Implementation and Validation of Multiple Target Tracking Technique for Indoor Applications

Uday Kumar Singh
SnT, University of Luxembourg
Luxembourg, Luxembourg
uday.singh@uni.lu

Mohammad Alaee-Kerahroodi
SnT, University of Luxembourg
Luxembourg, Luxembourg
mohammad.alaee@uni.lu

M. R. Bhavani Shankar
SnT, University of Luxembourg
Luxembourg, Luxembourg
bhavani.shankar@uni.lu

***Abstract*—Multiple target tracking (MTT) in the presence of false alarms (effect of clutter) and miss detections has always been a hot topic for research in a radar system. However, there is a shortage of literature describing the complex working and components of the MTT algorithm, especially in an indoor environment. Hence, this paper proposes a unique implementation of the MTT algorithm. The various features that make the working of the MTT algorithm possible are compiled in one place and thoroughly discussed. The developed MTT algorithm is validated for all potential challenges in an indoor scene, making the developed MTT algorithm more appealing to be used in practice. Further, extensive computer simulations with real and simulated measurements validate the working of the designed MTT algorithm. Lastly, the developed MTT algorithm is also used for people counting in an indoor environment, which has a shortage in the existing literature.**

I. Introduction

With the rapid increase in the demand for surveillance and monitoring of indoor scenes using radar sensors, tracking multiple targets in a complex environment (in the presence of clutter and blind spots) has become challenging [1]. The monitoring of indoor scenes could include surveillance of physically challenged people and the elderly. In literature, a family of solutions collectively known as multiple target tracking (MTT) algorithms are considered [2], [3]. To deal with multiple targets, particularly in false alarms, data association is the crucial component of the MTT algorithm [4], which filters the measurements from the targets of interest, thereby discarding the false alarms. Also, the data association assigns the new measurements to appropriate tracks. Subsequently, the Kalman filter [5] utilizes the assigned measurements to update the states of the tracks and predict the value of future measurements. For the association of measurements (sorting the measurements in the presence of false alarms), the literature suggests the nearest neighbor (NN) [6], global nearest neighbor (GNN) [2], and joint probabilistic data association (JPDA) [7] algorithm. While the NN and GNN are similar, GNN guarantees a global solution utilizing the Munkres algorithm [8]. The JPDA is the most advanced method for data association as it considers the probabilities and hypothesis testing. Although advanced data association techniques exist, until high precision is not required, the literature suggests using NN due to its simplicity and ease of deploy-ability [3].

In addition to utilizing different criteria, the concept of a validation gate is common in all data association algorithms. The validation gate acts as a filter to sort the targets' measurements from the spurious measurements. Even though data association is an essential component of the MTT algorithm, initialization, confirmation, and deletion of tracks are equally essential. The initialization, confirmation, and deletion of tracks in the combine are called track maintenance. However, most of the existing literature on MTT algorithms stops short of providing a complete working model from measurement to tracks, especially in an indoor environment. Therefore, to compile the entire work of the MTT algorithm in one place, a comprehensive implementation of the MTT algorithm is proposed in this paper. Moreover, the paper describes every component of the developed MTT algorithm in detail. These components have been designed considering the particularities of the indoor scene. For instance, the developed MTT algorithm is capable of dealing with the following justified challenges amid indoor environment and radar sensor irregularities:

1) The random number of the targets in the scene.
2) The time the targets enter in the scene and leave is entirely random.
3) The targets maneuver with random trajectories.
4) The radar sensor misses the targets detections (because of the limited field of view (FOV) of a radar sensor or targets' fluctuating RCS).

It is worth mentioning that rather introducing a new data association algorithm, this work aims to compile the different aspects of the MTT algorithm in one place via a novel implementation. Further, apart from data association, to deal with the indoor challenges mentioned above, more importantly for 1) and 2), initialization, confirmation, and deletion of tracks require extreme attention, notably in the presence of clutter. The presence of clutter in an indoor environment is inevitable because of the various static objects like chairs, tables, doors, and walls. Hence, in this paper, targeting the challenges in harsh indoor scenes, we cover all the key factors that enable the working of the MTT algorithm possible.

The first step of the developed MTT algorithm is data association, for which a popular NN algorithm is used. The measurements under a validation gate are subsequently assigned to the existing tracks during data association and used to update the tracks utilizing the Kalman filter. Further, the DBSCAN clustering algorithm [9] is used at each scan to check if the unassociated measurements can form new tracks. If DBSCAN clusters enough measurements, the new tracks are initialized. The updated tracks after data association are nominated as tentative for some predefined scans. Subsequently, when tracks are continuously updated for a predefined number of scans, they are nominated as confirmed tracks. The tracks which are not confirmed are finally deleted.

