

# Landmark Management in the Application of Radar SLAM

Shuai Sun\* and Beth Jelfs† and Kamran Ghorbani‡ and Glenn Matthews‡ and Christopher Gilliam†

\* Navigation College, Dalian Maritime University, Liaoning, China

E-mail: shuai.sun@dlnu.edu.cn

† Dept. Electronic, Electrical & Systems Engineering, University of Birmingham, Birmingham, UK

E-mail: {b.jelfs, c.gilliam.1}@bham.ac.uk

‡ School of Engineering, RMIT University, Melbourne, Australia

E-mail: {kamran.ghorbani, glenn.matthews}@rmit.edu.au

**Abstract**—This paper focuses on efficient landmark management in radar based simultaneous localization and mapping (SLAM). Landmark management is necessary in order to maintain a consistent map of the estimated landmarks relative to the estimate of the platform’s pose. This task is particularly important when faced with multiple detections from the same landmark and/or dynamic environments where the location of a landmark can change. A further challenge with radar data is the presence of false detections. Accordingly, we propose a simple yet efficient rule based solution for radar SLAM landmark management. Assuming a low-dynamic environment, there are several steps in our solution: new landmarks need to be detected and included, false landmarks need to be identified and removed, and the consistency of the landmarks registered in the map needs to be maintained. To illustrate our solution, we run an extended Kalman filter SLAM algorithm in an environment containing both stationary and temporally stationary landmarks. Our simulation results demonstrate that the proposed solution is capable of reliably managing landmarks even when faced with false detections and multiple detections from the same landmark.

## I. INTRODUCTION

Simultaneous localization and mapping (SLAM) aims to estimate the pose (location and heading) of a mobile platform, whilst simultaneously sensing the surrounding environment. SLAM in vehicular based applications is an active field of research [1], [2]. The design of a SLAM algorithm is closely related to the modeling of the environment. Depending on the specific environment, as well as the types of on-board sensor employed, there are two main approaches to modeling the surrounding environment. One is a landmark based formulation [3], where the sensors are used to detect surrounding objects which are then abstracted as landmarks, with the goal to simultaneously estimate the location of these landmarks. The second approach is an occupancy grid based formulation [4], where the surrounding environment is segmented into a number of disjoint occupancy grids, and the task of environment mapping is conducted by estimating whether each occupancy grid is occupied or not.

In the past few decades, researchers have mainly focused on developing algorithms and conducting practical experiments on Lidar based SLAM or visual based SLAM [5]. However,

these approaches can have limitations in harsh weather conditions, hence more recently there has been increased interest in SLAM using radar sensors which appear to be better suited to these conditions, especially with rain or fog [6]. Applying high resolution radar sensors for SLAM can have its own challenges, especially regarding measurement modeling. Firstly, unlike many existing point landmark based SLAM approaches, landmarks of a certain size contain multiple scattering/reflecting points and thus return multiple radar detections [7]. Secondly, the aspect angle between the mobile platform and the landmarks varies during the motion of the mobile platform, leading to varying number of radar detections returned for the same landmark. Finally, radar detections may occur due to clutter rather than true landmarks. Thus, an effective landmark management scheme requires the ability to reject spurious measurements at landmark initialization, and remove false landmarks from the system.

In addition to the complexity in the radar sensor data, SLAM in a dynamic environment proves to be more challenging than in a static environment [8]. Most existing works on SLAM assume either a stationary environment or the capability to discriminate between static and dynamic features [9]. A common strategy is to eliminate the moving objects and only use static objects to update the pose of the mobile platform [10]. However, this approach does not account for low-dynamic environments, that is, one in which landmarks can be stationary or temporally stationary [11]. For example, in a car park, most vehicles are stationary however some will enter and exit the parking area. In this setting, landmarks that have been confirmed and registered in the system should be removed if their positions are altered.

In this paper, we propose a rule based landmark management scheme for SLAM in low-dynamic environments using radar sensors. We aim to produce a consistent and efficient landmark management scheme that can be applied to existing SLAM solutions. We provide a detailed illustration regarding the key steps required to confirm/initialize new landmarks, associate landmarks with multiple radar detections, remove landmarks and merge landmarks. To illustrate the proposed landmark management scheme we choose the

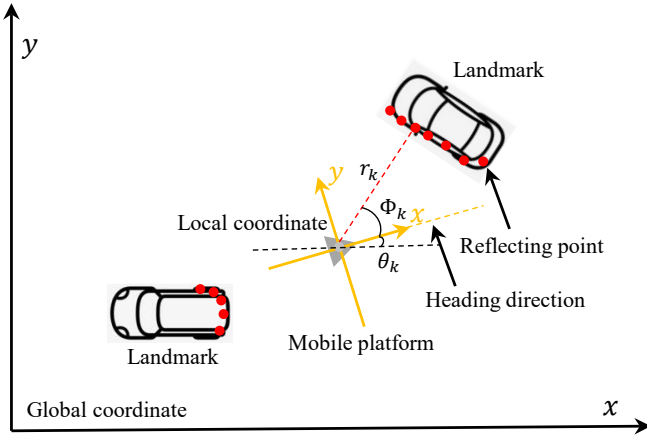


Fig. 1. Illustration of a typical considered SLAM scene.

extended Kalman filter (EKF) SLAM algorithm [12]. We evaluate our management scheme by simulating a mobile platform with a frequency-modulated continuous-wave (FMCW) radar sensor moving around a typical car park environment. The scene contains a number of parked vehicles as landmarks and the locations for some of the landmarks in the scene may change. The simulation results show that our management scheme is robust to false radar detections, multiple detections from a single landmark and landmarks changing position.

