the number of PFs is  
unknown and varies with the UE position. We use the indicator variable \( \ell_{k,a}^{(j)} = \{0,1\} \) to  
indicate whether PFs truly exist, where \( \ell_{k,a}^{(j)} = 1 \) means that PF \((j, k)\) exists at time \( n \), and  
vice versa. The indicator variable \( \ell_{k,a}^{(j)} \) is used to estimate the existence probability of PFs,  
which will be introduced in Section 4.1. The state of the PF \((j, k)\) at time \( n \) is represented  
as an augmented state including the indicator variable \( \ell_{k,a}^{(j)} \), i.e., \( \hat{y}_{k,a}^{(j)} = \begin{bmatrix} a_{k,a}^{(j)} \ e_{k,a}^{(j)} \end{bmatrix}^T \), where \( a_{k,a}^{(j)} \) denotes the PF \((j, k)\) position. We also define the stacked vectors of \( \hat{y}_{k,a}^{(j)} \), i.e.,  
\( \hat{y}_{k,a}^{(j)} = \begin{bmatrix} y_{k,a}^{(j)} & \cdots & y_{k,M_n^{(j)}}^{(j)} \end{bmatrix}^T \), \( y_n = \begin{bmatrix} y_1^{(j)} & \cdots & y_{M_n^{(j)}}^{(j)} \end{bmatrix}^T \) and \( y = \begin{bmatrix} y_1^T & \cdots & y_{M_n^{(j)}}^T \end{bmatrix}^T \).

At any time \( n \), two kinds of PFs exist: one is the legacy PF, which was established in  
the past, and the other is the new PF, which is established for the first time at time \( n \). The  
augmented states of legacy PFs and new PFs for PT \( j \) are expressed as \( \hat{y}_{k,a}^{(j)} = \begin{bmatrix} a_{k,a}^{(j)} \ e_{k,a}^{(j)} \end{bmatrix}^T \),  
\( k \in K_n^{(j)} \) and \( \hat{y}_{m,a}^{(j)} = \begin{bmatrix} a_{m,a}^{(j)} \ e_{m,a}^{(j)} \end{bmatrix}^T \), \( m \in M_n^{(j)} \), respectively. The number of new PFs is  
equal to the number of measurements \( M_n^{(j)} \) at time \( n \); thus, the set and number of legacy  
PFs are updated as

\[
K_n^{(j)} = K_n^{(j)} \cup M_n^{(j)}, \quad K_n^{(j)} = K_n^{(j)} + M_n^{(j)}
\]

From Equation (2), we see that the PF set updates as a dynamic increasing process,  
i.e., legacy PFs at time \( n - 1 \) and new PFs at time \( n \) constitute the legacy PFs at time \( n + 1 \).  
(Note that the number of PFs \( K_n^{(j)} \) at time \( n \) does not actually increase \( M_n^{(j)} \) compared  
with that at time \( n - 1 \), since some PFs with low existence probability are pruned, which  
will be explained in Section 4.1.) We also define the stacked vectors related to PF states as  
follows: for legacy PFs for PT \( j \), \( \hat{y}_{k,a}^{(j)} = \begin{bmatrix} a_{k,a}^{(j)} & \cdots & a_{k,M_n^{(j)}}^{(j)} \end{bmatrix}^T \), \( e_{k,a}^{(j)} = \begin{bmatrix} e_{k,a}^{(j)} & \cdots & e_{k,M_n^{(j)}}^{(j)} \end{bmatrix}^T \) and  
\( \hat{y}_{k,a}^{(j)} = \begin{bmatrix} y_{k,a}^{(j)} & \cdots & y_{k,M_n^{(j)}}^{(j)} \end{bmatrix}^T \); for new PFs for PT \( j \), \( \hat{y}_{m,a}^{(j)} = \begin{bmatrix} a_{m,a}^{(j)} & \cdots & a_{m,M_n^{(j)}}^{(j)} \end{bmatrix}^T \),  
\( e_{m,a}^{(j)} = \begin{bmatrix} e_{m,a}^{(j)} & \cdots & e_{m,M_n^{(j)}}^{(j)} \end{bmatrix}^T \) and \( \hat{y}_{m,a}^{(j)} = \begin{bmatrix} y_{m,a}^{(j)} & \cdots & y_{m,M_n^{(j)}}^{(j)} \end{bmatrix}^T \); and for the combination of legacy PFs
and new PFs for PT \( j \), \( Y_{n}^{(j)} = \left[ Y_{n, k}^{(j)T} \right]^{T} \). Note that the entries of \( Y_{n}^{(j)} \) are \( Y_{n, k}^{(j)} \) and \( k \in \mathcal{A}_{m}^{(j)} \); for all legacy PFs and new PFs, \( Y_{n}^{(j)} = \left[ Y_{n, k}^{(j)T} \right]^{T} \) and \( Y_{n}^{(j)} = \left[ Y_{n, k}^{(j)T} \right]^{T} \).

For each PT, the measurements \( z_{n, m}^{(j)} \) are affected by the DA (measurement source) uncertainty. In other words, it is hard to know which measurement \( z_{n, m}^{(j)} \) corresponds to which PF \( k \in \mathcal{A}_{m}^{(j)} \). The measurement \( z_{n, m}^{(j)} \) may not originate from any PF (false alarms or clutter), or it may not be detected by the channel estimator (missed detections). We assume that each PF generates at most one measurement at any time \( n \) and that each measurement can be generated by at most one PF [36]. At time \( n \), the association between the measurements \( z_{n, m}^{(j)} \) and PFs can be expressed as a feature-oriented \( \mathcal{K}_{n}^{(j)} \) -dimensional DA vector \( h_{n}^{(j)} = \left[ h_{n, k}^{(j)} \right]^{T} \). If the legacy PF \( (j, k) \) generates a measurement \( z_{n, m}^{(j)} \), the \( k \)th entry of \( h_{n}^{(j)} \) is \( h_{n, k}^{(j)} = m \in \mathcal{A}_{m}^{(j)} \); if it does not generate any measurement, then we have \( h_{n, k}^{(j)} = 0 \). Furthermore, we consider a measurement-oriented \( M_{n}^{(j)} \)-dimensional DA vector \( q_{n}^{(j)} = \left[ q_{n, m}^{(j)} \right]^{T} \). If the measurement \( z_{n, m}^{(j)} \) originates from the legacy PF \( (j, k) \), the \( m \)th entry of \( q_{n}^{(j)} \) is \( q_{n, m}^{(j)} = k \in \mathcal{A}_{m}^{(j)} \); if the measurement \( z_{n, m}^{(j)} \) does not originate from any legacy PF, then \( q_{n, m}^{(j)} = 0 \). Additionally, we define the stacked vectors of DA vectors \( h_{n} = \left[ h_{n, k}^{(j)T} \right]^{T} \), \( h = \left[ h_{n, k}^{(j)T} \right]^{T} \), \( q_{n} = \left[ q_{n, m}^{(j)T} \right]^{T} \), and \( q = \left[ q_{n, m}^{(j)T} \right]^{T} \). The DA vectors \( h \) and \( q \) are random and completely equivalent since one of them can be determined by the other.

3.2. Markov Chain Modeling of the Process Noise and the Clutter Intensity

In addition to the essential information to be tracked, i.e., the UE state, PF states and DA vectors, model parameters such as the process noise and the clutter intensity are also unknown and time-varying. The process noise determines the difference of the UE dynamic models (assuming that the UE is in a nearly constant velocity model), and the clutter intensity describes the amount of clutter (false alarms) in the environment. Significant mismatches in the process and clutter model will seriously affect the SLAM accuracy. To obtain a more robust multipath-assisted SLAM algorithm, we propose tracking unknown model parameters, process noise and clutter intensity in the framework of the BP-SLAM algorithm. The process noise and the clutter intensity can be modeled in the same way: \( \rho_{n}^{(s)} \), \( s = 1, \ldots, S \) is assumed to be an unknown model parameter. We model \( \rho_{s}^{(s)} \) as a time-varying discrete random variable taking values from a finite set \( \mathcal{A} = \{ \alpha_{1}^{(s)}, \ldots, \alpha_{N_{s}}^{(s)} \} \) and assume that the distribution of the initial parameter \( \rho_{1}^{(s)} \) corresponds to a probability mass function (PMF) \( p(\rho_{1}^{(s)}) \), \( \rho_{1}^{(s)} \in \mathcal{A} \). Additionally, we assume that the evolution \( \rho_{n}^{(s)} \) is independent, which corresponds to the first-order Markov chain model with transition matrices \( P_{s} \in [0,1]^{N_{s} \times N_{s}} \), where \( [0,1]^{N_{s} \times N_{s}} \) refers to the elements in the matrix with size \( N_{s} \times N_{s} \) being \([0,1]\). Thus, the PMF of the state transition of \( \rho_{n}^{(s)} \) is

\[
p(\rho_{n}^{(s)}) = \omega_{n}^{(s)} | \rho_{n-1}^{(s)} = \alpha_{n}^{(s)} = [P_{s}]_{n,n-1}
\]  

(3)
Note that \( \sum_{i=1}^{N_x} \mathbf{P}_{x,i}^{(i)} \) is the stacked vector of \( \rho_n^{(i)} \)
\[
\mathbf{p} = \begin{bmatrix} \rho_n^{(1)} & \cdots & \rho_n^{(S)} \end{bmatrix}^T,
\]
where \( \mathbf{p} = \begin{bmatrix} \rho_n^{(1)} & \cdots & \rho_n^{(S)} \end{bmatrix}^T \). Based on the assumptions above, the prior PMF of \( \mathbf{p} \) factorizes as
\[
p(\mathbf{p}) = \prod_{i=1}^{N_x} p(\rho_n^{(i)}) \prod_{n=2}^{N_x} p(\rho_n^{(i)}) \rho_n^{(i-1)}
\] (4)

3.3. State Evolution with Unknown Process Noise

The probability density function (PDF) of the UE state transition is determined by the process noise variance if the UE is in a nearly constant velocity model. Using the interacting multiple model (IMM) algorithm [37], the UE can switch among nearly constant velocity models with different process noise variances at any time. To track the change of the UE dynamic model, the original state of the UE \( \tilde{x}_n \) is augmented by the UE process noise variance as \( x_n = \begin{bmatrix} \tilde{x}_n^T & \sigma_n^2 \end{bmatrix}^T \), where \( \sigma_n^2 \) denotes the process noise variance of the UE at time \( n \). We model the process noise variance \( \sigma_n^2 \) as a discrete random variable taking values from a finite set \( \mathcal{B} = \{\theta_1, \ldots, \theta_{N_e}\} \), the evolution of which is based on the Markov chain model of Section 3.2 with the transition matrix \( Q_e \in [0,1]^{N_e \times N_e} \). Thus, the PMF of the state transition of \( \sigma_n^2 \) is
\[
p(\sigma_n^2 = \theta_j | \sigma_{n-1}^2 = \theta_i) = \begin{bmatrix} \mathbf{Q}_{e,i,j} \end{bmatrix}, \quad \theta_i, \theta_j \in \mathcal{B}
\] (5)