It should be noted that, apart from the NN algorithm and DBSCAN, the confirmation and deletion of tracks are based on logic. In this work, to account for the random movement of the targets, which is the case in the indoor environment, the unknown targets' motion is modeled as a discrete white noise acceleration (DWNA) motion model. The DWNA represents the targets' motion in an indoor scene and explicitly models the random motion of the targets via selectively choosing the accelerations variances along the coordinates axis. The developed MTT algorithm is validated for a fixed probability of false alarm (P_{FA}) and varying probability of detection (P_D). The performance of the developed algorithm is also validated with real radar measurements. Lastly, we propose using the developed MTT algorithm to count people in an indoor scene.

The rest of the paper is organized as follows: Section II describes the scenario and problem statement. In Section III, the proposed implementation of the MTT algorithm is discussed in detail. Section IV discussed the simulation results relevant to the validation of the developed MTT algorithm. Finally, Section V concludes the paper.

Notations: Scalar variables (constants) are denoted by lower (upper) case letters. Vectors (matrices) are denoted by bold face lower (upper) case letters. Superscripts $(\cdot)^T$, $(\cdot)^H$ and $(\cdot)^*$ denote matrix/vector transpose, complex conjugate transpose and scalar complex conjugation operation respectively. $\mathbb{E}[\cdot]$ denotes statistical expectation and $\mathbb{R}$ denote the set of real numbers. $\mathbf{I}_n$ denotes the identity matrix of cardinality n. $len(\cdot)$ denotes the number of elements in a vector and number of columns in a matrix. Set subtraction and null set is denoted by $\setminus$ and $\varnothing$, respectively. To access the i^{th} column of a matrix $\mathbf{A}$, $\mathbf{A}^i$ is used. Similarly, to access the i^{th} element of a vector $\mathbf{a}$, $\mathbf{a}^i$ is used.

II. Scenario and Problem Statement

In this section, we present the scenario and the problem considered. As mentioned in the introduction, with the motive of addressing the challenges in the indoor scene; the analysis of the developed MTT algorithm is generalized in a way that a) at a given instant of time, the number of targets is random, and b) the time the targets enter the scene and leave is also random. Further, to model the random movements of the targets, the DWNA motion model is considered. Assuming Q point targets in the scene, the evolution of the q^{th} target kinematic state vector with cardinality n_s $(\mathbf{s}_k^p \in \mathbb{R}^{n_s \times 1})$ at the k^{th} time instance (scan) is governed by

$$\mathbf{s}_k^q = \mathbf{f}(\mathbf{s}_{k-1}^q) + \mathbf{\Gamma}_a \mathbf{w}_k, \tag{1}$$

where, $\mathbf{s}_{k-1}^q$ is the state vector at the previous time instance of the q^{th} target, $\mathbf{f}(\cdot) \in \mathbb{R}^{n_s \times 1}$ governs the motion of the q^{th} target and is modeled as constant velocity (CV), and $\mathbf{w}_k \in \mathbb{R}^{2\times 1}$, accounts for random accelerations of the target along x and y axis. Further, $\mathbf{w}_k$, is assumed to be Gaussian distributed with zero mean and known covariance $\mathbf{Q}_a = \mathbb{E}[\mathbf{w}_k {\mathbf{w}_k}^T] = \begin{bmatrix} \sigma_{a_x} & 0 \\ 0 & \sigma_{a_y} \end{bmatrix}$, s.t. $\mathbf{w}_k \sim \mathcal{N}_{\mathbb{R}}(\mathbf{0}, \mathbf{Q}_a)$.

In this work, based on the 2D motion of the targets under the DWNA model, the target state vector is composed of target locations in 2D space (x_k^q, y_k^q), and the corresponding constant velocities $({v_x}_k^q, {v_y}_k^q)$ i.e. $\mathbf{s}_k^q = [x_k^q, {v_x}_k^q, y_k^q, {v_y}_k^q]^T$. Therefore, the noise-free $\mathbf{f}(\mathbf{s}_{k-1}^q)$ is given by

$$\mathbf{f}(\mathbf{s}_{k-1}^q) = \mathbf{F}\mathbf{s}_{k-1}^q.$$

where $\mathbf{F} = \begin{bmatrix} 1 & T_o & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_o \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and T_o is the coherent processing interval (CPI).

Further, in (1), as $\mathbf{\Gamma}_a \mathbf{w}_k$ is modelling the acceleration term, the

$$\mathbf{\Gamma}_a = \begin{bmatrix} 0.5T_o^2 & 0 \\ T_o & 0 \\ 0 & 0.5T_o^2 \\ 0 & T_o \end{bmatrix}.$$

Regarding the motion of targets in an indoor scene, the DWNA motion model facilitates the following peculiar requirements.

1) For low values of acceleration variances $(\sigma_{a_x}, \sigma_{a_y})$, the DWNA model matches the CV motion of the target (simple straight line motion).
2) For high values of acceleration variances, the DWNA model covers the random motion of the target (variable velocity).
3) Choosing the suitable values of acceleration variances helps in modeling the motion of the targets following different trajectories.