This paper is organized as follows: in Section II we formulate the FMCW Radar SLAM problem. Section III provides our detailed rule-based landmark management scheme. Simulation results are presented in Section IV and Section V concludes the work.

## II. PROBLEM FORMULATION

Following the formulation of the EKF-SLAM [12], we define an augmented state vector at time  $k$ , denoted by  $\mathbf{x}_k^a$ , as

$$\mathbf{x}_k^a = \left[ \underbrace{x_k, y_k, \theta_k}_{\mathbf{x}_k^m}, \underbrace{\mathbf{p}_k^1, \dots, \mathbf{p}_k^{N_k}}_{\mathbf{x}_k^\ell} \right]^T, \quad (1)$$

where  $\mathbf{x}_k^m = [x_k, y_k, \theta_k]^T$  represents the pose of the mobile platform, namely the 2D location and heading of the mobile platform, and  $\mathbf{x}_k^\ell = [\mathbf{p}_k^1, \dots, \mathbf{p}_k^{N_k}]^T$  is the vector of landmarks registered in the system, with each  $\mathbf{p}_k^n = [p_k^n(x), p_k^n(y)]$  representing the 2D global location of landmark  $n$ . Note that the total number of landmarks  $N_k$  will vary with time as landmarks are added and removed.

In order to identify landmarks, we assume the platform has a FMCW radar to sense its surrounding environment. This sensor measures the range and azimuth direction of a landmark relative to the platform's pose. Realistically, however, each landmark may produce multiple radar detections the number of which varies with the aspect angle between the mobile platform and the landmark [13]. An example of

this scenario is shown in Fig. 1 where the landmarks are exemplified by vehicles. The sensor measurements obtained in this scenario can be represented by using a reflecting point data model [14]. Thus, a radar measurement  $\mathbf{z}_k = [r_k, \phi_k]^T$  obtained at time  $k$  contains the relative range,  $r_k$ , and azimuth,  $\phi_k$ , that correspond to a certain reflecting point  $(x_n^i, y_n^i)$  from landmark  $n$  such that

$$\mathbf{z}_k = h_n(\mathbf{x}_k^a) + \boldsymbol{\xi}_k, \quad (2)$$

where

$$h_n(\mathbf{x}_k^a) = \begin{bmatrix} \sqrt{(x_k - x_n^i)^2 + (y_k - y_n^i)^2} \\ \arctan\left(\frac{y_k - y_n^i}{x_k - x_n^i}\right) - \theta_k \end{bmatrix}, \quad (3)$$

and  $\boldsymbol{\xi}_k \sim \mathcal{N}(0, R_k)$  is additive zero mean Gaussian noise with covariance matrix,  $R_k$ . We assume the sensor has a 360 degree field of view but is limited to a maximum detection range [15].

## III. LANDMARK MANAGEMENT

In this section, we present our landmark management scheme under the recursive EKF framework. Since each landmark may produce multiple radar detections, the total number of detection points can, in practice, be very large. Hence, an efficient landmark management scheme which does not significantly increase the computational load is required. If we consider the radar point cloud at time point  $k$ , denoted by  $Z_k$  then there are three sources for the data:

- from existing objects that have already been registered as landmarks in the state vector  $\mathbf{x}_k^a$ ;
- from objects that have not been registered in the state vector  $\mathbf{x}_k^a$ ;
- from false detections.

Hence, when processing the radar detections,  $Z_k$ , the first step is data association between the registered landmarks and the received radar measurements, where each landmark may be associated with multiple radar point detections. The associated measurements can then be used to update the system state. Next any remaining detections which have not been associated with an existing landmark are used to determine if any new landmarks are to be initialized.

In the following we present a detailed description of our landmark management scheme for a single iteration of the EKF-SLAM. The main processing consists of the following three modules: state prediction; state update; landmark removal and new landmark confirmation/inclusion. We provide a complete summary of the steps in this process in Algorithm 1.

### A. EKF Prediction

Before performing the update of the system using the radar detections we first perform prediction of the system state given the filtered augmented state vector from the previous time step,  $\mathbf{x}_{k-1|k-1}^a$ , and the state covariance matrix,  $P_{k-1|k-1}^a$ . The prediction is implemented based on a motion

---

**Algorithm 1: Landmark management for EKF SLAM**


---

**Input:** Initial pose of the mobile platform,  $\mathbf{x}_0^m = [x_0, y_0, \theta_0]$ , Model parameters in **Table I**, Radar detections  $Z_k$ , Odometer reading  $\bar{\mathbf{u}}_k$

**Output:** System state  $\mathbf{x}_{k|k}^a, P_{k|k}^a$

Recursion: **for**  $k = 1; k < K; k++$  **do**

EKF Prediction  $\rightarrow (\mathbf{x}_{k|k-1}^a, P_{k|k-1}^a)$

Compute the predicted state based on (7);

EKF Update  $\rightarrow (\mathbf{x}_{k|k}^u, P_{k|k}^u)$

Apply “Sifting” to select  $Z_k^s$  from  $Z_k$ ;

**for**  $\mathbf{p}_k^n \in \mathbf{x}_{k|k-1}^a$  **do**

Associate  $\mathbf{p}_k^n$  with  $Z_k^s$  using (8);

Update  $\mathbf{x}_{k|k-1}^a, P_{k|k-1}^a$  using (9);

Apply landmark removal logic  $\rightarrow (\mathbf{x}_{k|k}^r, P_{k|k}^r)$

**if**  $\mathbf{p}_k^n$  is confirmed to be removed **then**

Remove the corresponding entries in  $\mathbf{x}_{k|k}^u, P_{k|k}^u$ ;