Additionally, we define the stacked vectors \( \sigma = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_n^2 \end{bmatrix} \) and \( x = \begin{bmatrix} x_1^T & \cdots & x_n^T \end{bmatrix} \). The evolutions of the UE augmented state \( x_n \) and the legacy PF augmented state \( y_n \) are assumed to be independent and based on the Markov state model. Their evolutions can be jointly expressed as
\[
f(x_n, y_n | x_{n-1}, y_{n-1}) = f(\tilde{x}_n, \sigma_n^2 | \tilde{x}_{n-1}, \sigma_{n-1}^2) f(y_n | y_{n-1})
\]
\[
= f(\tilde{x}_n, \sigma_n^2 | \tilde{x}_{n-1}) p(\sigma_n^2 | \sigma_{n-1}^2)
\]
\[
\times \prod_{j=1}^{N_e} \prod_{k=1}^{N_e} f(\mathbf{y}_{j,k}^{(j)} | y_{k,n-1}^{(j)})
\] (6)

where \( f(\tilde{x}_n, \sigma_n^2 | \tilde{x}_{n-1}) \) denotes the state transition PDF of the UE with process noise variance \( \sigma_n^2 \), and we assume that the UE original states \( \tilde{x}_n \) and \( \sigma_n^2 \) are statistically independent across time \( n \) here; \( f(\mathbf{y}_{j,k}^{(j)} | y_{k,n-1}^{(j)}) = f(\mathbf{a}_{j,k}^{(j)} | \mathbf{a}_{k,n-1}^{(j)}, e_{k,n-1}^{(j)}) \) denotes the state transition PDF of the legacy PF \( (j, k) \). If PF \( (j, k) \) exists at time \( n-1 \) \( (e_{k,n-1}^{(j)} = 1) \), it either disappears \( (e_{k,n}^{(j)} = 0) \) or continues to exist \( (e_{k,n}^{(j)} = 1) \) at time \( n \). In the latter case, PF \( (j, k) \) becomes a legacy PF at time \( n \). If PF \( (j, k) \) survives at time \( n \), its update can be represented by the state transition PDF \( f(\mathbf{a}_{j,k}^{(j)} | \mathbf{a}_{k,n-1}^{(j)} = 1) \). Thus, when \( e_{k,n}^{(j)} = 1 \), \( f(\mathbf{y}_{j,k}^{(j)} | y_{k,n-1}^{(j)}) \) in Equation (6) is expressed as
\[
f(\mathbf{a}_{j,k}^{(j)} | \mathbf{a}_{k,n-1}^{(j)}, e_{k,n}^{(j)} = 1) \]
\[
= \left\{ \begin{array}{ll} (1 - P) f(\mathbf{a}_{j,k}^{(j)}), & e_{k,n}^{(j)} = 0 \\ P_j f(\mathbf{a}_{j,k}^{(j)} | \mathbf{a}_{k,n-1}^{(j)}), & e_{k,n}^{(j)} = 1 \\ \end{array} \right.
\] (7)
where \( P_s \) denotes the survival probability of the PF and \( f_D(\mathbf{a}^{(j)}_{s,a}) \) denotes an arbitrary “dummy PDF” [38]. If PF \((j, k)\) does not exist at time \( n - 1 \), it cannot exist at time \( n \). Thus, we have

\[
f(\mathbf{a}^{(j)}_{s,a}, \mathbf{z}^{(j)}_{s,a} | \mathbf{a}^{(j)}_{s,a-1}, \mathbf{z}^{(j)}_{s,a-1} = 0) = \begin{cases} f_D(\mathbf{a}^{(j)}_{s,a}), & \mathbf{z}^{(j)}_{s,a} = 0 \\ 0, & \mathbf{z}^{(j)}_{s,a} = 1 \end{cases}
\]  

(8)

### 3.4. Prior Distributions with Unknown Clutter Intensity

Clutter are false measurements in the environment, which are usually modeled as Poisson point processes; hence, all the characteristics of the clutter can be completely described by the intensity information. The clutter intensity is defined as the product of the average number of clutter at each time \( \lambda^{(j)}_{CL,a} \) and the spatial distribution PDF of clutter \( f_{CL}(z^{(j)}_{m,a}) \). \( f_{CL}(z^{(j)}_{m,a}) \) is generally considered to be a priori; therefore, the clutter intensity can only be determined by the average number of clutter \( \lambda^{(j)}_{CL,a} \). \( \lambda^{(j)}_{CL,a} \) is assumed to be statistically independent across \( j \) and modeled as a discrete random variable with values taken from a finite set \( \mathcal{G} = \{ r_1, \ldots, r_{N_g} \} \). The evolution of \( \lambda^{(j)}_{CL,a} \) is based on the Markov chain model of Section 3.2 with the transition matrix \( \mathbf{G}^{(j)} \in [0,1]^{N_g \times N_g} \), where \( \mathbf{G}^{(j)} \) is equal at all times and only related to PT. The initial distribution of \( \lambda^{(j)}_{CL,a} \) is given by the PMF \( p(\lambda^{(j)}_{CL,a}) \). We also define the stacked vector \( \mathbf{\lambda}_{CL} = \begin{bmatrix} \lambda_{CL,1}^T \ldots \lambda_{CL,n}^T \end{bmatrix}^T \), where \( \lambda_{CL,a} = \begin{bmatrix} \lambda^{(j)}_{CL,1} \ldots \lambda^{(j)}_{CL,a} \end{bmatrix}^T \). According to (2), the prior PMF of \( \mathbf{\lambda}_{CL} \) factorizes as

\[
p(\mathbf{\lambda}_{CL}) = \prod_{j=1}^{M} p(\lambda^{(j)}_{CL,1}) \prod_{n=2}^{N} p(\lambda^{(j)}_{CL,n} | \lambda^{(j)}_{CL,n-1})
\]  

(9)

where \( p(\lambda^{(j)}_{CL,n} = r_j | \lambda^{(j)}_{CL,n-1} = r_i) = \mathbf{G}^{(j)} \). Similar to clutter, we model the number of newly detected features as a Poisson point process with mean value \( \lambda^{(j)}_{new,a} \). We also define the detection probability \( P_d^{(j)}(x_a, \mathbf{a}^{(j)}_s) \), which refers to the probability that the PF can be detected or the probability that the PF can generate the measurement \( z^{(j)}_{m,a} \) in the channel estimation process [39]. The detection probability \( P_d^{(j)}(x_a, \mathbf{a}^{(j)}_s) \in (0,1) \) is supposed known in this paper.

Then, given the UE augmented state \( x_a \), the legacy PF augmented state \( \mathbf{z}^{(j)}_{a,a} \) and the average number of clutter \( \lambda^{(j)}_{CL,a} \), the joint prior conditional PMF of the association vectors \( \mathbf{h}^{(j)}_s \) and \( \mathbf{q}^{(j)}_s \), the existence indicator variable of new PFs \( \mathbf{\tau}^{(j)}_s \) and the number of measurements \( M^{(j)}_s \) is expressed as [38]

\[
p(h^{(j)}_s, q^{(j)}_s, \mathbf{\tau}^{(j)}_s, M^{(j)}_s | x_a, \mathbf{z}^{(j)}_{a,a}, \mathbf{z}^{(j)}_{CL,a}) \propto \psi(h^{(j)}_s, q^{(j)}_s) \prod_{m=1}^{M^{(j)}_s} c_l(\mathbf{\tau}^{(j)}_m, q^{(j)}_m, \lambda^{(j)}_{CL,a}, K^{(j)}_a) \times \prod_{k=1}^{K^{(j)}_a} b_{l}(x_a, \mathbf{a}^{(j)}_s, \mathbf{z}^{(j)}_{CL,a}, h^{(j)}_s, M^{(j)}_s)
\]  

(10)

The exclude indicator function in (10) is
\[ y(\mathbf{h}_{k,n}^{(i)}, \mathbf{q}_{m,n}^{(j)}) = \prod_{k=1}^{K} \prod_{m=1}^{M} \phi(h_{k,n}^{(i)}, q_{m,n}^{(j)}) \]  

(11)

where when \( h_{k,n}^{(i)} = m \), \( q_{m,n}^{(j)} \neq k \) or \( q_{m,n}^{(j)} = k \), \( h_{k,n}^{(i)} \neq m \), \( \phi(h_{k,n}^{(i)}, q_{m,n}^{(j)}) = 0 \); otherwise, \( \phi(h_{k,n}^{(i)}, q_{m,n}^{(j)}) = 1 \). The indicator function enforces the assumption in Section 3.1 that each PF generates at most one measurement at any time \( n \) and that each measurement can be generated by at most one PF. Note that this DA constraint function is redundant since it considers both the feature-oriented and measurement-oriented association vectors \( \mathbf{h}_{k,n}^{(i)} \) and \( \mathbf{q}_{m,n}^{(j)} \). However, such a redundant representation is the key to obtaining an algorithm that has low computational complexity and excellent scalability [38,40]. Furthermore, \( c_i(\overline{\mathbf{e}}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}; K_{m,n}^{(j)}) \) and \( b_i(x_n, \overline{\mathbf{e}}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, M_{m,n}^{(j)}) \) in (10) are defined as

\[ c_i(1, q_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}; K_{m,n}^{(j)}) = \begin{cases} 0, & q_{m,n}^{(j)} \in K_{m,n}^{(j)} \\ \mathbf{q}_{m,n}^{(j)}, & q_{m,n}^{(j)} = 0 \end{cases} \]  

(12)

\[ c_i(0, q_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}; K_{m,n}^{(j)}) = f_D(\mathbf{a}_{m,n}^{(j)}) \]  

(13)

\[ b_i(x_n, \overline{\mathbf{e}}_{m,n}^{(j)}, \mathbf{h}_{k,n}^{(i)}, \mathbf{q}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, M_{m,n}^{(j)}) = \begin{cases} 0, & \mathbf{h}_{k,n}^{(i)} \in M_{m,n}^{(j)} \\ 1 - P_{2}^{(i)}(x_n, \mathbf{a}_{m,n}^{(j)}), & \mathbf{h}_{k,n}^{(i)} = 0 \end{cases} \]  

(14)

\[ b_i(x_n, \overline{\mathbf{e}}_{m,n}^{(j)}, 0, \mathbf{h}_{k,n}^{(i)}, \mathbf{q}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, M_{m,n}^{(j)}) = 1(\mathbf{h}_{k,n}^{(i)}) \]  

(15)

where the indicator function \( 1(\mathbf{h}_{k,n}^{(i)}) \) is defined as \( 1(\mathbf{h}_{k,n}^{(i)})=1 \) when \( \mathbf{h}_{k,n}^{(i)}=0 \); otherwise, \( 1(\mathbf{h}_{k,n}^{(i)})=0 \). In particular, when \( n = 1 \), since the legacy PF \( \mathbf{a}_{m,n}^{(j)} \) is an empty vector, we have

\[ p(h_{k,n}^{(i)}, q_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, M_{1}^{(j)}|x_1, y_1^{(j)}, \mathbf{q}_{m,n}^{(j)}, A_{CH}^{(j)}) = \underbrace{p(h_{k,n}^{(i)}, q_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, M_{1}^{(j)}|x_1, \mathbf{q}_{m,n}^{(j)})}_{f_D(\mathbf{a}_{m,n}^{(j)})} \]  

Assume that the spatial distribution PDF of newly detected features is \( f_{\text{new},a}(\overline{\mathbf{a}}_{m,n}^{(j)}) \) (the calculation of which will be discussed in Section 5.1); then, for PT \( j \), given \( \overline{\mathbf{a}}_{m,n}^{(j)} \) and \( M_{m,n}^{(j)} \), the prior PDF of the new PF state \( \overline{\mathbf{a}}_{m,n}^{(j)} \) is

\[ f(\overline{\mathbf{a}}_{m,n}^{(j)}, \mathbf{q}_{m,n}^{(j)}, M_{m,n}^{(j)}) = \prod_{m \in M_{m,n}^{(j)}} f_{\text{new},a}(\overline{\mathbf{a}}_{m,n}^{(j)}) \prod_{m \in M_{m,n}^{(j)}} f_D(\overline{\mathbf{a}}_{m,n}^{(j)}) \]  

(16)

where \( \overline{\mathbf{a}}_{m,n}^{(j)} \) and \( M_{m,n}^{(j)} \) are \( m \in M_{m,n}^{(j)}; \mathbf{a}_{m,n}^{(j)} = 1 \).
Remote Sens. 2021, 13, 1625

\[ f(x, y, h, q, m, \lambda_{CL}) = f(x) \left( \prod_{j=1}^{J} p(\lambda_{CL,j}) f(y_j^{(j)}, h_j^{(j)}, q_j^{(j)}, M_j^{(j)}) \left| x, \lambda_{CL,j}^{(j)} \right| \right) \]

\[ \times \prod_{n=2}^{N} f(x_n, s_n^2, \lambda_{CL,n}) p(\sigma_n^2 | \sigma_{n-1}^2) \]

\[ \times \prod_{j=1}^{J} \left( \prod_{k=1}^{K} f_y(y_j^{(j)}, y_j^{(j)}) \right) p(\lambda_{CL,n}^{(j)}, q_n^{(j)}, \vec{x}_n^{(j)}, M_n^{(j)}) \left| x_n, \lambda_{CL,n}^{(j)} \right| \]

\[ \times f(\vec{x}_n^{(j)}, M_n^{(j)}) p(\kappa_n^{(j)}, q_n^{(j)}, \vec{x}_n^{(j)}, M_n^{(j)}) \left| x_n, \lambda_{CL,n}^{(j)} \right| \]

where \( f(y_j^{(j)}, h_j^{(j)}, q_j^{(j)}, M_j^{(j)}) \left| x, \lambda_{CL,j}^{(j)} \right| \) (see Equation (10) and Equation (16)), and the last four factors are obtained from Equation (6), Equation (9), Equation (10) and Equation (16), respectively.