With an assumption that at each k, the radar sensor reports the estimate of the targets' positions in Cartesian coordinate, the measurements reported by the radar sensor for the q^{th} target $(\mathbf{z}_k^q \in \mathbb{R}^{2\times 1})$ is model by

$$\mathbf{z}_k^q = \mathbf{H}\mathbf{s}_k^q + \mathbf{v}_k, \tag{2}$$

where, $\mathbf{H}$ is the measurement matrix, $\mathbf{v}_k \in \mathbb{R}^{2\times 1}$ accounts for the error in estimating $\mathbf{z}_k^q$ via radar sensor. In this work, $\mathbf{v}_k$ is assumed to be Gaussian distributed with zero mean and a known covariance $\mathbf{R}_m = \mathbb{E}[\mathbf{v}_k {\mathbf{v}_k}^T]$, s.t. $\mathbf{v}_k \sim \mathcal{N}_{\mathbb{R}}(\mathbf{0}, \mathbf{R}_m)$.

Further, since, the radar sensor is reporting the 2D coordinates of targets, hence, $\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. In the presence of clutter, which is a case of point in this work, the radar sensor measurements are corrupted by the false alarm. Consequently, apart from the measurements from the targets $(\mathbf{z}_k^q,\ q = 1, 2, \ldots, Q)$, the measurements $(\mathbf{Z}_k)$, contains the false measurements $(\mathbf{Z}_k^c \in \mathbb{R}^{2\times n_c})$, where n_c depends on the severity of clutter (value of P_{FA}). Hence, the overall measurement used for target tracking is given by

$$\mathbf{Z}_k = [\mathbf{Z}_k^t\,,\, \mathbf{Z}_k^c] \in \mathbb{R}^{2\times M}, \tag{3}$$

where $\mathbf{Z}_k^t = [\mathbf{z}_k^1\,,\, \mathbf{z}_k^2\,,\, \ldots\,,\, \mathbf{z}_k^Q]$ and $M = Q + n_c$.

Referring to (3), this work concerns with the design of the MTT algorithm that effectively tracks the motion of the multiple targets. The proposed implementation of MTT includes filtering of target-oriented observations $(\mathbf{Z}_k^t)$ from $\mathbf{Z}_k$ thereby discarding spurious measurements $(\mathbf{Z}_k^c)$. The developed technique also includes assigning the filtered measurements to the appropriate tracks and tracks maintenance.

III. Proposed Implementation of MTT algorithm

In this section, the implementation of the developed MTT algorithm and its various components are discussed thoroughly. The prominent steps are data association, track maintenance, and Kalman filtering. The track maintenance, in turn, deals with the initialization, confirmation, and deletion of tracks.

A. Data Association and track deletion:

The data association acts as a filter and filter out the target measurements from $\mathbf{Z}_k$. In this work, the NN algorithm is used for the association. Among the various association techniques available in the literature, the NN is the most straightforward and suitable to apply when the measurements are sparsely distributed. Firstly, at k^{th} time instance, the NN algorithm calculates the Euclidean distances between the predicted measurement ($\mathbf{Z}_{k-1}^{p} \in \mathbb{R}^{2\times N}$) for the existing N tracks and the measurement from a sensor given by (3) as

$$d_{ij} = \|\mathbf{Z}_{k-1}^{p^i} - \mathbf{Z}_k^j\|, \;\; \forall i, \forall j \tag{4}$$

where $\|\cdot\|$ is the l_2 norm, $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$ are the index to access the columns of $\mathbf{Z}_{k-1}^{p}$ and $\mathbf{Z}_k$, respectively.

For simplicity, (4) can be equivalently written in vector as $\mathbf{d}_i = [d_{i1}, d_{i2}, \ldots, d_{ij}, \ldots, d_{iM}]$. Subsequently, for each existing track index (i), the minimum of $\mathbf{d}_i$ is calculated as $j_i^a = \arg\min_j \mathbf{d}_i \;\forall i$. Further, if the minimum is less than or equal to the predefined circular gate radius r, the corresponding measurement index is saved in an associated index array ($\mathbf{i}_a$) as $\mathbf{i}_a = [j_1^a, j_2^a, \ldots, j_j^a, \ldots, j_N^a]$. Henceforth, the measurements indexed by the elements of $\mathbf{i}_a$ are considered to be associated and processed for filtering. In filtering, the future states of the existing tracks are estimated using Kalman filter. Consequently, for $i_a = [1, 2, \ldots, len(\mathbf{i}_a)]$ and for an initial state estimate ($\hat{\mathbf{s}}_{k-1}^{i_a}$) and corresponding error covariance matrix ($\mathbf{P}_{k-1}^{i_a}$), the Kalman filter [10] executes the prediction and update as

1) Prediction:

$$\hat{\mathbf{s}}_k^{i_a} = \mathbf{F}\hat{\mathbf{s}}_{k-1}^{i_a}, \;\; \mathbf{P}_k^{i_a} = \mathbf{F}\mathbf{P}_{k-1}^{i_a}\mathbf{F}^T + \mathbf{Q}, \tag{5}$$

where $\mathbf{Q} = \mathbf{\Gamma}_a \mathbf{Q}_a \mathbf{\Gamma}_a^T$.