**end**

**end**

New landmark Inclusion  $\rightarrow (\mathbf{x}_{k|k}^{a, N_k^\ell}, P_{k|k}^{a, N_k^\ell})$

Cluster radar detections in  $Z_k^r$ ;

Apply landmark confirmation for each cluster using Rule 1 and Rule 2;

**if**  $\mathbf{c}_k^j$  is confirmed **then**

Include  $\mathbf{c}_k^j$  to the system using (10)–(11);

**end**

Landmark Merging  $\rightarrow (\mathbf{x}_{k|k}^a, P_{k|k}^a)$

Apply landmark merging for the registered landmarks;

**end**

---

model of the mobile platform, the odometer data, as well as the assumption that the position of landmarks remains static. The motion of the mobile platform is modeled by a state transition equation [16]

$$\mathbf{x}_k^m = f_m(\mathbf{x}_{k-1}^m, \mathbf{u}_k) + \boldsymbol{\omega}_k, \quad (4)$$

where  $\mathbf{u}_k$  is the control input for the mobile platform, and  $\boldsymbol{\omega}_k \sim \mathcal{N}(\boldsymbol{\omega}_k; 0, Q_k)$  is the zero mean Gaussian process noise with covariance matrix  $Q_k$ . The control input is defined as  $\mathbf{u}_k = [v_k, \psi_k]$  where  $v_k$  is the velocity and  $\psi_k$  is the yaw rate, such that the transition of each component can be expressed as

$$\begin{aligned} x_k &= x_{k-1} + v_k \Delta t \cos\left(\theta_{k-1} + \frac{\Delta t \psi_k}{2}\right), \\ y_k &= y_{k-1} + v_k \Delta t \sin\left(\theta_{k-1} + \frac{\Delta t \psi_k}{2}\right), \\ \theta_k &= \theta_{k-1} + \psi_k \Delta t, \end{aligned} \quad (5)$$

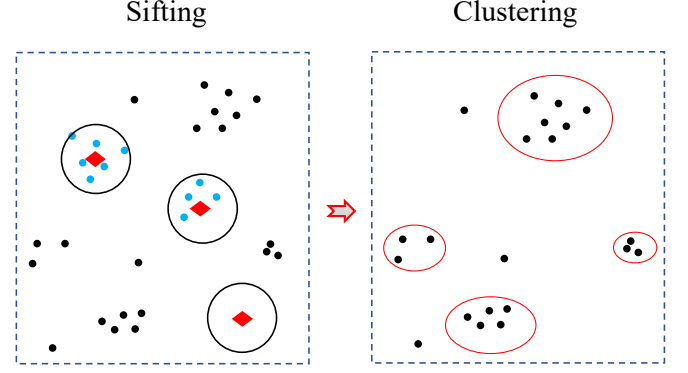


Fig. 2. Illustration of sifting (on the left); each red diamond shape represents a registered landmark in the system. The remaining measurements (on the right) are clustered for the subsequent new landmark identification and inclusion. Measurements that do not belong to a cluster are treated as false detections and removed.

where  $\Delta t$  is the time interval between time  $k-1$  and time  $k$ .

In reality, the control inputs are obtained from wheel rotation sensors, which provide noisy measurements of the true control inputs. Therefore the sensor readings of the velocity and yaw rate are denoted by  $\bar{\mathbf{u}}_k = [\bar{v}_k, \bar{\psi}_k]$ , and modelled as

$$\bar{\mathbf{u}}_k = \mathbf{u}_k + \boldsymbol{\eta}_k, \quad (6)$$

with  $\boldsymbol{\eta}_k \sim \mathcal{N}(\boldsymbol{\eta}_k; 0, U_k)$  a zero mean Gaussian noise and  $U_k$  the corresponding covariance matrix. Given these noisy odometer measurements, the EKF prediction from time  $k-1$  to time  $k$  is [17]

$$\begin{aligned} \mathbf{x}_{k|k-1}^a &= f\left(\mathbf{x}_{k-1|k-1}^a, \bar{\mathbf{u}}_k\right) = \begin{bmatrix} f_m\left(\mathbf{x}_{k-1|k-1}^m, \bar{\mathbf{u}}_k\right) \\ \mathbf{x}_{k-1|k-1}^\ell \end{bmatrix}, \\ P_{k|k-1}^a &= \nabla f_{\mathbf{x}_k^a} P_{k-1|k-1}^a \nabla f_{\mathbf{x}_k^a}^T + \nabla f_{\mathbf{u}_k} U_k \nabla f_{\mathbf{u}_k}^T + Q_k, \end{aligned} \quad (7)$$

with detailed definitions provided in **Appendix A**.

### B. EKF Update

Before updating the predicted state vector, the association between the predicted registered landmarks and the radar detections is conducted. To alleviate the computational burden of performing association on all of the measurements, a “sifting” step is used as in [7] to coarsely select *plausible* detections as candidate measurements. The candidate measurements are identified by calculating the Euclidean distances between the measurements and the existing landmarks. Any measurements which fall within a circle of pre-defined radius,  $\gamma_s$ , of an existing landmark are considered as a candidate measurement, as illustrated in **Fig. 2**.