3.5. Measurement Model and Likelihood Function

The relation between the measurement \( z_{m,n}^{(j)} \) and the position-dependent states \( \vec{x}_n \) and \( \vec{a}_n^{(j)} \) is described by the conditional PDF \( f(z_{m,n}^{(j)} \left| \vec{x}_n, \vec{a}_n^{(j)} \right|) \), an example of which will be given in Section 5.1. This PDF is the central element of the likelihood function \( f(z_{m,n}^{(j)} \left| \vec{x}_n, \vec{y}_n^{(j)}, \vec{y}_n^{(j)}, h_n^{(j)}, M_n^{(j)} \right|) \), which is a function of \( \vec{x}_n, \vec{y}_n^{(j)}, h_n^{(j)} \) and \( M_n^{(j)} \) when the measurement \( z_{m,n}^{(j)} \) is observed. Assuming that \( z_{m,n}^{(j)} \) is observed, i.e., \( z_{m,n}^{(j)} \) is fixed, then, up to a constant factor, the likelihood function can be expressed as [25]

\[ f(z_{m,n}^{(j)} \left| \vec{x}_n, \vec{y}_n^{(j)}, \vec{y}_n^{(j)}, h_n^{(j)}, M_n^{(j)} \right| \propto \prod_{m,n,j} \frac{f(z_{m,n}^{(j)} \left| \vec{x}_n, \vec{y}_n^{(j)} \right|, h_n^{(j)}, M_n^{(j)} \right|)}{f(z_{m,n}^{(j)})} \]

\[ \times \prod_{k=1}^{K} b_2(x_n, \vec{a}_n^{(j)}, \vec{z}_n^{(j)}, \vec{h}_n^{(j)}, z_{m,n}^{(j)}) \]

where \( b_2(x_n, \vec{a}_n^{(j)}, \vec{z}_n^{(j)}, \vec{h}_n^{(j)}, z_{m,n}^{(j)}) \) is defined as

\[ b_2(x_n, \vec{a}_n^{(j)}, \vec{z}_n^{(j)}, \vec{h}_n^{(j)}, z_{m,n}^{(j)}) \triangleq \begin{cases} \frac{f(z_{m,n}^{(j)} \left| \vec{x}_n, \vec{a}_n^{(j)} \right|, h_n^{(j)}}{f(z_{m,n}^{(j)})}, & h_n^{(j)} = m \\ 1, & h_n^{(j)} = 0 \end{cases} \]

and \( b_2(x_n, \vec{a}_n^{(j)}, 0, p_n^{(j)}, z_{m,n}^{(j)}) \triangleq 1 \). In particular, at the initial time when \( n = 1 \), we have \( f(z_{m,1}^{(j)} \left| \vec{x}_1, \vec{y}_1^{(j)}, \vec{y}_1^{(j)}, h_1^{(j)}, M_1^{(j)} \right|) = f(z_{m,1}^{(j)} \left| \vec{x}_1, \vec{y}_1^{(j)} \right|, h_1^{(j)}, M_1^{(j)}) \). Based on Equation (18), the likelihood function of the stacked vector \( z \) of the measurement \( z_{m,n}^{(j)} \) involving all PTs \( j = 1, \ldots, J \) and all times \( n' = 1, \ldots, n \) is
Now, we consider the product of the likelihood function Equation (18), the prior PDF of the association vectors Equation (10) and the prior PDF of new PFs Equation (16), and we have
\[
f(z|\mathbf{x}, y, h, m) \propto \prod_{j=1}^{J} \left( \prod_{m \in \mathcal{M}_j} \frac{f(z_{i,j}^{(m)}|\mathbf{x}_m, \mathbf{a}_m^{(j)})}{f_{CL}(z_{i,j}^{(m)})} \right) \times \prod_{m \in \mathcal{M}_j} \left( \prod_{n \in \mathcal{N}_m} \frac{f(z_{i,n}^{(m)}|\mathbf{x}_n, \mathbf{a}_n^{(m)})}{f_{CL}(z_{i,n}^{(m)})} \right) \times \prod_{m \in \mathcal{M}_j} b_2(\mathbf{x}_m, \mathbf{a}_m^{(j)}, \mathbf{z}_{i,j}^{(m)}; h_{i,j}^{(m)}, \mathbf{z}_{i,n}^{(m)})
\]
(20)

where
\[
b_j(\ldots, z_{i,j}^{(m)}) \equiv b_1(\ldots, h_{i,j}^{(m)}) b_2(\ldots, z_{i,j}^{(m)})
\]
and
\[
c_j(\ldots, z_{i,j}^{(m)}) \equiv c_1(\ldots, K_{i,j}^{(m)}) c_2(\ldots, z_{i,j}^{(m)}) \quad b_1(\ldots, h_{i,j}^{(m)}) \quad b_2(\ldots, z_{i,j}^{(m)}) \quad \text{and} \quad c_1(\ldots, K_{i,j}^{(m)})
\]
are given by Equation (14), Equation (12) and Equation (19), respectively, and
\[
c_2(\mathbf{x}_m, \mathbf{a}_m^{(j)}, \mathbf{z}_{i,j}^{(m)}, q_{m,a}^{(j)}; z_{i,n}^{(m)})
\]
is defined as
\[
c_2(\mathbf{x}_m, \mathbf{a}_m^{(j)}, 0, q_{m,a}^{(j)}; z_{i,n}^{(m)}) \equiv \begin{cases} 0, & q_{m,a}^{(j)} = k \\ \frac{f_{new,a}(\mathbf{a}_m^{(j)}) f(z_{i,j}^{(m)}|\mathbf{x}_m, \mathbf{a}_m^{(j)})}{f_{CL}(z_{i,j}^{(m)})}, & q_{m,a}^{(j)} = 0 \end{cases}
\]
(22)
and
\[
c_2(\mathbf{x}_m, \mathbf{a}_m^{(j)}, 0, q_{m,a}^{(j)}; z_{i,n}^{(m)}) = f_D(\mathbf{a}_m^{(j)})
\]
and
\[
c(\mathbf{x}_m, \mathbf{a}_m^{(j)}, \mathbf{z}_{i,j}^{(m)}, q_{m,a}^{(j)}; z_{i,n}^{(m)}) \quad \text{can be expressed as}
\]
\[
b(\mathbf{x}_m, \mathbf{a}_m^{(j)}, 0, h_{i,j}^{(m)}; z_{i,j}^{(m)}, z_{i,n}^{(m)}) = \begin{cases} 0, & h_{i,j}^{(m)} = m \\ \frac{P_d^{(j)}(\mathbf{x}_m, \mathbf{a}_m^{(j)}) f(z_{i,j}^{(m)}|\mathbf{x}_m, \mathbf{a}_m^{(j)})}{f_{CL}(z_{i,j}^{(m)})}, & h_{i,j}^{(m)} = 0 \end{cases}
\]
(23)
and
\[
b(\mathbf{x}_m, \mathbf{a}_m^{(j)}, 0, h_{i,j}^{(m)}; z_{i,j}^{(m)}, z_{i,n}^{(m)}) = f_D(\mathbf{a}_m^{(j)})
\]
and
\[
c(\mathbf{x}_m, \mathbf{a}_m^{(j)}, 0, q_{m,a}^{(j)}; z_{i,j}^{(m)}, z_{i,n}^{(m)}) \equiv \begin{cases} 0, & q_{m,a}^{(j)} = k \\ \frac{f_{new,a}(\mathbf{a}_m^{(j)}) f(z_{i,j}^{(m)}|\mathbf{x}_m, \mathbf{a}_m^{(j)})}{f_{CL}(z_{i,j}^{(m)})}, & q_{m,a}^{(j)} = 0 \end{cases}
\]
(24)
and
\[
c(\mathbf{x}_m, \mathbf{a}_m^{(j)}, 0, q_{m,a}^{(j)}; z_{i,j}^{(m)}, z_{i,n}^{(m)}) = f_D(\mathbf{a}_m^{(j)})
\]
4. The Proposed Algorithm

In this section, the proposed BP-based robust multipath-assisted SLAM algorithm is described in detail.

4.1. Detection and Estimation

The ultimate goal of our algorithm is to estimate the UE original state \( \hat{x}_n \) and PF states \( \hat{a}_{k,n}^{(j)} \) from the measurements \( \mathbf{z} \) involving all PTs and all times with unknown process noise and clutter intensity. In the framework of the Bayesian theorem, estimations of the UE and PF states are based on their respective posterior PDFs. First, to estimate \( \hat{x}_n \), we use the minimum mean square error (MMSE) estimator, i.e.,

\[
\hat{x}_n^{\text{MMSE}} = \int \hat{x}_n f(\hat{x}_n, z) d\hat{x}_n
\]  

To obtain \( f(\hat{x}_n | z) \), we have to marginalize the posterior PDF of the UE augmented state \( f(x_n | z) = f(\hat{x}_n, \sigma_n^2 | z) \), i.e.,

\[
f(\hat{x}_n | z) = \sum_{\sigma_n^2 \in \Theta} f(\hat{x}_n, \sigma_n^2 | z)
\]

Furthermore, detecting whether PFs exist at time \( n \) depends on the posterior existence probability \( p(\epsilon_{k,n}^{(j)} = 1 | z) \), which is obtained by marginalizing the posterior PDF of PF augmented states

\[
p(\epsilon_{k,n}^{(j)} = 1 | z) = \int f(\epsilon_{k,n}^{(j)}, x_{k,n}^{(j)} = 1 | z) d\epsilon_{k,n}^{(j)}
\]

Then, if \( p(\epsilon_{k,n}^{(j)} = 1 | z) > P_{\text{th}} \), PF \( k \in \mathcal{K}_n^{(j)} \) is detected at time \( n \), where \( P_{\text{th}} \) is the detection threshold. Thus, the estimation of the state of the detected PF \( \hat{a}_{k,n}^{(j)} \) is

\[
\hat{a}_{k,n}^{(j)\text{MMSE}} = \int a_{k,n}^{(j)} f(a_{k,n}^{(j)} | \epsilon_{k,n}^{(j)} = 1, z) d\epsilon_{k,n}^{(j)}
\]

where

\[
f(a_{k,n}^{(j)} | \epsilon_{k,n}^{(j)} = 1, z) = \frac{f(a_{k,n}^{(j)}, \epsilon_{k,n}^{(j)} = 1 | z)}{p(\epsilon_{k,n}^{(j)} = 1 | z)}
\]

Therefore, to estimate \( \hat{x}_n \) and \( \hat{a}_{k,n}^{(j)} \), we must calculate the marginal posterior PDFs \( f(x_n | z) \) and \( f(y_{k,n}^{(j)} | z) \) of the augmented states of the UE and PFs. To prevent the unlimited increase in the number of PFs caused by Equation (2) in Section 3.1, namely, \( K^{(j)}_n = K^{(j)}_n + M^{(j)}_n \), the number of PFs by \( p(\epsilon_{k,n}^{(j)} = 1 | z) \) is also used to prune some PFs. Specifically, PFs are preserved only when \( p(\epsilon_{k,n}^{(j)} = 1 | z) \) exceeds the predefined prune threshold \( P_{\text{pc}} \).