2) Update:

$$\mathbf{z}_k^{i_a} = \mathbf{H}\hat{\mathbf{s}}_k^{i_a}, \quad \mathbf{K}^{i_a} = \mathbf{P}_k^{i_a}\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^{i_a}\mathbf{H}^T + \mathbf{R}_m)^{-1}, \tag{6}$$

$$\hat{\mathbf{s}}_k^{i_a} = \hat{\mathbf{s}}_k^{i_a} + \mathbf{K}^{i_a}(\mathbf{Z}_k^{i_a} - \mathbf{z}_k^{i_a}), \quad \mathbf{P}_k^{i_a} = \mathbf{P}_k^{i_a} - \mathbf{K}^{i_a}\mathbf{H}\mathbf{P}_k^{i_a}. \tag{7}$$

Subsequently, using $\hat{\mathbf{s}}_k^{i_a}$, $\mathbf{Z}_k^{p^{i_a}}$, is updated as

$$\mathbf{Z}_k^{p^{i_a}} = \mathbf{H}\hat{\mathbf{s}}_k^{i_a}.$$

The updated $\mathbf{Z}_k^{p^{i_a}}$ $\forall i_a$ are used in the next scan i.e. at $k + 1^{th}$ instance for data association. In case of miss detections at $k + 1^{th}$ instance, since the target observations ($\mathbf{Z}_k^t$) are necessarily not available for some existing tracks, there states are updated by the predicted measurements at k^{th} instance as

$$\hat{\mathbf{s}}_{k+1}^{i_a} = \hat{\mathbf{s}}_{k+1}^{i_a} + \mathbf{K}^{i_a}(\mathbf{Z}_k^{p^{i_a}} - \mathbf{Z}_{k+1}^{i_a}). \tag{8}$$

Also, in the case of the miss detections, attention has to be paid to the two possible alternative scenarios a) the targets have left the scene permanently, or b) the detections are missed because of the sensor irregularities (limited FOV or targets RCS fluctuations). For the latter scenario, the tracks should be kept on updated using (8) for a few scans, and for the former, the tracks should be deleted. The scenarios mentioned above are tackled properly in the developed tracker by setting a delete counter ($\mathbf{dc}$) for each existing track. In the case of miss detection at each scan, the deleted counter is updated and checked if they cross the predefined threshold. If a delete counter for a particular track crosses the threshold, the corresponding tracks are deleted. Otherwise, the states of the existing tracks continue to update using (8). Further, the track deletion is performed by removing the relevant entry in $\mathbf{Z}_k^p$ so that in future scans, no measurements in $\mathbf{Z}_k$ are associated with the deleted tracks.

B. Tracks Initialization and Confirmation

Apart from false alarms, the $(M - N)$ measurements, which are not associated with any of the existing N tracks, could be the detections from valid targets. However, due to the absence of relative entry in $\mathbf{Z}_k^p$, the measurements are unassociated. Therefore, the unassociated measurements are checked in the next step to form new tracks. The formation of the new tracks is undertaken by exploiting the fact that the measurements from a particular target can be classified using distance metrics; consequently, the measurements can be grouped into clusters. Accordingly, the number of clusters reflects the number of newly formed tracks. Therefore, in this work, the celebrated DBSCAN algorithm is used to cluster the unassociated measurements.

If $\mathbf{Z}_k^{\mathbf{i}_a}$ denotes the group of associated measurements, the group of unassociated measurements ($\mathbf{Z}_k^{ua} = \mathbf{Z}_k \setminus \mathbf{Z}_k^{\mathbf{i}_a}$) are stacked for the consecutive scans ($\mathbf{Z}_k^{cl} = [\mathbf{Z}_{k-1}^{cl}, \mathbf{Z}_k^{ua}]$). Next, utilizing $\mathbf{Z}_k^{cl}$, the DBSCAN algorithm will look for the formation of the number of clusters with their corresponding labels. Let at k^{th} instance, the DBSACN algorithm identify L clusters with labels $\mathbf{l} = [l_1, l_2, \ldots, l_L]$, then for $i_l = [1, 2, \ldots, len(l)]$, the labelled measurement $\mathbf{Z}_k^{cl^{i_l}}$ from $\mathbf{Z}_k^{cl}$ at k^{th} instance is treated as a predicted measurements for the newly formed L tracks. The measurements which are not labeled in any of the L cluster are the measurements from clutter and treated as false alarm.