The candidate measurement data identified during the sifting procedure is denoted as  $Z_k^s$ . This measurement data is then further refined using a stricter association step. This step is based on the log likelihood distance between a given

detection  $\mathbf{z}_k$  and landmark  $n$ :

$$D_n = -\log \Lambda \left( \mathbf{z}_k | \mathbf{x}_{k|k-1}^n, \mathbf{P}_{k-1}^n \right), \quad (8)$$

where the likelihood is defined in Appendix B. If  $D_n$  is less than a threshold,  $\beta$ , the measurement is associated with the landmark. Once a radar measurement is associated with a registered landmark, it is used to update the system state via the standard EKF update formula:

$$\begin{aligned} \mathbf{x}_{k|k}^{a,u} &= \mathbf{x}_{k|k-1}^a + W_k \left( \mathbf{z}_k - h_n \left( \mathbf{x}_{k|k-1}^a \right) \right), \\ P_{k|k}^{a,u} &= P_{k|k-1}^a - W_k S_k W_k^T, \end{aligned} \quad (9)$$

where the corresponding definitions and implementation are provided in Appendix B.

### C. Landmark Removal

After completing the update we next need to consider the fact that, in a slowly dynamic environment, there may be previously registered landmarks that have since changed their location. To address this problem we propose to use an  $M/N$  logic based scheme to identify and remove landmarks that no longer exist from the registered list. The rationale of this logic is that landmarks that currently exist, and are within the radar detection range, should produce consistent radar detections. Using a sliding window of length  $M$  we determine, based on the landmark and radar data association results from the EKF update step, if a registered landmark is associated with radar detections at least  $N$  times. If a landmark is not associated with radar data at least  $N$  times in the sliding window, and provided the registered landmark is still within the radar detection range, this landmark is identified for removal from the system state vector. Fig. 3 shows a detailed illustration of the landmark removal procedure using the  $M/N$  logic.

Note that the landmark removal procedure is conducted after the EKF update, hence, the number of landmarks immediately after the EKF update is the same as the number at the beginning of the EKF prediction. If there are landmarks to be removed, the corresponding entries in  $\mathbf{x}_{k|k}^{a,u}$  and  $P_{k|k}^{a,u}$  will be removed. This leads to a landmark reduction in the system, from  $N_{k-1}$  to  $N_k^r$  ( $N_k^r \leq N_{k-1}$ ) and correspondingly the cardinality of the system is  $3 + 2N_k^r$ . We denote the system state after landmark removal as  $\mathbf{x}_{k|k}^{a,r}$  and  $P_{k|k}^{a,r}$ .

### D. New Landmark Inclusion

After completing the removal of any landmarks that no longer exist, we next consider if there are any new landmarks to be included. There are two key issues regarding landmark inclusion: confirming it is a new landmark, and then registering this landmark in the system state. Intuitively, a new landmark must be far apart from the registered landmarks and, to avoid including any potential landmarks which may have originated from false radar detections, must produce consistent radar detections. To identify any potential new landmarks the first step is to cluster the radar detections,  $Z_k^r$ , which are those detections from  $Z_k$  which are not included

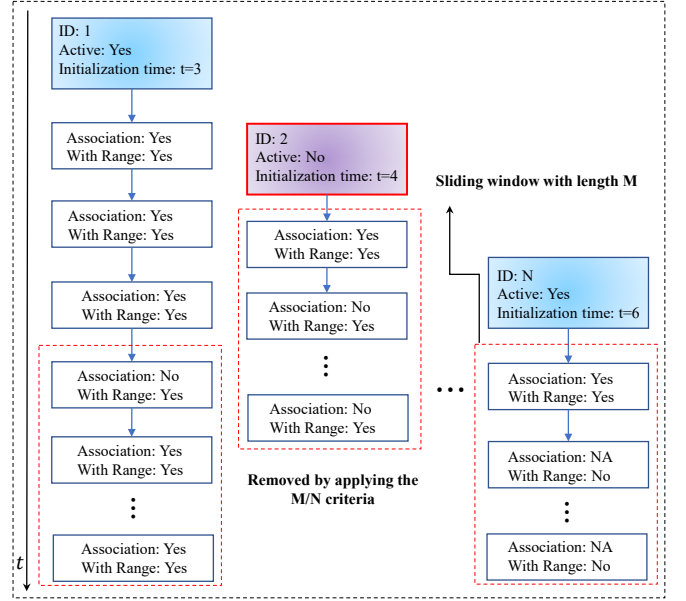


Fig. 3. Illustration of applying an  $M/N$  logic for landmark removal.

in  $Z_k^s$ , such that  $Z_k = \{Z_k^s, Z_k^r\}$ . Clustering is performed using DBSCAN [18], with the minimum number of detections required to form a cluster determined by the threshold  $N_{c2}$  and the maximum distance between two detections for them to be considered part of the same cluster determined by the threshold  $\gamma_c$ , as shown in Fig. 2. Identified clusters are represented by their centers, corresponding to the detection with the largest return, such that for the  $j^{\text{th}}$  cluster the center is  $\mathbf{c}_k^j = [r_k^j, \phi_k^j]$ . The clusters are then assessed to determine whether they should be confirmed as a new landmark and registered in the system.

We propose a rule based scheme for confirmation of clusters as new landmarks. First, we confirm that the cluster center is sufficiently far from any existing landmarks by calculating the log likelihood distance  $D$  defined in (8) of the cluster center with respect to the registered landmarks. If the distance exceeds a pre-defined threshold,  $\alpha$ , we progress to determining if it meets either of the following rules:

- **Rule 1** if the number of radar detections in a cluster exceeds threshold  $N_{c1}$ , it is identified as a landmark;
- **Rule 2** if the number of radar detections is less than  $N_{c1}$ , but it meets a multi-frame based criteria, as described below, it is identified as a landmark.

The multi-frame criteria in Rule 2 first associates current clusters with clusters from the previous time step based on the Euclidean distances between the cluster centers being less than the threshold  $\gamma_a$ . To be confirmed as a new landmark  $M/N$  logic is used, i.e., during  $M$  consecutive time steps, if there are at least  $N$  times that this cluster is detected/observed, it is confirmed as a new landmark. Fig. 4 further illustrates the rules applied for landmark confirmation.