4.2. Joint Posterior Distribution and Factor Graph

The posterior PDFs \( f(x_n | z) \) and \( f(y_{k,n}^{(j)} | z) \) related in Equations (25–29) are marginal distributions of the joint posterior PDF \( f(x, y, h, q, m, \mathbf{A}_n | z) \), which involves the UE augmented state \( x \), PFs augmented states \( y \), DA vectors \( h \) and \( q \), the number of measurements \( m \) and the average number of clutters \( \mathbf{A}_n \) at all times. By running the BP message passing algorithm on the factor graph representing the factorization of the joint posterior PDF \( f(x, y, h, q, m, \mathbf{A}_n | z) \), effective estimations of the corresponding marginal posterior PDFs can be obtained. Based on the Bayesian theorem and related
independence hypotheses, using Equation (6), Equation (17), Equation (20) and Equation (21) and some simple derivations, the joint posterior PDF factorizes as

\[
f(x, y, h, q, m, \lambda_{CL}|z) 
\propto f(z|x, y, h, m) f(x, y, h, q, m, \lambda_{CL}) 
\]

\[
= f(x) \prod_{j=1}^{n} p(\lambda_{CL}^{(j)}|\lambda_{CL}^{(j-1)}) \prod_{m=1}^{M} c(x, \lambda_{CL}^{(j)}, \lambda_{CL}^{(j-1)}) 
\times \prod_{a=2}^{n} f(x_a|x_{a-1}) \prod_{j=1}^{j} p(\lambda_{CL}^{(j)}|\lambda_{CL}^{(j-1)}) \psi(h_a^{(j)}, q_a^{(j)}) 
\times \prod_{j=1}^{k'(0)} f(y_{j,n}^{(j)}) b(x_n, \lambda_{CL}^{(j)}, \lambda_{CL}^{(j-1)}; \lambda_{CL}^{(j)}, \lambda_{CL}^{(j-1)}; z_{j,n}^{(j)}) 
\times \prod_{m=1}^{M} c(x, \lambda_{CL}^{(j)}, \lambda_{CL}^{(j-1)}) 
\] (30)

This factorization is represented by the factor graph [34] in Figure 2, in which the variable node labeled \( x \) denotes the UE augmented state \( x_a = \left[ \begin{array}{c} x_n^T \\ \sigma_n^{-1} \end{array} \right] \), and the average number of clutter \( \lambda_{CL}^{(j)} \) is denoted by the independent variable node labeled \( \lambda_{CL} \).

**Figure 2.** Factor graph representing the factorization of the joint posterior probability density function (PDF) in (30). All factor nodes, variable nodes and messages related to the user equipment (UE) augmented state are represented in red color, those related to legacy potential features (PF) states are represented in blue color, and the section of the clutter intensity and the iterative data association (DA) are represented in orange color and green color, respectively. The abbreviations are defined as:

- \( K \triangleq K^{(j)}_a \), \( M \triangleq M^{(j)}_a \), \( x \triangleq x_a \), \( y_{j,n}^{(j)} \triangleq y_{j,n}^{(j)} \), \( h_{a}^{(j)} \triangleq h_{a}^{(j)} \), \( q_{a}^{(j)} \triangleq q_{a}^{(j)} \), \( \lambda_{CL} \triangleq \lambda_{CL}^{(j)} \), \( \phi_{a} \triangleq \phi(h_{a}^{(j)}, q_{a}^{(j)}) \), \( c_{a} \triangleq c(x, \lambda_{CL}^{(j)}, \lambda_{CL}^{(j-1)}) \), \( b_{a} \triangleq b(x_n, \lambda_{CL}^{(j)}, \lambda_{CL}^{(j-1)}; \lambda_{CL}^{(j)}, \lambda_{CL}^{(j-1)}; z_{j,n}^{(j)}) \), \( f \triangleq f(x_a|x_{a-1}) \), \( f_{a} \triangleq f(y_{j,n}^{(j)}) \), \( \alpha_{a} \triangleq \alpha(x_n) \), \( \alpha_{a} \triangleq \alpha(\lambda_{CL}^{(j)}) \), \( z_{a}^{-} \triangleq f(x_{a-1}) \), \( z_{a}^{+} \triangleq f(x_n) \), \( \beta_{a} \triangleq \beta(\lambda_{CL}^{(j)}) \), \( \chi_{a} \triangleq \chi(\lambda_{CL}^{(j)}) \), \( \zeta_{a} \triangleq \zeta(h_{a}^{(j)}, q_{a}^{(j)}) \), and \( \mu_{a} \triangleq \mu(\lambda_{CL}^{(j)}) \).
4.3. BP Message Passing Algorithm

Now, we calculate the marginal posterior PDFs \( f(x_k | z) \) and \( f(y_{k,a} | z) \) from Section 4.1 by running the BP message passing algorithm in the factor graph in Figure 2. Since the factor graph has loops, the computation sequence of each message must be specified. The order of the message calculation obeys the following rules: (1) the message in the current time is passed only to the next time; (2) the iterative message passing is conducted only in the DA section and executed once for each PT at all times; and (3) if an edge connects the variable nodes of the UE augmented state and new PF augmented states, the message can only be passed from the former to the latter. For the calculation equations of the message passing algorithm, we need to consider the BP message when PFs do not exist \( o(y_{k,a}) = o(a_{k,a}^{(i)}, e_{k,a}^{(i)}) \), \( e_{k,a}^{(i)} = 0 \). Similar to the “dummy PDF” introduced in Section 3.3, we define \( o(a_{k,a}^{(i)}, 0) = o(a_{k,a}^{(i)}) \) (note that this message is a single number, not a PDF). Combining the calculation order of messages and the computation method of BP messages, the steps of the BP message passing algorithm are described as follows:

1. Prediction.

First, the prediction message of the UE augmented state \( \alpha(x_k) = \alpha(\bar{x}_k, \sigma_k^2) \) is expressed as

\[
\alpha(\bar{x}_k, \sigma_k^2) = \sum_{\sigma_{k-1}^2 \in \mathbb{S}} \int f(\bar{x}_k, \sigma_k^2 | \bar{x}_{k-1}, \sigma_{k-1}^2) \tilde{f}(\bar{x}_{k-1}, \sigma_{k-1}^2) d\bar{x}_{k-1}
\]

(31)

Substituting (6) into (31), we have

\[
\alpha(\bar{x}_k, \sigma_k^2) = \sum_{\sigma_{k-1}^2 \in \mathbb{S}} \int f(\bar{x}_k | \sigma_k^2, \bar{x}_{k-1}) p(\sigma_k^2 | \sigma_{k-1}^2) \tilde{f}(\bar{x}_{k-1}, \sigma_{k-1}^2) d\bar{x}_{k-1}
\]

(32)

where \( \tilde{f}(\bar{x}_{k-1}, \sigma_{k-1}^2) \) is the calculated belief of the UE augmented state at time \( n-1 \). Then, the prediction message of legacy PF augmented states \( \bar{\alpha}_{k,a}^{(i)} = [a_{k,a}^{(i)} T, e_{k,a}^{(i)} T] \) is

\[
\bar{\alpha}_{k}^{(i)}(a_{k,a}^{(i)}, e_{k,a}^{(i)}) = \sum_{\alpha_{k-1}^{(i)}, e_{k-1}^{(i)} \in \mathbb{S}} \int f(a_{k,a}^{(i)}, e_{k,a}^{(i)} | a_{k-1,a}^{(i)}, e_{k-1,a}^{(i)}) \tilde{f}(a_{k-1,a}^{(i)}, e_{k-1,a}^{(i)}) da_{k-1,a}^{(i)}
\]

(33)

where the belief of the legacy PF augmented state \( \tilde{f}(a_{k,a}^{(i)}, e_{k,a}^{(i)}) \) is also calculated at time \( n-1 \). Substituting (7) and (8) into (33), when \( e_{k,a}^{(i)} = 1 \), we obtain

\[
\bar{\alpha}_{k,a}^{(i)}(a_{k,a}^{(i)}) = P_s \int f(a_{k,a}^{(i)} | a_{k-1,a}^{(i)}, e_{k-1,a}^{(i)}) \tilde{f}(a_{k-1,a}^{(i)}, 1) da_{k-1,a}^{(i)}
\]

(34)

while when \( e_{k,a}^{(i)} = 0 \),

\[
\bar{\alpha}_{k,a}^{(i)} = (1-P_s) \int f(a_{k,a}^{(i)}, 0) da_{k,a}^{(i)} + \tilde{f}(a_{k,a}^{(i)}, 0)
\]

(35)

where \( \tilde{f}(a_{k,a}^{(i)}, 0) \) and \( \tilde{f}(a_{k,a}^{(i)}, 0) \) are the values of the message of the average number of clutter, which is calculated as

\[
\chi(\lambda_{k,a}^{(i)}) = \sum_{\lambda_{k,a}^{(i)} \in \Xi} p(\lambda_{k,a}^{(i)}) \hat{\alpha}_{k,a}^{(i)}(\lambda_{k,a}^{(i)}) \hat{\alpha}_{k,a}^{(i)}(\lambda_{k,a}^{(i)})
\]

(36)

Similarly, \( \hat{\alpha}_{k,a}^{(i)}(\lambda_{k,a}^{(i)}) \) denotes the belief of the average number of clutter at time \( n-1 \). After the prediction step, the computation of other messages will be performed with all PTs \( j \in \{1, \ldots, J\} \), all legacy PFs \( k \in K_{n-1}^{(i)} \) and all new PFs \( m \in M_{n-1}^{(i)} \) in parallel.