In the successive scans, for associating the measurements with the newly form L tracks, $\mathbf{Z}_k^{cl^{i_l}}$ are stacked in $\mathbf{Z}_k^p$ as $\mathbf{Z}_k^p = [\mathbf{Z}_k^p, \mathbf{Z}_k^{cl^{i_l}}]$. Further, along-with updating $\mathbf{Z}_k^p$, $\hat{\mathbf{s}}_k^{i_l}$ and $\mathbf{P}_k^{i_l}$ for new L tracks are initialized as $\mathbf{0}_{n_s\times 1}$ and $\mathbf{I}_{n_s\times n_s}$, respectively. The updated $\hat{\mathbf{s}}_k^{i_l}$, are initially nominated as tentative tracks. Once the tracks are updated successively for some predefined scans, the tracks are nominated as confirmed. Like track deletion, the track confirmation is also performed by setting a confirmed counter ($\mathbf{cc}$) for each track. The confirmed counter updates every time the associated measurement updates $\hat{\mathbf{s}}_k$. The confirmed counter for each track is checked at each scan to cross the predefined threshold. If for a particular track the confirmed counter exceeds the predefined threshold, the corresponding track is nominated as confirmed. Otherwise, the track remains nominated as a tentative track and eventually deleted by using the track deletion step elaborated in III-A.

The developed MTT algorithm can be implemented by following the steps mentioned in the pseudo code 1. The symbols used in the presentation of pseudo code 1, are defined in Table. I.

Algorithm 1: Proposed implementation of MTT algorithm

```
   Data: $r$, $\mathbf{Z}_k$, $K$, $Q$, $T$, $n_s$, $\mathbf{Q} = \mathbf{\Gamma}_a \mathbf{Q}_a \mathbf{\Gamma}_a^T$, $\sigma_{a_x}, \sigma_{a_y}$, $\mathbf{R}_m = r_m \mathbf{I}_2$,
         $r_m$, $P_{FA}$, $P_D$, $\delta_1$, $\delta_2$
1  initialization: $MNT$, $\mathbf{dc} = \mathbf{0}_{1\times MNT}$, $\mathbf{cc} = \mathbf{0}_{1\times MNT}$, $\mathbf{Z}_0^p$, $\mathbf{Z}_{k-1}^{cl}$;
2  Data Association using NN (distance metric):
3  while $k <= K$ do
4      for $i \leftarrow 1$ to $len(\mathbf{Z}_{k-1}^p)$ do
5          for $j \leftarrow 1$ to $len(\mathbf{Z}_k)$ do
6              $d_{ij} = \|\mathbf{Z}_k^{p^i} - \mathbf{Z}_k^j\|$;
7          end
8          $\mathbf{d}_i = [d_{i1}, d_{i2}, \ldots, d_{ij}, \ldots, d_{iM}]$;
9          $[val, indx] = \arg\min_j \mathbf{d}_i$;
10         Check the circular gating condition;
11         if $val <= r$ then
12             $j_i^a = indx$;
13         else
14             $j_i^a = 0$;
15         end
16         Store the associated measurement indexes in $\mathbf{i}_a$;
17         $\mathbf{i}_a = [j_1^a, j_2^a, \ldots, j_{i-1}^a, j_i^a]$;
18     end
19     Update the existing tracks using Kalman Filter;
20     for $i_a \leftarrow 1$ to $len(\mathbf{i}_a)$ do
21         if $j_i^a \neq 0$ then
22             1) Prediction;
23             Use (5);
24             2) Update;
25             Use (6) and (7);
26             To confirm the tracks;
27             if $\mathbf{cc}^{i_a} <= \delta_1$ then
28                 $tentativetracks^{i_a} = \hat{\mathbf{s}}_k^{i_a}$;
29                 $\mathbf{cc}^{i_a} = \mathbf{cc}^{i_a} + 1$;
30             else
31                 $confirmedtracks^{i_a} = \hat{\mathbf{s}}_k^{i_a}$;
32             end
33         else
34             For handling target miss detections;
35             The predict and update step are same as line $23$ and $25$.
                 Nevertheless, since the measurements are missing, $\mathbf{Z}_k^p$ is
                 used in place of $\mathbf{Z}_k$.;
36             if $\mathbf{cc}^{i_a} <= \delta_1$ then
37                 $tentativetracks^{i_a} = \hat{\mathbf{s}}_k^{i_a}$;
38                 $\mathbf{cc}^{i_a} = \mathbf{cc}^{i_a} + 1$;
39             else
40                 $confirmedtracks^{i_a} = \hat{\mathbf{s}}_k^{i_a}$;
41                 update delete counter;
42                 $\mathbf{dc}^{i_a} = \mathbf{dc}^{i_a} + 1$;
43             end
44         end
45     end
46     Clustering to Identify if unassociated observations can form new
         tracks;
47     $\mathbf{Z}_k^{ua} = \mathbf{Z}_k \setminus \mathbf{Z}_k^{\mathbf{i}_a}$;
48     if $\mathbf{Z}_k^{ua} \neq \varnothing$ then
49         $\mathbf{Z}_k^{cl} = [\mathbf{Z}_{k-1}^{cl}, \mathbf{Z}_k^{ua}]$;
50         $[\mathbf{l}] = DBSCAN(\mathbf{Z}_k^{cl})$;
51         for $i_l \leftarrow 1$ to $len(\mathbf{l})$ do
52             $\mathbf{Z}_k^p = [\mathbf{Z}_k^p, \mathbf{Z}_k^{cl^{i_l}}]$, $\hat{\mathbf{s}}_k^{i_l} = \mathbf{0}_{n_s\times 1}$, $\mathbf{P}_k^{i_l} = \mathbf{I}_{n_s\times n_s}$,
                 $\mathbf{dc}^{i_l} = 0$;
53         end
54     else
55         do nothing;
56     end
57     To delete the tracks;
58     for $d \leftarrow 1$ to $len(\mathbf{dc})$ do
59         if $\mathbf{dc}^d <= \delta_2$ then
60             $\mathbf{Z}_k^{p^d} = \varnothing$, $\hat{\mathbf{s}}_k^d = \varnothing$, $\mathbf{P}_k^d = \varnothing$, $\mathbf{dc}^d = 0$ ;
61         else
62             do nothing;
63         end
64     end
65 end
```