Once a landmark is confirmed it needs to be initialized in

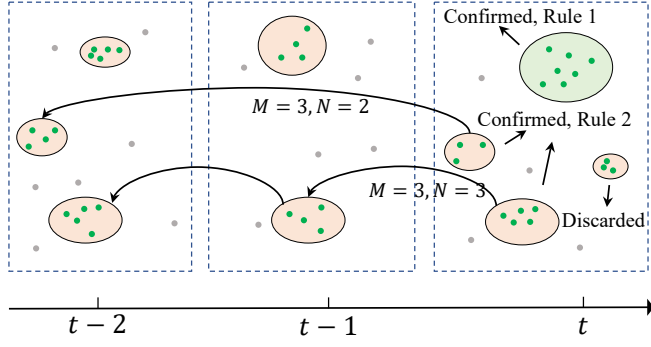


Fig. 4. Illustration of new landmark identification based on Rule 1 and Rule 2 using a 3/2 logic; note that the clustering results are from the remaining radar detections after sifting in Fig. 2.

the system. If there are  $N_k^\ell$  new landmarks to be initialized, then the 2D location of the  $j^{\text{th}}$  landmark ( $1 \leq j \leq N_k^\ell$ ) is  $\mathbf{p}_k^{N_k^r+j}$  and is concatenated to the end of the state vector to give the updated state vector

$$\mathbf{x}_{k|k}^{a,j} = \left[ \mathbf{x}_{k|k}^{a,r}, \underbrace{\mathbf{p}_k^{N_k^r+1}, \dots, \mathbf{p}_k^{N_k^r+j-1}}_{\mathbf{x}_{k|k}^{a,j-1}}, \mathbf{p}_k^{N_k^r+j} \right]^T. \quad (10)$$

The corresponding covariance matrix is then updated to give

$$P_{k|k}^{a,j} = J_1 P_{k|k}^{a,j-1} J_1^T + J_2 R_k J_2^T, \quad (11)$$

and derivations of (10) and (11) are provided in Appendix C [19]. At time  $k$  before landmark inclusion there are  $N_k^r$  landmarks, after landmark inclusion the total number of landmarks are  $N_k = N_k^r + N_k^\ell$  and the system state is denoted as  $\mathbf{x}_{k|k}^{a,N_k^\ell}$  and  $P_{k|k}^{a,N_k^\ell}$ .

#### E. Landmark Merging

Finally, the last step in our landmark management scheme is to address the possibility that two registered landmarks in the system may correspond to the same physical object<sup>1</sup>. We address this possibility by introducing a landmark merging logic step into our management scheme. The logic merges two landmarks if the Euclidean distance between the two is less than a threshold  $\gamma_m$ . The system state at the completion of landmark merging at time  $k$  is denoted as  $\mathbf{x}_{k|k}^a$  and  $P_{k|k}^a$  which completes the single iteration of Algorithm 1.

### IV. SIMULATION RESULTS

To assess the performance of our proposed landmark management scheme we run the EKF-SLAM algorithm in a simulated car park environment. The environment consists of several parked vehicles as shown in Fig. 5a. The cars are represented by rectangles and multiple radar detections are drawn uniformly from each rectangle. During the simulation the mobile platform moves along a designated trajectory, at

<sup>1</sup>Based on our observations, this usually occurs at the clustering step, when measurements originating from the same object (typically with a large extent) are clustered into more than one cluster.

TABLE I  
KEY PARAMETERS USED IN THE SIMULATION.

Parameter	Value
Radar detection range, $R_{\max}$	20m
Measurement noise covariance, $R_k$	$\begin{bmatrix} 0.5^2 & 0 \\ 0 & (1 \times \frac{\pi}{180})^2 \end{bmatrix}$
Process noise covariance, $Q_k$	$\begin{bmatrix} 1.5e^{-3} & 0 & 0 \\ 0 & 1.5e^{-3} & 0 \\ 0 & 0 & 5e^{-5} \end{bmatrix}$
Odometer noise covariance, $U_k$	$\begin{bmatrix} 0.02^2 & 0 \\ 0 & (0.008 \times \frac{\pi}{180})^2 \end{bmatrix}$
Threshold $\alpha$	500
Threshold $\beta$	20
Clustering threshold $\gamma_c$	2.5m
Sifting threshold $\gamma_s$	3m
Cluster association threshold $\gamma_a$	3.5m
Landmark merging threshold $\gamma_m$	1.5m
Minimum detection points $N_{c_1}$	6
Minimum clustering points $N_{c_2}$	2
M/N logic (landmark initialization)	5/3
M/N logic (landmark removal)	10/2

a speed of roughly 4 m/s for  $k = 120$  time steps with a time interval of  $\Delta t = 0.16$  seconds. The main parameters used in the algorithm are provided in Table I. We also provide sample Matlab code on github: [https://github.com/shuai000/SLAM\\_LandmarkManagement](https://github.com/shuai000/SLAM_LandmarkManagement).

Examples of several time points from a single simulation are shown in Fig. 5. Due to a relatively high clutter rate, at  $k = 14$  a landmark is mistakenly confirmed. As this landmark is not consistently associated with radar detections, by  $k = 25$  this false landmark has been removed by the M/N landmark removal logic. New landmarks are confirmed and registered in the state vector when they are within radar detection range, and meet the landmark confirmation rule. For example, at  $k = 25$  we can see there is a new landmark which has been confirmed in the system. At  $k = 40$ , one of the vehicles (registered in the state vector at  $k = 4$ ) drives out of scene. The landmark management scheme is not aware of this change until at  $k = 85$  the location of the “disappeared” vehicle is within the radar detection range. Since the expected detections from this registered landmark are not observed, this landmark is removed according to the M/N landmark removal logic.