2. Measurement evaluation of legacy PFs.
In Figure 2, the message $\varepsilon(h_{k,n}^{(j)})$ passed to the variable node representing the feature-oriented DA vector $h_{k,n}^{(j)}$ is calculated as

$$
\varepsilon(h_{k,n}^{(j)}) = \sum_{\lambda_{k,n}^{(j)}} \sum_{\sigma_{k,n}^{(j)}} \sum_{\lambda_{k,n}^{(j)}} \sum_{\sigma_{k,n}^{(j)}} \int \chi(h_{k,n}^{(j)}) \alpha_{k,n}^{(j)}(x_{k,n}^{(j)})dx_{k,n}^{(j)} + 1(h_{k,n}^{(j)})\alpha_{k,n}^{(j)}
$$

(37)

3. Measurement evaluation of new PFs.

The calculation of the message $\xi(q_{m,n}^{(j)})$ passed to the variable node representing the measurement-oriented DA vector $q_{m,n}^{(j)}$ is

$$
\xi(q_{m,n}^{(j)}) = \sum_{\lambda_{m,n}^{(j)}} \sum_{\sigma_{m,n}^{(j)}} \sum_{\lambda_{m,n}^{(j)}} \sum_{\sigma_{m,n}^{(j)}} \int \chi(h_{m,n}^{(j)}) \alpha_{m,n}^{(j)}(x_{m,n}^{(j)})dx_{m,n}^{(j)} + 1(h_{m,n}^{(j)})\alpha_{m,n}^{(j)}
$$

(38)

Substituting (24) into (38), $\xi(q_{m,n}^{(j)})$ can be expressed as

$$
\xi(q_{m,n}^{(j)}) = \begin{cases} 1, & q_{m,n}^{(j)} \in S_{m,n}^{(j)} \\ 1 + \sum_{\lambda_{m,n}^{(j)}} \sum_{\sigma_{m,n}^{(j)}} \sum_{\lambda_{m,n}^{(j)}} \sum_{\sigma_{m,n}^{(j)}} \int \chi(h_{m,n}^{(j)}) \alpha_{m,n}^{(j)}(x_{m,n}^{(j)})dx_{m,n}^{(j)} & q_{m,n}^{(j)} = 0 \\ \alpha(x_{m,n}^{(j)}, \sigma_{m,n}^{(j)},f_{\text{new}},(\bar{a}_{m,n}^{(j)}))f(\bar{a}_{m,n}^{(j)})dx_{m,n}^{(j)} \end{cases}
$$

(39)

4. Iterative DA

Next, using $\varepsilon(h_{k,n}^{(j)})$ and $\xi(q_{m,n}^{(j)})$, the messages $\gamma(h_{k,n}^{(j)})$ and $\nu(q_{m,n}^{(j)})$ can be calculated by the iterative BP method. First, for each measurement $m \in \mathcal{M}_{k,n}^{(j)}$, the iterative calculations of the messages $\delta_{m,n}^{(i)}(h_{k,n}^{(j)})$ and $\zeta_{k,n}^{(i)}(q_{m,n}^{(j)})$ are [41]

$$
\delta_{m,n}^{(i)}(h_{k,n}^{(j)}) = \sum_{q_{m,n}^{(j)}} \xi(q_{m,n}^{(j)}) \phi(h_{k,n}^{(j)}, q_{m,n}^{(j)}) \prod_{k \in \mathcal{K}_{m,n}^{(j)} \setminus \{i\}} \zeta_{k,n}^{(i-1)}(q_{m,n}^{(j)})
$$

(40)

$$
\zeta_{k,n}^{(i)}(q_{m,n}^{(j)}) = \sum_{h_{k,n}^{(j)}} \varepsilon(h_{k,n}^{(j)}) \phi(h_{k,n}^{(j)}, q_{m,n}^{(j)}) \prod_{m \in \mathcal{M}_{k,n}^{(j)} \setminus \{i\}} \delta_{m,n}^{(i)}(h_{k,n}^{(j)})
$$

(41)

where the number of iterations is $t = 1, \ldots, T$. When $i = 0$, the initial value of the iterations in (40) and (41) is

$$
\zeta_{k,n}^{(0)}(q_{m,n}^{(j)}) = \sum_{h_{k,n}^{(j)}} \varepsilon(h_{k,n}^{(j)}) \phi(h_{k,n}^{(j)}, q_{m,n}^{(j)})
$$

(42)

Then, after the last iteration, the messages $\gamma(h_{k,n}^{(j)})$ and $\nu(q_{m,n}^{(j)})$ are expressed as

$$
\gamma(h_{k,n}^{(j)}) = \prod_{m=1}^{M_{k,n}^{(j)}} \delta_{m,n}^{(T)}(h_{k,n}^{(j)})
$$

(43)

$$
\nu(q_{m,n}^{(j)}) = \prod_{k=1}^{K_{m,n}^{(j)}} \zeta_{k,n}^{(T)}(q_{m,n}^{(j)})
$$

(44)

According to $\gamma(h_{k,n}^{(j)})$, $\chi(h_{k,n}^{(j)})$ and $\delta_{k,n}^{(j)}$, the message $h_{k,n}^{(j)}(x_{k,n})$ related to the UE is
\[ \eta_k^{(j)}(x_n) = \sum_{h_{k,n}^{(j)}=0}^{M_k^{(j)}} \sum_{\lambda_{CL,n}^{(j)} \in \mathcal{X}} \gamma(h_{k,n}^{(j)}) \mathcal{X}(\lambda_{CL,n}^{(j)}) \int b(x_n, a_{k,n}^{(j)}, h_{k,n}^{(j)}, \lambda_{CL,n}^{(j)}, \varepsilon_{k,n}^{(j)}) \alpha(\hat{x}_n, \sigma_n^2) d\hat{x}_n \]  

(45)

5. Measurement update for legacy PFs

Similar to the measurement update process for the UE, the message about legacy PFs can be expressed as

\[ \eta(a_{m,n}^{(j)}, 0) = \gamma(h_{k,n}^{(j)} = 0) \]

(46)

6. Measurement update for new PFs

The message \( \mu(\bar{a}_{m,n}^{(j)}, \bar{z}_{m,n}^{(j)}) \) related to new PFs is calculated as

\[ \mu(m, 1) = \nu(a_{m,n}^{(j)}) \mathcal{X}(\lambda_{CL,n}^{(j)}) \int c(x_n, \bar{a}_{m,n}^{(j)}, 1, 0, \lambda_{CL,n}^{(j)}, \varepsilon_{m,n}^{(j)}) \alpha(\hat{x}_n, \sigma_n^2) d\hat{x}_n \]

(48)

\[ \mu(m, 0) = \mu(\bar{a}_{m,n}^{(j)}, 0) = \sum_{\eta_{k,n}^{(j)} = 0}^{K_{k,n}} \nu(\eta_{k,n}^{(j)}) \]

(49)

7. Calculation of beliefs

After calculating the messages above, we can obtain the beliefs for approximating marginal posterior PDFs. First, up to a constant factor, the belief of the UE augmented state \( f(x_n) = f(\hat{x}_n, \sigma_n^2) \) is expressed as

\[ \tilde{f}(\hat{x}_n, \sigma_n^2) \propto \alpha(\hat{x}_n, \sigma_n^2) \prod_{j=1}^{J} \prod_{k = 1}^{L} \eta_k^{(j)}(x_n) \]

(50)

where the belief \( \tilde{f}(\hat{x}_n, \sigma_n^2) \) provides an approximate estimation of the marginal posterior PDF of the UE augmented state and is used to substitute \( f(\hat{x}_n, \sigma_n^2 | z) \) in (26). Furthermore, the belief \( \tilde{f}(a_{k,n}^{(j)}, z_{k,n}^{(j)}) \) of the legacy PF augmented states \( \bar{X}_{k,n}^{(j)} = \begin{bmatrix} a_{k,n}^{(j)} \bar{z}_{k,n}^{(j)} \end{bmatrix}^T \) is calculated as

\[ \tilde{f}(a_{k,n}^{(j)}, 1) \propto \alpha_k(a_{k,n}^{(j)}, 1) \eta_k^{(j)}(a_{k,n}^{(j)}) \]

(51)

and \( \tilde{f}(a_{k,n}^{(j)}, 0) \propto \alpha_k(a_{k,n}^{(j)}, 0) \). The belief \( \tilde{f}(\bar{a}_{m,n}^{(j)}, \bar{z}_{m,n}^{(j)}) \) of new PF augmented states \( \bar{X}_{m,n}^{(j)} = \begin{bmatrix} \bar{a}_{m,n}^{(j)} \bar{z}_{m,n}^{(j)} \end{bmatrix}^T \) is

\[ \tilde{f}(\bar{a}_{m,n}^{(j)}, 1) \propto \mu(\bar{a}_{m,n}^{(j)}, 1) \]

(52)

and \( \tilde{f}(\bar{a}_{m,n}^{(j)}, 0) \propto \mu(\bar{a}_{m,n}^{(j)}, 0) \). Then, \( \tilde{f}(a_{k,n}^{(j)}, 1) \) and \( \tilde{f}(\bar{a}_{m,n}^{(j)}, 1) \) are leveraged to approximate the marginal posterior PDF of PF augmented states \( f(a_{k,n}^{(j)}, \varepsilon_{k,n}^{(j)} = 1 | z) \) in (27) jointly, where \( k' \in \mathcal{K}_{k,n} \cup \mathcal{M}_{m,n}^{(j)} \).

Next, the belief \( \tilde{p}(\lambda_{CL,n}^{(j)} | z) \) used to approximate the marginal posterior PMF of the mean number of clutter \( p(\lambda_{CL,n}^{(j)} | z) \) is expressed as
\begin{equation}
\hat{\rho}(\lambda_{\text{CL}}^{(j)}) = \chi(\lambda_{\text{CL}}^{(j)}) \beta(\lambda_{\text{CL}}^{(j)})
\end{equation}

where

\begin{equation}
\beta(\lambda_{\text{CL}}^{(j)}) = \sum_{h_{k,n}} \sum_{a_{k,n}} \gamma(h_{k,n}^{(j)}) \int h(x_n, a_{k,n}, 1, h_{k,n}^{(j)}, \lambda_{\text{CL}}^{(j)}; z_{n}^{(j)}) \times \alpha_t(a_{k,n}^{(j)}, 1) \alpha(\tilde{x}_n, \sigma^2) \text{d}a_{k,n}^{(j)} \text{d}\tilde{x}_n + \gamma(h_{k,n}^{(j)} = 0) \alpha_{k,n}^{(j)}
\end{equation}

However, like other nonlinear non-Gaussian filtering models, there are no closed-form solutions for the messages and beliefs above; that is, calculating the beliefs of the UE and PFs directly is infeasible. Therefore, a particle filter-based implementation is used to approximate the beliefs of the UE and PFs (refer to [38,42] for details).

5. Simulation Results and Discussions

This section will evaluate the proposed algorithm from three scenarios and compare it with the conventional BP-SLAM algorithm with unknown parameters and the ideal BP-SLAM algorithm with known parameters.

5.1. Simulation Parameters

We use the environment map shown in Figure 1 as the reference scenario of our simulation. The UE is assumed to follow a nearly constant velocity model, the evolution PDF \( f(x_n | x_{n-1}) \) of which is \( x_n = Ax_{n-1} + Bw_n \), where \( A \in \mathbb{R}^{4 \times 4} \) and \( B \in \mathbb{R}^{2 \times 2} \) are defined in [43] (assuming that the sampling period is \( \Delta T = 1 \) s). The process noise \( w_n \) is zero-mean and Gaussian with covariance matrix \( \sigma^2 P \), where \( P \) denotes the identity matrix of size \( 2 \times 2 \). Thus, the difference of the UE dynamic models is determined by the UE process noise variance \( \sigma^2 P \). \( \sigma^2 P \) describes the average increment of the UE speed within the sampling period \( \Delta T \), and \( \sigma^2 P \) with a high value is especially used to model the acceleration and turning process of the UE. Therefore, to accurately describe the motion state of the UE, we propose to use two dynamic models with different process noise variances to represent the motion of the UE. These two process noise variances are expressed as \( \sigma^2 P_1 \) and \( \sigma^2 P_2 \), where \( \sigma^2 P_1 \) can be chosen as a rule of thumb and \( \sigma^2 P_2 \) is at least one order of magnitude larger than \( \sigma^2 P_1 \) to describe the state mutation of the UE.