TABLE I: Definition of symbols used in psuedo code 1.

Symbols	Definition
r_m	Measurement noise variance along x and y axis
δ_1	Tracks deletion threshold
δ_2	Tracks confirmation threshold
MNT	Maximum number of tracks managed by the tracker
$\mathbf{dc}$	Tracks deletion counter
$\mathbf{cc}$	Tracks confirmation counter
$\varnothing$	Null set (used to denotes deletion of entry.)

IV. Simulation Results

This section discusses a detailed and thorough description of the simulation setup and results. The simulations are performed for two simulated scenarios and one real scenario. Scenario I: the multiple targets are present in the observation scene throughout the scanning time of the radar sensor. Scenario II: the targets enter the scene and leave the observation scene randomly. Scenario I depicts the performance of the developed tracker when the scene is static (in the sense of a number of targets); on the contrary, Scenario II resembles a practical situation where the scene is dynamic. Scenario III: the measurements are recorded in a room using the TI's IWR6843AOPEVM mm-wave radar sensor.

The Scenario I and Scenario II, both are simulated for 200 CPIs, i.e. $K = 200$. To realize the effects of sensor irregularities in simulations, at each time sample, the P_D for each target in the scene is chosen randomly between 0.7 and 1. The r for data association is chosen 1; it should be worth mentioning that the larger value of r could be chosen if the measurements are extremely noisy. For simulating the random motion for each target, such that it mainly represents CV and accelerating motion model, the σ_{ax} and σ_{ay} both are chosen randomly between 10^{-1} and 10^4. Also in simulations $T_o = 0.01$ sec, $\delta_1 = 4$ scans , $\delta_2 = 5$ scans, and $MNT = 20$ tracks.

The Scenario III, deals with collecting the real measurements in an indoor scene using the TI's IWR6843AOPEVM mm-wave radar sensor, whose configurations are detailed in Table II. The measurements are recorded for two people moving in a circular trajectory. The measurements are recorder for 269 CPIs, i.e $K = 269$. It should be noted that the measurements inherently contains miss detections amid P_D of the radar sensor. In the Kalman filter, to match the targets' state model with the circular trajectory, the σ_{ax} and σ_{ay} are tuned and set at $10^{-0.5}$ and 10^2, respectively. Apart from K, σ_{ax}, and σ_{ay}, all other parameters are the same as Scenario I and Scenario II.

A. Scenario I

The simulations are performed for 4 targets, i.e., $Q = 4$. The targets are in the scene for $K = 200$ time samples, $r_m = 0.01$, and $P_{FA} = 10^{-4}$. The ground truth (GTs), estimated tracks (ETs), target measurements (TMs), and false measurements (FMs) are shown in Fig. 1. In Fig. 2, the number of target measurements available at k^{th} time sample and the number of targets estimated by the developed tracker are shown. The targets' observation time samples, the measurements time samples, and the time samples for which the targets are considered to be lost (due to miss detections)

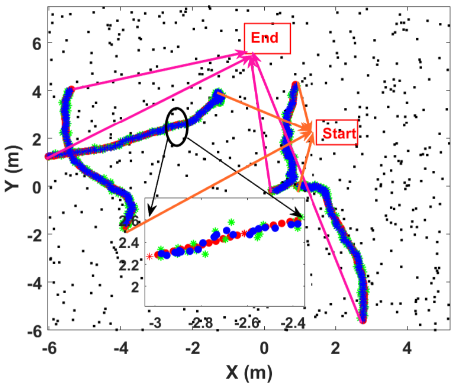


Fig. 1: GTs(red), ETs(blue), TMs(green), and FMs(black).