Next, we statistically evaluate the performance of the proposed solution using Monte Carlo simulations. The simulation is designed with two scenarios, one with a low clutter rate and the other with a high clutter rate. The results for 100 Monte Carlo simulations are shown in Table II. The table lists the average root mean square error (RMSE) for

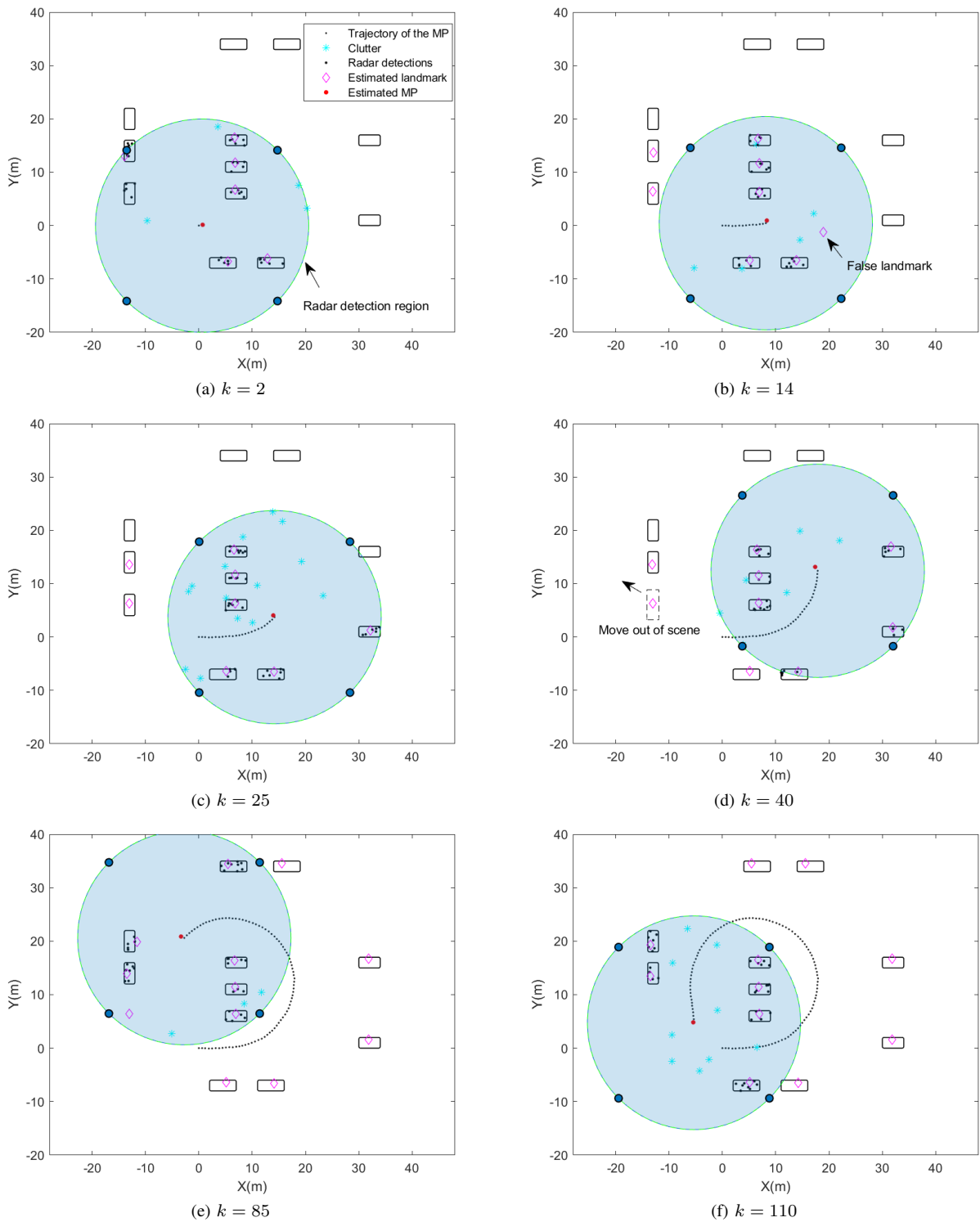


Fig. 5. Illustration of the results of the proposed landmark management scheme using the radar EKF-SLAM algorithm. The simulated environment equates to a car park with the rectangles representing the parked cars.

TABLE II  
MONTE CARLO SIMULATION RESULTS

Metric	Low Clutter	High Clutter
Platform position avg. RMSE (m)	0.81	0.90
Platform heading avg. RMSE (deg)	3.26	3.50
Landmark estimation MAE (m)	1.23	1.34
Landmark inclusion mean delay	2.45	3.22
Landmark removal mean delay	10.85	11.00
Mean (Max) false landmarks	0.13 (4)	3.02 (7)
Mean (Max) missed landmarks	0.2 (4)	0.23 (5)

both platform pose and landmark position estimation. For the landmark management, we provide the mean delay for new landmark inclusion and landmark removal. We also include the mean number of false and missed landmarks per simulation along with the maximum number of false and missed landmarks observed in a single simulation. Based on these results, we conclude that: 1) the proposed rule based landmark scheme is able to efficiently conduct landmark initialization, maintenance, removal and merging; 2) the rule based landmark management scheme can be easily incorporated into the existing EKF-based SLAM framework.