Since PFs are static, the evolution model can be considered to be \( f(a_{k,n}^{(j)} | a_{k,n-1}^{(j)}) = \delta(a_{k,n}^{(j)} - a_{k,n-1}^{(j)}) \), where \( \delta(\cdot) \) is the Dirac delta function. However, for the numerical stability of our algorithm, we introduce a small process noise into the evolution model of PFs, i.e., \( a_{k,n}^{(j)} = a_{k,n-1}^{(j)} + \alpha_{k,n}^{(j)} \), where \( \alpha_{k,n}^{(j)} \) is the zero-mean Gaussian white noise with covariance matrix \( \sigma^2_a I_2 \).

Different from most SLAM algorithms, our measurement model is only based on range measurements of MPCs in the radio channel, and the range measurement \( z_{m,n}^{(j)} \) is expressed as

\begin{equation}
z_{m,n}^{(j)} = ||u_m . a_{k,n}^{(j)}|| + \epsilon_{m,n}^{(j)}
\end{equation}

where the measurement noise \( \epsilon_{m,n}^{(j)} \) is independent and identically distributed with \( m, n \) and \( j \) and is a zero-mean Gaussian distribution with variance \( \nu_{m,n}^{(j)} \). We assume that the standard deviation of the measurement noise is \( \nu_{m,n}^{(j)} = 0.1 \). This measurement model (55) determines the factors \( f(z_{m,n}^{(j)} | \tilde{x}_n, \sigma_{m,n}^{(j)}) \) and \( f(z_{m,n}^{(j)} | \tilde{x}_n, a_{k,n}^{(j)}) \) in the likelihood function (18). In each simulation run, noisy range measurements are produced according to (55).
The realization of (55) is based on the UE position \( u_s \) in its trajectory, the fixed PT positions \( a_i^{(1)} \) and \( a_i^{(2)} \), and the fixed VT positions \( a_j^{(1)} \), \( i = 2, \ldots, L_u \), \( j = 1,2 \) in Figure 1, where \( L_u^{(1)} = 6 \) and \( L_u^{(2)} = 5 \) are the number of features (including the PT and the associated VTs).

In our proposed algorithm, clutter is modeled as Poisson distribution with mean value \( \lambda_{cl,a}^{(j)} \). We assume that the spatial PDF of clutter is uniform in the region of interest (ROI), which is a circular area with a radius of 30 m around the center of the environment map in Figure 1. Therefore, the clutter intensity can be uniquely determined by the mean value \( \lambda_{cl,a}^{(j)} \). We model \( \lambda_{cl,a}^{(j)} \) as a discrete random variable taking values from the finite set \( \mathcal{G} = \{0.1,1,2,\ldots,9\} \), that is, \( \lambda_{cl,a}^{(j)} \) has \( N_s = 10 \) possible values. The corresponding transition matrix \( G^{(j)} = G \) is chosen as [33]: when \( i = 2,3,\ldots,9 \), there are \( G_{2,1} = 0.02 \), \( G_{3,2} = 0.01 \), \( G_{4,3} = 0.01 \) and \( G_{9,1} = 0.03 \); furthermore, \( G_{2,2} = 0.99 \), \( G_{3,3} = 0.99 \). The PDF \( f_{\Delta_{\text{n},a}}(\Delta_{\text{n},a}^{(j)}) \) and mean value \( \lambda_{\text{n},a}^{(j)} \) of the newly detected features are estimated online by the PHD filter, which uses the spatial PDF \( f_{\Delta_{\text{n},a}}(\Delta_{\text{n},a}^{(j)}) \) and mean number \( \lambda_{\text{n}}^{(j)} \) of newborn features [44]. We assume that \( f_{\Delta_{\text{n},a}}(\Delta_{\text{n},a}^{(j)}) \) is uniform in the ROI. When \( n = 1 \), the initialization of newly detected features is realized by setting \( \lambda_{\text{n},a}^{(j)} = 6 \) and the initial PDF \( f_{\Delta_{\text{n},a}}(\Delta_{\text{n},a}^{(j)}) \) to be uniform in the ROI. In addition, the detection probability \( P_d^{(j)}(x_s,y_s^{(j)}) \) is considered fixed, i.e., \( P_d^{(j)}(x_s,y_s^{(j)}) = P_2 \).

The messages and beliefs in Section 4.3 are approximately calculated via the particle filter-based implementation with \( N_{\text{par}} = 10000 \) particles in the proposed algorithm. Particles representing the initial state of the PT follow the two-dimensional Gaussian distribution \( \mathcal{N}(a_i^{(j)},\sigma_a^2 I_3) \), where \( a_i^{(j)} \) denotes the initial positions of PT1 \( (j = 1) \) and PT2 \( (j = 2) \). Particles representing the initial state of the UE follow a 4-dimensional uniform distribution with the center \( x_i = [p_1^T \ 0 \ 0]^T \), where \( x_1 \) is the starting position of the UE. This implies that the prior information of the initial state of the PT and the UE is known, but the prior information related to VTs is unknown. Finally, the termination condition of the iterative DA is \( \left\| \sum_{l=1}^{I_{\text{max}}} \sum_{n=1}^{N_{\text{par}}} \left( \delta_{\text{n},a}^{(j)}(h_{\text{n},a}^{(j)}) - \delta_{\text{n},a}^{(j)}(h_{\text{n},a}^{(j)}) \right) \right\|^2 < 10^{-7} \) or the maximum number of iterations \( I_{\text{max}} = 1000 \) is reached. Next, we will divide the scenarios into groups to evaluate the performance of our proposed algorithm and provide the corresponding numerical results. The common simulation parameters of all scenarios are shown in Table 1. The parameter settings in Table 1 are based on [25] and it is verified in [25] that the BP-SLAM algorithm can achieve excellent performance under these conditions.

| Parameter                      | Value       |
|-------------------------------|-------------|
| \( \sigma_a \)                | \( 10^{-3} \) m |
| \( \sigma_{b,1} \)            | \( 0.1 \) m  |
| \( \lambda_{\text{n},a}^{(j)} \) | \( 10^{-4} \) |
| \( \nu_{\text{n},a}^{(j)} \)   | \( 0.999 \)  |
| \( P_d^{(j)} \)               | \( 0.5 \)    |
| \( P_{\text{ib}} \)           | \( 10^{-4} \) |
| \( P_{\text{pr}} \)           | \( 10,000 \) |
| \( N_{\text{par}} \)          | \( 100 \)    |

5.2. First Scenario: Unknown Process Noise Only

In the first scenario, the UE switches between two nearly constant velocity models with different process noise variances, \( \sigma_{p_i}^2 = 0.1^2 \) and \( \sigma_{p_i}^2 = 0.1^2 \). The simulation time is 800 s. For 0–720 s, the UE process noise variance is \( \sigma_{p_i}^2 = 0.1^2 \); then, the UE accelerates suddenly with the process noise variance changing to \( \sigma_{p_i}^2 = 0.1^2 \) for 720–730 s; finally, for 730–800 s, the UE process noise variance is the same as that for the first stage.

---

Remote Sens. 2021, 13, 1625

18 of 28
transition matrix of the UE process noise variance is $[Q_{x}]_{1,1} = [Q_{x}]_{2,2} = 0.9975$ and $[Q_{x}]_{1,2} = [Q_{x}]_{2,1} = 0.0025$. In this scenario, the clutter intensity is considered to be a fixed prior. We assume that the mean number of clutter is $A_{t,s}^{(i)}=0.1$. We compare our proposed algorithm with the conventional BP-SLAM algorithm [25] with unknown process noise variance and the ideal BP-SLAM algorithm with known process noise variance, which can obtain the prior information of the UE process noise variance at each time in advance. The ideal BP-SLAM algorithm is impossible in practice but can be used as a benchmark to evaluate the performance of our proposed algorithm. In the conventional BP-SLAM algorithm with unknown process noise variance, we considered two different parameter settings labeled BP-SLAM A and BP-SLAM B. The UE process noise variances in BP-SLAM A and BP-SLAM B both remain unchanged, but their settings are different, i.e., $\sigma_p^2 = \sigma_n^2 = 0.01^2$ for BP-SLAM A and $\sigma_p^2 = \sigma_n^2 = 0.1^2$ for BP-SLAM B.

Figure 3 shows the performance of the proposed algorithm in the first scenario, which is obtained based on the average of 100 simulation runs. Figure 3a,b depict the mean optimal subpattern assignment (MOSPA) [45] errors for PT1 and PT2 and their associated VTs, respectively. The MOSPA error is based on Euclidean distance with a cutoff parameter of 5 m and order 1. The MOSPA error combines the estimation errors of correctly detected PFs and incorrectly detected PFs, which is often used as the standard to evaluate the mapping accuracy of multipath-assisted SLAM algorithms. In Figure 3a, period $n \in [0,720]$, the MOSPA error of BP-SLAM B for PT1 and its associated VTs is always higher than the other three algorithms. Then, when the UE process noise variance suddenly changes, i.e., $n \in [720,730]$, the MOSPA error of BP-SLAM A rises sharply with a maximum of more than 3 m. However, the MOSPA error of the proposed algorithm is always smaller than that of BP-SLAM A and BP-SLAM B in this period. This relationship still exists even after the UE returns to the normal motion state, i.e., $n \in [730,800]$. The MOSPA error for PT2 and its associated VTs shown in Figure 3b also have similar performance. Figure 3c,d describe the average number of detected PFs associated with PT1 and PT2, respectively. From these two figures, we can see that the average number of detected PFs for the proposed algorithm is always close to the actual number of features in the environment, namely, $L_{x}^{(1)} = 6$ and $L_{x}^{(2)} = 5$. However, the average number of detected PFs for BP-SLAM A deviates from the truth after 720 s, i.e., the UE process noise variance changes suddenly. The average number of detected PFs for BP-SLAM B deviates from the truth even when the UE process noise remains unchanged, i.e., $n \in [0,720]$. Figure 3e depicts the root mean square errors (RMSEs) of the UE position for the four algorithms. Figure 3e shows that in period $n \in [720,730]$, the UE position RMSE for BP-SLAM A rise sharply with a maximum of more than 1.2 m, while the UE position RMSE of the proposed algorithm is always less than 0.4 m in this period, which is close to the ideal BP-SLAM algorithm with known parameters. Although the UE position RMSE for BP-SLAM B has no sharp rise in this period, it is higher than that of the proposed algorithm all the time.
Table 2 summarizes the error statistics of different algorithms in the first scenario within 800 s, including the average UE position RMSE, the average MOSPA error and the average error for the estimated number of detected PFs. The latter two errors are based on the average of PT1 and PT2 statistics. From Table 2, we can see that the statistical error of the proposed algorithm is significantly smaller than that of BP-SLAM A and BP-SLAM B, and close to the ideal BP-SLAM algorithm with known parameters. Therefore, Figure 3 and Table 2 both illustrate that the proposed algorithm is robust to the uncertainty of the process noise.
Table 2. Error statistics of different algorithms in the first scenario.

| Algorithm       | UE RMSE | MOSPA  | No. of PFs |
|-----------------|---------|--------|------------|
| BP-SLAM A       | 0.2891 m| 1.1143 m| 0.1090     |
| BP-SLAM B       | 0.1984 m| 1.0442 m| 0.2564     |
| Proposed algorithm | 0.1605 m| 0.8759 m| 0.0005     |
| BP-SLAM (known params) | 0.0949 m| 0.7739 m| 0 (True)   |

5.3. Second Scenario: Unknown Clutter Intensity Only

In the second scenario, we assume that only the clutter intensity parameter is unknown, which means that the UE does not have a sudden change in the dynamic model, with the process noise variance keeping $\sigma^2_p=0.01^2$ during the movement. Therefore, it is unnecessary to consider the uncertainty of process noise in this scenario. The simulation time is also 800 s. Clutter is generated by a Poisson distribution with mean number $\lambda_{cl,a}^{(j)}=6$, and the spatial PDF of clutter is uniform in the ROI, i.e., $f_{cl}(z_{m,a}^{(j)})=1/D$, where $D$ is the area of the ROI. Thus, the clutter intensity can be expressed as $6/D$. We also compare the proposed algorithm with the conventional BP-SLAM algorithm with unknown parameters and the ideal BP-SLAM algorithm with known parameters in this scenario. Similar to the first scenario, we consider two parameter settings in the conventional BP-SLAM algorithm labeled BP-SLAM A and BP-SLAM B. They have no prior information on the true clutter intensity and just set this parameter by experience. BP-SLAM A and BP-SLAM B initialize the clutter intensity to $0.1/D$ and $12/D$, respectively. The ideal BP-SLAM algorithm with known parameters can obtain the prior information of the true clutter intensity in advance, which can also be used as the benchmark to evaluate the performance of our proposed algorithm in this scenario.