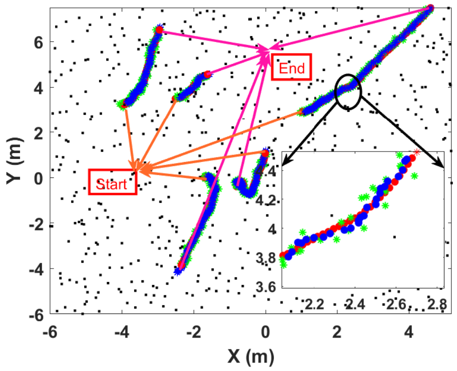


Fig. 3: GTs(red), ETs(blue), TMs(green), and FMs(black).

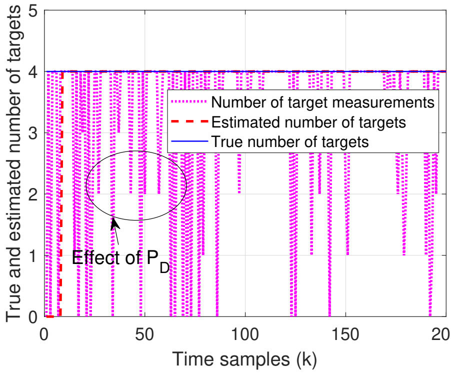


Fig. 2: True and estimated number of targets with number of target measurements.

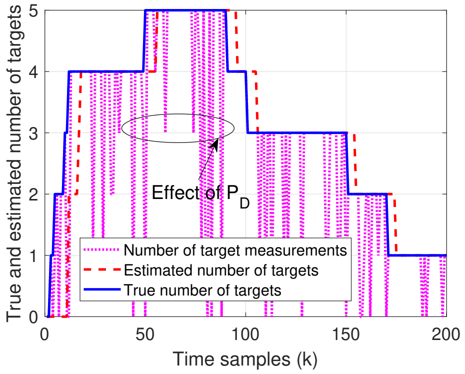


Fig. 4: True and estimated number of targets with a number of target measurements.

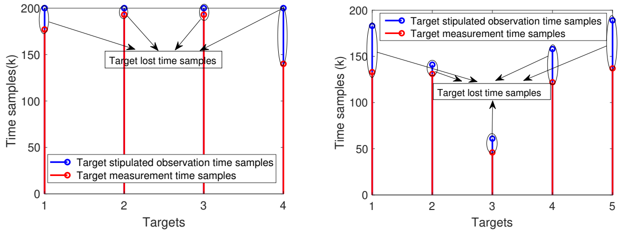
are shown in Fig. 6. It is evident from Fig. 1 that although the tracker input are the targets' measurements and false measurements, the tracker is successfully tracking the targets' trajectories without initiating any false tracks. Also, as shown in Fig. 2 and Fig. 6, although there are drops in the number of target measurements due to P_D, the tracker is not losing the tracks. This is because the predicted measurements from the immediate last scan are used to update the tracks. Moreover, Fig. 2 also reveals that the developed tracker is immune to the effect of P_D and estimates the exact number of targets present in the radar's observation scene.

B. Scenario II

Unlike Scenario I, the $Q = 5$ targets are not present in the scene for the whole observation time. They enter and leave the scene randomly. The setup helps to realize the real time situation where at any given time, the number of targets is not fixed and people counting is an important application. Apart from the random presence of the targets, the other simulation parameters are the same as Scenario I. The relevant simulation results are shown in Fig. 3, Fig. 4, and Fig. 5b. It is evident from Fig. 3 that the developed tracker tracks the targets' trajectory and creates and deletes the tracks successfully. For instance, the developed MTT algorithm tracks the trajectory of the fourth target for the duration it is active in the scene. Subsequently, the corresponding trajectory is deleted when the fourth target leaves the scene. Fig. 5b also shows that unlike Scenario I, the targets are not present in the scene for all time samples. Also, the time samples for which the target measurements are available and the time samples for which the target is considered to be lost are shown in Fig. 5b. Further, in Fig. 4, it is shown that the true number of targets in the scene is dynamically changing, and the tracker is estimating the same. Notably, between time samples 50 to 90, all the five targets are active in the scene, and the tracker estimates the 5 targets. This implies that the developed tracker counts the number of people active in the scene at any particular instance and could be successfully used for people counting. Form Fig. 4; it can also be observed that the curve for estimated and true number of targets are not aligned. The reason is that the tracker takes some scans to confirm and delete the tracks.

TABLE II: Radar sensor configurations.