## V. CONCLUSIONS

In this paper we proposed a rule based landmark management scheme which can be easily incorporated into existing SLAM frameworks and was demonstrated using the EKF-SLAM algorithm. A detailed discussion of the implementation of the state prediction and update, as well as landmark removal and inclusion are provided. Simulation results show that the proposed solution can effectively manage the landmarks while the mobile platform is conducting a SLAM task. This potentially offers promising results for practical vehicular based applications such as autonomous driving.

## APPENDIX A EKF PREDICTION

For the prediction of the system state in the EKF, unlike the standard linear Kalman filter, the state vector (mean) is no longer independent of the state covariance. They are linked via the Jacobians  $\nabla f_{\mathbf{x}_k^a}$  and  $\nabla f_{\mathbf{u}_k}$  which are obtained via the partial differentiation of  $f$  w.r.t.  $\mathbf{x}^a$  and  $\mathbf{u}$ , taken at the point  $(\mathbf{x}_{k-1|k-1}^a, \mathbf{u}_k)$  giving

$$\nabla f_{\mathbf{x}_k^a} = \frac{\partial f}{\partial \mathbf{x}_k^a} \Big|_{(\mathbf{x}_{k-1|k-1}^a, \mathbf{u}_k)}$$

$$= \begin{bmatrix} 1 & 0 & -\Delta t v_k A & \mathbf{0}_{1 \times 2N_k} \\ 0 & 1 & \Delta t v_k B & \mathbf{0}_{1 \times 2N_k} \\ 0 & 0 & 1 & \mathbf{0}_{1 \times 2N_k} \\ \mathbf{0}_{2N_k \times 1} & \mathbf{0}_{2N_k \times 1} & \mathbf{0}_{2N_k \times 1} & I_{2N_k \times 2N_k} \end{bmatrix},$$

$$\nabla f_{\mathbf{u}_k} = \frac{\partial f}{\partial \mathbf{u}_k} \Big|_{(\mathbf{x}_{k-1|k-1}^m, \mathbf{u}_k)}$$

$$= \begin{bmatrix} \Delta t B & -\frac{\Delta t^2}{2} v_k A \\ \Delta t A & \frac{\Delta t^2}{2} v_k B \\ 0 & \Delta t \\ \mathbf{0}_{2N_k \times 1} & \mathbf{0}_{2N_k \times 1} \end{bmatrix},$$

where

$$A = \sin \left( \theta_{k-1|k-1} + \frac{\Delta t \psi_k}{2} \right),$$

$$B = \cos \left( \theta_{k-1|k-1} + \frac{\Delta t \psi_k}{2} \right).$$

Note that there is no dimension change during state prediction. The size of  $\nabla f_{\mathbf{x}^a}$  is  $N_{k-1}^a \times N_{k-1}^a$  where  $N_{k-1}^a$  is the size of the augmented state vector at time  $k-1$  and the size of  $\nabla f_{\mathbf{u}_k}$  is  $N_{k-1}^a \times 2$ .  $I_{2N_k}$  denotes the identity matrix with size  $2N_k \times 2N_k$ .

## APPENDIX B LIKELIHOOD & EKF UPDATE

The likelihood is defined as

$$\Lambda(\mathbf{z}_k | \mathbf{x}_{k|k-1}^m, \mathbf{p}_{k-1}^n) = \frac{1}{\sqrt{(2\pi)^2 |S_k|}} \exp \left( -\frac{1}{2} \mathbf{e}_{n,k}^T S_k^{-1} \mathbf{e}_{n,k} \right),$$

with  $\mathbf{e}_{n,k} = \mathbf{z}_k - h_n(\mathbf{x}_{k|k-1}^m)$ , where following the standard EKF update procedure,

$$S_k = \nabla h_{\mathbf{x}_k^n}^n P_{k|k-1}^a \nabla h_{\mathbf{x}_k^n}^n{}^T + R_k,$$

$$W_k = P_{k|k-1}^a \nabla h_{\mathbf{x}_k^n}^n{}^T S_k^{-1},$$

and the partial differentiation matrix is given by

$$\nabla h_{\mathbf{x}_k^n}^n = \frac{\partial h_n}{\partial \mathbf{x}_k^a} \Big|_{\mathbf{x}_{k|k-1}^a}$$

$$= \begin{bmatrix} -\frac{\Delta x}{r_n} & -\frac{\Delta y}{r_n} & 0 & \dots & \frac{\Delta x}{r_n} & \frac{\Delta y}{r_n} & \dots \\ \frac{\Delta y}{r_n^2} & -\frac{\Delta x}{r_n^2} & -1 & \dots & -\frac{\Delta y}{r_n^2} & \frac{\Delta x}{r_n^2} & \dots \end{bmatrix},$$

with

$$\Delta x = p_k^n(x) - x_{k|k-1}$$

$$\Delta y = p_k^n(y) - y_{k|k-1}$$

$$r_n = \sqrt{(\Delta x)^2 + (\Delta y)^2}.$$

## APPENDIX C LANDMARK INCLUSION

The augmented state can be expressed as a function of the original state and the cluster centers,

$$\mathbf{x}_{k|k}^{a,j} = g(\mathbf{x}_{k|k}^{a,j-1}, \mathbf{c}_k^j)$$

$$= \begin{bmatrix} \mathbf{x}_{k|k}^{a,j-1} \\ x_{k|k} + r_k^j \cos(\theta_{k|k} + \phi_k^j) \\ y_{k|k} + r_k^j \sin(\theta_{k|k} + \phi_k^j) \end{bmatrix},$$