Figure 4 illustrates the simulation results of the proposed algorithm in the second scenario, which are also obtained based on the average of 100 simulation runs. Figure 4a,b show the MOSPA errors for PT1 and PT2 and their associated VTs, respectively. From Figure 4a, we see that the MOSPA errors of the four algorithms for PT1 all have a decreasing trend and finally reach steady states. The MOSPA errors for PT1 and the associated VTs of BP-SLAM A and BP-SLAM B converge at approximately 0.6 m and 0.4 m, respectively, while the performance of the proposed algorithm is better, converging near 0.2 m, which is close to the result of the BP-SLAM algorithm with known parameters. The results in Figure 4b are similar to those of Figure 4a. When the MOSPA error of the four algorithms becomes steady, the MOSPA error of the proposed algorithm is significantly smaller than that of BP-SLAM A and BP-SLAM B. Figure 4c,d describe the average number of detected PFs associated with PT1 and PT2 every 60 s, respectively. These two figures show that the average numbers of detected PFs by the proposed algorithm converge to the true values within 60 s, while the numbers of detected PFs by BP-SLAM A are always in a disordered state. Meanwhile, although the results for BP-SLAM B are convergent, they converge to the wrong number. Figure 4e depicts the UE position RMSEs for the four algorithms. Figure 4e shows that the position accuracy of the proposed algorithm is significantly better than that of BP-SLAM A and BP-SLAM B. During the simulation run time of 800 s, the RMSE of the proposed algorithm is always lower than 0.15 m, which is close to the performance of the BP-SLAM algorithm with known parameters. However, the UE position RMSE of BP-SLAM A increases rapidly in 800 s, and the maximum is more than 0.65 m. Although the performance of BP-SLAM B is relatively good, its position accuracy is lower than that of the proposed algorithm all the time.
Figure 4. Results for the second scenario: (a) MOSPA error for PT1 and the associated VTs; (b) MOSPA error for PT2 and the associated VTs; (c) average number of detected PFs associated with PT1 every 60 s; (d) average number of detected PFs associated with PT2 every 60 s; and (e) UE position RMSE.

Table 3 summarizes the error statistics of different algorithms in the second scenario within 800 s. The MOSPA error and the error for the estimated number of detected PFs are also based on the average of PT1 and PT2 statistics. It can be seen from Table 3 that the statistical error of the proposed algorithm is significantly smaller than that of BP-SLAM A and BP-SLAM B, and close to the ideal BP-SLAM algorithm with known parameters. Therefore, Figure 4 and Table 3 both verify that the proposed algorithm is robust to the uncertainty of the clutter intensity.
Table 3. Error statistics of different algorithms in the second scenario.

| Algorithm            | UE RMSE | MOSPA   | No. of PFs |
|----------------------|---------|---------|------------|
| BP-SLAM A            | 0.3768 m| 1.2199 m| 8.0151     |
| BP-SLAM B            | 0.1083 m| 1.0006 m| 0.8329     |
| Proposed algorithm   | 0.0741 m| 0.7659 m| 0.0678     |
| BP-SLAM (known params)| 0.0714 m| 0.7376 m| 0 (True)   |

5.4. Third Scenario: Unknown Process Noise and Clutter Intensity

In the third scenario, the uncertainty of the process noise and the clutter intensity are considered together, and the simulation time is 800 s. The variation in the UE process noise variance is the same as that in the first scenario. Most of the time, the UE process noise variance is \( \sigma_p^2 = 0.01^2 \), but when \( n \in [720, 730] \), due to the sudden acceleration, the UE process noise variance changes to \( \sigma_p^2 = 0.1^2 \). The BP-SLAM algorithm with unknown parameters considers that the UE process noise variance is fixed to \( \sigma_p^2 = \sigma_n^2 = 0.01^2 \), while our proposed algorithm can adaptively switch between the two dynamic models with the transition matrix of the process noise variance the same as that in the first scenario. The generation mechanism of clutter is consistent with the second scenario. The true clutter intensity is \( 4/D \), but the proposed algorithm and the BP-SLAM algorithm with unknown parameters do not have this prior information. We also consider two parameter settings in the conventional BP-SLAM algorithm with unknown parameters in this scenario, labeled BP-SLAM A and BP-SLAM B. In BP-SLAM A, the UE process noise variance is fixed to \( \sigma_p^2 = \sigma_n^2 = 0.01^2 \) and the clutter intensity is initialized to \( 0.11/D \) by experience. In BP-SLAM B, these two parameters are set to \( \sigma_p^2 = \sigma_n^2 = 0.1^2 \) and \( 8/D \), respectively. In the third scenario, the BP-SLAM algorithm with known parameters can obtain the prior parameters of the process noise and the clutter intensity at each time in advance, which can still be used as the benchmark to evaluate the performance of the proposed algorithm.

Figure 5 depicts the performance of the proposed algorithm in the third scenario, which is also obtained based on the average of 100 simulation runs. Figure 5a,b describe the MOSPA errors for PT1 and PT2 and their associated VTs, respectively. Figure 5a shows that the MOSPA errors of the four algorithms all reach steady states within approximately 150 s, and BP-SLAM A and BP-SLAM B are stable at approximately 1.8 m and 0.5 m, respectively. However, the performance of the proposed algorithm and the BP-SLAM algorithm with known parameters is obviously better with both below 0.3 m. In particular, the MOSPA error of BP-SLAM A rises sharply during the sudden change of the UE process noise variance, i.e., \( n \in [720, 730] \), and the maximum arrives near 3.2 m. The MOSPA errors of the proposed algorithm and the BP-SLAM algorithm with known parameters do not change significantly in this period. The MOSPA error for PT2 and the associated VTs described in Figure 5b has a similar performance to that in Figure 5a, i.e., regardless of whether the UE process noise changes, the performance of the proposed algorithm is better than that of BP-SLAM A and BP-SLAM B. Figure 5c,d show the average number of detected PFs associated with PT1 and PT2 every 60 s, respectively. These two figures show that the average numbers of detected PFs by the proposed algorithm are always near the true value, while the results of BP-SLAM A are not convergent. Although the results for BP-SLAM B are convergent, they converge on the wrong number. Therefore, in the third scenario, the estimation performance of the proposed algorithm for the average number of detected PFs is obviously better than that of BP-SLAM A and BP-SLAM B. Figure 5e describes the UE position RMSEs for the four algorithms in the third scenario. The time average RMSE of the proposed algorithm is 0.11 m within 800 s, while that of BP-SLAM A and BP-SLAM B are 1.46 m and 0.25 m, respectively. In particular, the time average RMSE of the proposed algorithm is 0.37 m when the UE process noise variance changes suddenly, i.e., \( n \in [720, 730] \), while BP-SLAM A and BP-SLAM B are 3.18
m and 0.55 m in this period. For the BP-SLAM algorithm with known parameters, the time average RMSE of the UE position is 0.07 m during 800 s and 0.11 m when $n \in [720, 730]$. Based on the above data, the UE position accuracy of the proposed algorithm is significantly better than that of the BP-SLAM algorithm with unknown parameters and close to that of the BP-SLAM algorithm with known parameters.

![Graphs](image1.png) ![Graphs](image2.png) ![Graphs](image3.png) ![Graphs](image4.png) ![Graphs](image5.png)

**Figure 5.** Results for the third scenario: (a) MOSPA error for PT1 and the associated VTs; (b) MOSPA error for PT2 and the associated VTs; (c) average number of detected PFs associated with PT1 every 60 s; (d) average number of detected PFs associated with PT2 every 60 s; and (e) UE position RMSE.

Table 4 summarizes the error statistics of different algorithms in the third scenario within 800 s. The MOSPA error and the error for the estimated number of detected PFs are also based on the average of PT1 and PT2 statistics. From Table 4, we can see that the
statistical error of the proposed algorithm is significantly smaller than that of BP-SLAM A and BP-SLAM B, and close to the ideal BP-SLAM algorithm with known parameters. Therefore, Figure 5 and Table 4 show that the proposed algorithm is robust to both the clutter intensity and the process noise uncertainty.

Table 4. Error statistics of different algorithms in the third scenario.

| Algorithm                  | UE RMSE  | MOSPA   | No. of PFs |
|----------------------------|----------|---------|------------|
| BP-SLAM A                  | 1.4636 m | 2.2396 m| 8.4696     |
| BP-SLAM B                  | 0.2452 m | 1.0713 m| 0.8099     |
| Proposed algorithm         | 0.1121 m | 0.7659 m| 0.0628     |
| BP-SLAM (known params)     | 0.0699 m | 0.6738 m| 0 (True)   |

5.5. Calculation Complexity Analysis

The computational complexity of the conventional BP-SLAM algorithm is proportional to $O(N_{\text{par}})$, where $N_{\text{par}}$ is the number of particles [25]. The proposed algorithm adds the evolutions of the process noise variance and the clutter intensity to the Bayesian model compared with the conventional BP-SLAM algorithm; hence, its computational complexity is proportional to $O(N_{\text{par}} + N_{p}^{2} + N_{g}^{2})$ [33], where $N_{p}$ and $N_{g}$ denote the number of process noise variances and clutter intensity, respectively. Considering that $N_{p}$ and $N_{g}$ are generally small compared to the number of particles $N_{\text{par}}$, and therefore our proposed algorithm has nearly the same computational complexity as the conventional BP-SLAM algorithm. The average CPU run time consumed by each time step $n$ in the third scenario is also used to compare the calculation complexity of different algorithms, where $N_{\text{par}} = 10000$, $N_{p} = 2$, and $N_{g} = 10$. The results of CPU run-time are based on 100 simulation runs and 800 time steps using a MATLAB implementation on an Intel Core i7-3632QM CPU@2.20 GHz. Table 5 shows the complexity comparison of different algorithms in terms of both the complexity orders and the CPU run times. From Table 5, we can see that the proposed algorithm has almost the same calculation complexity as the conventional BP-SLAM algorithm with unknown parameters, but the performance of positioning and mapping accuracy is much better, close to the ideal BP-SLAM algorithm with known parameters. The reason is that the proposed algorithm considers the influence of the process noise and the clutter intensity uncertainty.