Frequency band	$60 - 64$GHz
Number of Tx	3
Number of Rx	4
Frame rate	10fps
Range resolution	0.044m
Maximum unambiguous range	9.02m
Maximum radial velocity	1m/s
Radial velocity resolution	0.13m/s



(a) Scenario I (b) Scenario II

Fig. 5: Targets observation, measurement, and lost time samples for (a) Scenario I and (b) Scenario II

C. Section III

The recorded radar measurements (RMs) for two people walking in a circular trajectory is shown in Fig. 6. As depicted in Fig. 6, unlike Scenarios I and II, the measurements from a single target are composed of multiple TMs, due to the extended target effect. Also, the RMs are comprised of FMs, which makes identification of the target trajectories impossible. The clustering would be used to tackle the extended target effect as measurements can be clustered into a group. However, before clustering, the FMs have to reduce as they can form a valid clustering class and can lead to false tracks during tracking. Therefore, to reduce the FMs, we used the amplitudes of the measurements as a sorting criterion motivated by the fact that the reflections from multi-paths are weaker than those from targets. For this, an amplitude threshold is set on the amplitudes of the measurements. Subsequently, only measurements crossing the amplitude threshold are clustered. The measurements obtained after the thresholding and clustering are shown in Fig. 6. The tracker performance are shown in Fig. 6 and Fig. 7. As shown in Fig. 6, the tracker correctly estimates and discern the targets' motion. In Fig. 7, the tracker performance is summarized in the presence of missed detections.

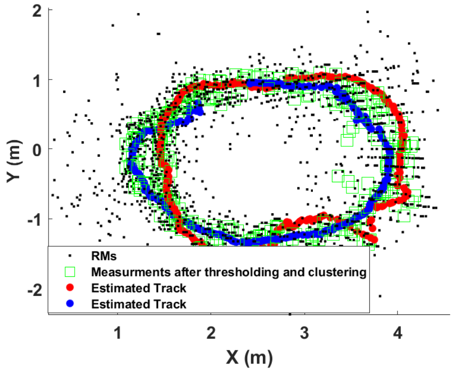


Fig. 6: ETs, measurements obtained after the thresholding and clustering, and RMs.

V. Conclusion

This work proposes and demonstrates a unique implementation of the MTT technique. The essential elements of the developed MTT algorithm involving data association, Kalman filtering, and track maintenance are discussed in detail. The track deletion and confirmation are executed by maintaining a delete counter and confirmation counter. The desirable performance of the developed MTT algorithm is demonstrated by the simulations performed for three scenarios; Scenario I, Scenario II, and Scenario III. Particularly, Scenario I deals with validating the designed tracker in the application where the scene is static regarding the number of targets. On the contrary, Scenario II validates the tracker's performance for the people counting application. Lastly, Scenario III validates the performance of the developed MTT algorithm on real measurements and establishes its utility in efficiently tracking people in an indoor scene.

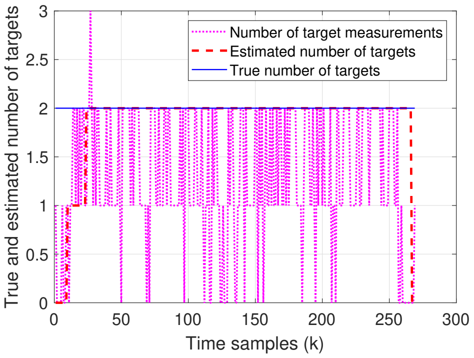


Fig. 7: True and estimated number of targets with the number of measurements.

Acknowledgment

The authors' work is supported by the Luxembourg National Research Fund (FNR) through the SPRINGER Project No. 12734677.

References

[1] M. A. Richards, *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005.

[2] S. S. Blackman, "Multiple-target tracking with radar applications," *Dedham*, 1986.

[3] J. Palacios, G. Bielsa, P. Casari, and J. Widmer, "Single-and multiple-access point indoor localization for millimeter-wave networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1927–1942, 2019.

[4] H. Leung, Z. Hu, and M. Blanchette, "Evaluation of multiple radar target trackers in stressful environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 2, pp. 663–674, 1999.

[5] G. Welch, G. Bishop *et al.*, "An introduction to the Kalman filter," 1995.

[6] H. You, X. Jianjuan, and G. Xin, *Radar data processing with applications*. John Wiley & Sons, 2016.

[7] Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable, "Tracking and data association," 1990.

[8] F. Bourgeois and J.-C. Lassalle, "An extension of the munkres algorithm for the assignment problem to rectangular matrices," *Communications of the ACM*, vol. 14, no. 12, pp. 802–804, 1971.

[9] Y. Chen, L. Zhou, S. Pei, Z. Yu, Y. Chen, X. Liu, J. Du, and N. Xiong, "KNN-BLOCK DBSCAN: Fast clustering for large-scale data," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3939–3953, 2021.

[10] U. K. Singh, M. Alaee–Kerahroodi, and M. R. B. Shankar, "RKHS based State Estimator for Radar Sensor in Indoor Application," in *2022 IEEE Radar Conference (RadarConf22)*, 2022, pp. 01–06.