with the Jacobin matrix defined as,

$$\begin{aligned}
 J_1 &= \frac{\partial g}{\partial \mathbf{x}^{a,j-1}} \Big|_{\mathbf{x}=\mathbf{x}_{k|k}^{a,j-1}, \mathbf{c}=\mathbf{c}_k^j} \\
 &= \begin{bmatrix} I_{N \times N} \\ 1 \quad 0 \quad -r_k^j \sin(\theta_k + \phi_k^j) \quad \mathbf{0}_{1 \times (N-3)} \\ 0 \quad 1 \quad r_k^j \cos(\theta_k + \phi_k^j) \quad \mathbf{0}_{1 \times (N-3)} \end{bmatrix}, \\
 J_2 &= \frac{\partial g}{\partial \mathbf{c}_k^j} \Big|_{\mathbf{x}=\mathbf{x}_{k|k}^{a,j-1}, \mathbf{c}=\mathbf{c}_k^j} \\
 &= \begin{bmatrix} \mathbf{0}_{N \times 1} & \mathbf{0}_{N \times 1} \\ \cos(\theta_k + \phi_k^j) & -r_k^j \sin(\theta_k + \phi_k^j) \\ \sin(\theta_k + \phi_k^j) & r_k^j \cos(\theta_k + \phi_k^j) \end{bmatrix}.
 \end{aligned}$$

#### REFERENCES

- [1] C. Cadena, L. Carlone, H. Carrillo, *et al.*, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016. DOI: [10.1109/TRO.2016.2624754](https://doi.org/10.1109/TRO.2016.2624754).
- [2] L. Li, M. Yang, C. Wang, and B. Wang, “Hybrid filtering framework based robust localization for industrial vehicles,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 941–950, 2017. DOI: [10.1109/TII.2017.2738016](https://doi.org/10.1109/TII.2017.2738016).
- [3] G. Dissanayake, S. B. Williams, H. Durrant-Whyte, and T. Bailey, “Map management for efficient simultaneous localization and mapping (SLAM),” *Autonomous Robots*, vol. 12, no. 3, pp. 267–286, 2002. DOI: [10.1023/A:1015217631658](https://doi.org/10.1023/A:1015217631658).
- [4] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007. DOI: [10.1109/TRO.2006.889486](https://doi.org/10.1109/TRO.2006.889486).
- [5] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017. DOI: [10.1109/TIV.2017.2749181](https://doi.org/10.1109/TIV.2017.2749181).
- [6] Z. Hong, Y. Petillot, A. Wallace, and S. Wang, “Radar SLAM: A robust slam system for all weather conditions.” arXiv: [2104.05347](https://arxiv.org/abs/2104.05347). (2021).
- [7] H. Lee, J. Chun, and K. Jeon, “Experimental results and posterior cramer-rao bound analysis of EKF-based radar SLAM with odometer bias compensation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 310–324, 2020. DOI: [10.1109/TAES.2020.3016873](https://doi.org/10.1109/TAES.2020.3016873).
- [8] T.-D. Vu, J. Burlet, and O. Aycard, “Grid-based localization and local mapping with moving object detection and tracking,” *Information Fusion*, vol. 12, no. 1, pp. 58–69, 2011. DOI: [10.1016/j.inffus.2010.01.004](https://doi.org/10.1016/j.inffus.2010.01.004).
- [9] C. S. Lee, D. E. Clark, and J. Salvi, “Slam with dynamic targets via single-cluster PHD filtering,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 543–552, 2013. DOI: [10.1109/JSTSP.2013.2251606](https://doi.org/10.1109/JSTSP.2013.2251606).
- [10] X. Dang, Z. Rong, and X. Liang, “Sensor fusion-based approach to eliminating moving objects for SLAM in dynamic environments,” *Sensors*, vol. 21, no. 1:230, 2021. DOI: [10.3390/s21010230](https://doi.org/10.3390/s21010230).
- [11] C.-C. Wang, C. Thorpe, and S. Thrun, “Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas,” in *IEEE International Conference on Robotics and Automation*, vol. 1, 2003, pp. 842–849. DOI: [10.1109/ROBOT.2003.1241698](https://doi.org/10.1109/ROBOT.2003.1241698).
- [12] J. E. Guivant and E. M. Nebot, “Optimization of the simultaneous localization and map-building algorithm for real-time implementation,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001. DOI: [10.1109/70.938382](https://doi.org/10.1109/70.938382).
- [13] K. Granström, S. Reuter, D. Meissner, and A. Scheel, “A multiple model PHD approach to tracking of cars under an assumed rectangular shape,” in *17th International Conference on Information Fusion (FUSION)*, 2014, pp. 1–8.
- [14] K. Granstrom, M. Baum, and S. Reuter, “Extended object tracking: Introduction, overview and applications,” *Journal of Advances in Information Fusion*, vol. 12, no. 2, pp. 139–174, 2017. arXiv: [1604.00970](https://arxiv.org/abs/1604.00970).
- [15] D. Vivet, P. Checchin, and R. Chapuis, “Localization and mapping using only a rotating FMCW radar sensor,” *Sensors*, vol. 13, no. 4, pp. 4527–4552, 2013. DOI: [10.3390/s130404527](https://doi.org/10.3390/s130404527).
- [16] M. Fazekas, P. Gáspár, and B. Németh, “Calibration and improvement of an odometry model with dynamic wheel and lateral dynamics integration,” *Sensors*, vol. 21, no. 2:337, 2021. DOI: [10.3390/s21020337](https://doi.org/10.3390/s21020337).
- [17] T. Bailey, “Mobile robot localisation and mapping in extensive outdoor environments,” Ph.D. dissertation, University of Sydney, 2002.
- [18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231. DOI: [10.1.1.121.9220](https://doi.org/10.1.1.121.9220).
- [19] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006. DOI: [10.1109/MRA.2006.1678144](https://doi.org/10.1109/MRA.2006.1678144).