Table 5. Complexity comparison of different algorithms.

| Algorithm                  | Complexity Order | CPU Run Time |
|----------------------------|------------------|--------------|
| BP-SLAM (unknown params)   | $O(N_{\text{par}})$ | 0.0645 s     |
| Proposed algorithm         | $O(N_{\text{par}} + N_{p}^{2} + N_{g}^{2})$ | 0.0761 s     |
| BP-SLAM (known params)     | $O(N_{\text{par}})$ | 0.0652 s     |

6. Conclusions

In multipath-assisted SLAM using radio signals, the estimation of the process noise and the clutter intensity is difficult in practice. Therefore, it is highly significant to study the multipath-assisted SLAM algorithm with the ability to adaptively estimate these unknown parameters. A robust multipath-assisted SLAM algorithm is proposed in this paper to solve the problem that the accuracy of the multipath-assisted SLAM algorithm decreases when the process noise and the clutter intensity are statistically uncertain. The proposed algorithm can still work with unknown time-varying parameters by continuously inferring an unknown parameter model, the states of the UE and environmental features. Specifically, we integrate the process noise variance and the
clutter intensity into the factor graph representing the statistical structure of the SLAM problem and use the Markov chain model to describe the evolution of these unknown parameters. Finally, the BP message passing algorithm is performed to calculate the marginal posterior PDFs of each parameter to achieve the goal of simultaneous localization and mapping and adaptively learning the process noise variance and the clutter intensity.

The simulation results demonstrate that the proposed algorithm is superior to the conventional BP-SLAM algorithm in positioning and mapping accuracy with the uncertainty of the process noise and the clutter intensity. In particular, when the process noise and the clutter intensity are both unknown, the UE position time average RMSE at all times of the proposed algorithm is significantly less than that of the conventional BP-SLAM algorithm (0.11 m versus 1.46 m), and the performance in the MOSPA error and the average number of detected PFs is also obviously better than that of the conventional BP-SLAM algorithm (0.77 m versus 2.24 m and 0.06 versus 8.5, respectively), taking BP-SLAM A as an example. Therefore, the proposed algorithm can effectively restrain the impact of the process noise and the clutter intensity uncertainty on SLAM estimation accuracy and improve the robustness of a multipath-assisted SLAM system using radio signals.

We assume time synchronization between the transmitters and the UE in this paper. A promising direction for future research is the extension of our robust multipath-assisted SLAM algorithm to an unsynchronized sensor network.

Author Contributions: Methodology, Z.D.; Project administration, B.L.; Software, Z.D.; Writing—Original draft, Z.D.; Writing—Review & Editing, B.L. and C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Basic Research Program of Shaanxi, grant number 2021JQ-122.

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

References
1. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. IEEE Robot. Autom. Mag. 2006, 13, 99–110, doi:10.1109/MRA.2006.1638022.
2. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. IEEE Robot. Autom. Mag. 2006, 13, 108–117, doi:10.1109/MRA.2006.1678144.
3. Lajoie, P.Y.; Ramtoula, B.; Chang, Y. DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams. IEEE Robot. Autom. Lett. 2020, 5, 1656–1663, doi:10.1109/LRA.2020.2967681.
4. Chiang, K.W.; Tsai, G.J.; Chu, H.J. Performance Enhancement of INS/GNSS/Refresed-SLAM Integration for Acceptable Lane-Level Navigation Accuracy. IEEE Trans. Veh. Technol. 2020, 69, 2463–2476, doi:10.1109/TVT.2020.2966765.
5. Mendrzik, R.; Meyer, F.; Bauch, G. Enabling Situational Awareness in Millimeter Wave Massive MIMO Systems. IEEE J. Sel. Top. Signal Process. 2019, 13, 1196–1211, doi:10.1109/JSTSP.2019.2933142.
6. Chen, Z.; Zhu, G.; Wang, S. M3: Multipath Assisted Wi-Fi Localization with a Single Access Point. IEEE Trans. Mob. Comput. 2020, 20, 588–602, doi:10.1109/TMC.2019.2950315.
7. Ulmschneider, M.; Gentner, C.; Jost, T. Association of Transmitters in Multipath-Assisted Positioning. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Singapore, 4–8 December 2017.
8. Li, Z.; Tian, Z.; Wang, Z. Multipath-Assisted Indoor Localization Using a Single Receiver. IEEE Sens. J. 2021, 21, 692–705, doi:10.1109/JSEN.2020.3012786.
9. Li, Z.; Tian, Z.; Wang, Z. Multipath-Assisted Indoor Localization: Turning Multipath Signal from Enemy to Friend. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019.
10. Wielandt, S.; Strycker, L.D. Indoor Multipath Assisted Angle of Arrival Localization. Sensors 2017, 17, 2522, doi:10.3390/s17112522.
11. Witrisal, K.; Meissner, P.; Leitinger, E. High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend. IEEE Signal Mag. 2016, 33, 59–70, doi:10.1109/MSP.2015.2504328.
12. Gui, F.; Closas, P.; Dardari, D. Personal mobile radars with millimeter-wave massive arrays for indoor mapping. IEEE Trans. Mobile Comput. 2016, 15, 1471–1484, doi:10.1109/TMC.2015.2504328.
13. Guerra, A.; Gui, F.; Dardari, D. Occupancy grid mapping for personal radar applications. In Proceedings of the IEEE Statistical Signal Processing Workshop (SSP), Freiburg, Germany, 10–13 June 2018.
14. Zhang, H.; Tan, S.Y. TOA based indoor localization and tracking via single-cluster PHD filtering. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Singapore, 4–8 December 2017.

15. Kim, H.; Granström, K.; Gao, L. 5G mmWave Cooperative Positioning and Mapping Using Multi-Model PHD Filter and Map Fusion. *IEEE Trans. Wirel. Commun.* 2020, 19, 3782–3795, doi:10.1109/TWC.2020.2978479.

16. Mendzrik, R.; Wymeersch, H.; Bauch, G. Joint Localization and Mapping through Millimeter Wave MIMO in 5G Systems. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018.

17. Leitinger, E.; Meissner, P.; Rudisser, C. Evaluation of position-related information in multipath components for indoor positioning. *IEEE J. Sel. Areas Commun.* 2015, 33, 2313–2328, doi:10.1109/JSAC.2015.2430520.

18. Leitinger, E.; Meissner, P.; Witrisal, K. Simultaneous localization and mapping using multipath channel information. In Proceedings of the IEEE International Conference on Communication Workshop (ICCW), London, UK, 8–12 June 2015.

19. Gentner, C.; Ma, B.; Ullmschneider, M. Simultaneous localization and mapping in multipath environments. In Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS), Savannah, GA, USA, 11–14 April 2016.

20. Li, Z.; Tian, Z.; Wang, Z. WalkAround: Multipath-assisted Indoor Localization and Mapping Using a Single Receiver. In Proceedings of the IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020.

21. Ullmschneider, M.; Gentner, C. Multipath assisted positioning for pedestrians using LTE signals. In Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS), Savannah, GA, USA, 11–14 April 2016.

22. Gentner, C.; Jost, T.; Wang, W. Multipath Assisted Positioning with Simultaneous Localization and Mapping. *IEEE Trans. Wirel. Commun.* 2016, 15, 6104–6117, doi:10.1109/TWC.2019.2937781.

23. Karásek, R.; Gentner, C. Stochastic Data Association for Multipath Assisted Positioning Using a Single Transmitter. *IEEE Access* 2020, 8, 46735–46752, doi:10.1109/ACCESS.2020.2977558.

24. Liu, Y.; Lian, B.; Zhou, T. Gaussian message passing-based cooperative localization with node selection scheme in wireless networks. *Signal Processing*. 2019, 156, 166–176, doi:10.1016/j.sigpro.2018.10.023.

25. Leitinger, E.; Meyer, F.; Hlawatsch, F. A Belief Propagation Algorithm for Multipath-Based SLAM. *IEEE Trans. Wirel. Commun.* 2019, 18, 5613–5629, doi:10.1109/TWC.2016.2578336.

26. Leitinger, E.; Meyer, F.; Tufvesson, F. Factor Graph Based Simultaneous Localization and Mapping Using Multipath Channel Information. In Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops), Paris, France, 21–25 May 2017.

27. Leitinger, E.; Grebien, S.; Li, X. On the Use of Mpc Amplitude Information in Radio Signal Based SLAM. In Proceedings of the IEEE Statistical Signal Processing Workshop (SSP), Freiburg, Germany, 10–13 June 2018.

28. Leitinger, E.; Grebien, S.; Witrisal, K. Multipath-based SLAM Exploiting AoA and Amplitude Information. In Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, 20–24 May 2019.

29. Bar-Shalom, Y.; Li, X.-R. Multitarget-Multisensor Tracking: Principles and Techniques; Yaakov Bar-Shalom: Storrs, CT, USA, 1995; pp. 1–4.

30. Mahler, R.P.S.; Vo, B.T.; Vo, B.N. CPHD Filtering with Unknown Clutter Rate and Detection Profile. *IEEE Trans. Signal. Process.* 2011, 59, 3497–3513, doi:10.1109/TSP.2011.228316.

31. Vo, B.T.; Vo, B.N.; Hoseinnezhad, R. Robust multi-Bernoulli filtering. *IEEE J. Sel. Top. Signal Process.* 2013, 7, 399–409, doi:10.1109/JSTSP.2013.2252325.

32. Ristic, B.; Clark, D.; Vo, B.T.; Vo, B.N. Adaptive target birth intensity for PHD and CPHD filters. *IEEE Trans. Aerosp. Electron. Syst.* 2012, 48, 1656–1668, doi:10.1109/TAES.2012.6179885.

33. Soldi, G.; Meyer, F.; Braca, P.; Hlawatsch, F. Self-Tuning Algorithms for Multisensor-Multitarget Tracking Using Belief Propagation. *IEEE Trans. Signal. Process.* 2019, 67, 3922–3937, doi:10.1109/TSP.2019.2916764.

34. Kschischang, F.R.; Frey, B.J.; Loeliger, H.-A. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* 2001, 47, 498–519, doi:10.1109/18.910572.

35. Salmi, J.; Richter, A.; Koivunen, V. Detection and tracking of MIMO propagation path parameters using state-space approach. *IEEE Trans. Signal. Process.* 2009, 57, 1538–1550, doi:10.1109/TSP.2008.211823.

36. Mahler, R.P.S. Statistical Multisource-Multitarget Information Fusion; Artech House: Norwood, MA, USA, 2007.

37. Jo, K.; Chu, K.; Sunwoo, M. Interacting Multiple Model Filter-Based Sensor Fusion of GPS With In-Vehicle Sensors for Real-Time Vehicle Positioning. *IEEE Trans. Intell. Transp.* 2012, 13, 329–343, doi:10.1109/TITS.2011.2171033.

38. 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, doi:10.1109/TSP.2017.2688966.

39. Wymeersch, H.; Seco-Granados, G. Adaptive Detection Probability for mmWave 5G SLAM. In Proceedings of the IEEE 2nd 6G Wireless Summit (6G SUMMIT), Levi, Finland, 17–20 March 2020.

40. Williams, J.L.; Lau, R. Approximate evaluation of marginal association probabilities with belief propagation. *IEEE Trans. Aerosp. Electron. Syst.* 2014, 50, 2942–2959, doi:10.1109/TAES.2014.120568.

41. Meyer, F.; Kropfreiter, T.; Williams, J.L. Message passing algorithms for scalable multitarget tracking. *Proc. IEEE* 2018, 106, 221–259, doi:10.1109/JPROC.2018.2789427.

42. Meyer, F.; Hlinka, O.; Wymeersch, H.; Hlawatsch, F. Distributed localization and tracking of mobile networks including noncooperative objects. *IEEE Trans. Signal Inf. Process. Netw.* 2016, 2, 57–71, doi:10.1109/TISPEN.2015.2511920.
43. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. part I. dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1333–1364, doi:10.1109/TAES.2003.1261132.

44. Horridge, P.; Maskell, S. Using a probabilistic hypothesis density filter to confirm tracks in a multi-target environment. In Proceedings of the Informatik Schafft Communities (INFORMATIK-11), Berlin, Germany, 4–7 July 2011.

45. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447–3457, doi:10.1109/TSP.2008.920469.