

An Ensemble Kalman Filter for Feature-Based SLAM with Unknown Associations

Fabian Sigges, Christoph Rauterberg and Marcus Baum
Institute of Computer Science
University of Goettingen, Germany
Email: {fabian.sigges, marcus.baum}@cs.uni-goettingen.de
christoph.rauterberg@stud.uni-goettingen.de

Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
Email: uwe.hanebeck@ieee.org

Abstract—In this paper, we present a new approach for solving the SLAM problem using the Ensemble Kalman Filter (EnKF). In contrast to other Kalman filter based approaches, the EnKF uses a small set of ensemble members to represent the state, thus circumventing the computation of the large covariance matrix traditionally used with Kalman filters, making this approach a viable application in high-dimensional state spaces. Our approach adapts techniques from the geoscientific community such as localization to the SLAM problem domain as well as using the Optimal Subpattern Assignment (OSPA) metric for data association. We then compare the results of our algorithm with an extended Kalman filter (EKF) and FastSLAM, showing that our approach yields a more robust, accurate, and computationally less demanding solution than the EKF and similar results to FastSLAM.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) describes the problem of creating a map of an unknown environment while simultaneously tracking a robot’s position within the map. Tasks such as navigation in autonomous driving need a detailed map of the surroundings to provide an accurate localization. However, detailed knowledge about the current position is needed to build a map with newly discovered features, which makes the nature of the SLAM problem a “chicken-and-egg-problem”.

Combining the state of the robot and the map into a single state makes this application suitable for state estimation solutions. Traditional approaches use techniques such as the extended Kalman filter (EKF) [1] or particle filters [2]. However, as the discovered map grows, so does the number of observed landmarks and therefore the dimension of the state vector. As approaches such as the EKF need to keep track of an $n \times n$ covariance matrix, they fall short to the curse of dimensionality and memory constraints.

The ensemble Kalman filter (EnKF) however, developed in the geosciences, has been mostly unused in the signal processing community: Typical applications, such as weather prediction, are extremely high-dimensional state estimation problems, which produce states with orders of several millions and are therefore not tangible using traditional state estimation.

Instead of representing the state using mean and covariance matrix, the EnKF uses ensemble members to represent the state - not unlike the particles of a particle filter. In the update

step of the filter, the posterior is then obtained using the ensemble mean and an estimate of the sample covariance from the ensemble members.

Although being established in the geoscientific community, the EnKF has only recently received attention from the tracking community [3], [4]. In the case of SLAM, the EnKF is mostly just applied to specific parts of the whole problem. For example, in [5] the EnKF is used to generate a suitable proposal distribution for FastSLAM or in [6] the EnKF realizes the merging operation on two approximate transformations.

In the following, we derive an algorithm to solve the SLAM problem based on the ensemble Kalman filter. We implement and discuss several techniques for improving the performance of the EnKF that are common in the geoscientific community. Furthermore, we show our use of the Optimal Subpattern Assignment (OSPA) metric [7] to achieve reliable data association. We then present our results in comparison to an EKF and FastSLAM.

In the remainder of this paper, we will first introduce the EnKF idea as well as derive the nature and challenges of the SLAM problem in Section 2. We will then discuss our algorithm in detail in Section 3. The following section shows the results from our simulations. Section 5 will give our conclusions and an outlook on future work.

II. THE ENSEMBLE KALMAN FILTER

The ensemble Kalman filter can be seen as a Monte Carlo implementation of the traditional Kalman filter or even as a Kalman filter version of a particle filter: As the Kalman filter represents the state estimation at time step k by its first moments, the mean \hat{x}_k and covariance matrix Σ_k , the Kalman filter is not applicable to high-dimensional states as the computation and storage of a $n \times n$ covariance matrix becomes computationally very hard.

We begin with a closer look at the Kalman filter for linear systems.

The Kalman filter algorithm consists of two steps:

- 1) In the prediction-step, we use the process model equation to predict the next state

$$\tilde{x}_{k+1} = \mathbf{F}x_k + w_k \quad \text{with} \quad w_k \sim \mathcal{N}_n(0, \mathbf{Q}), \quad (1)$$

where x_k denotes the previous state, \tilde{x}_{k+1} the prediction, \mathbf{F} the state transition model and w_k the state noise. We assume w_k to be normally distributed with zero mean and covariance matrix \mathbf{Q} .

- 2) In the update step, we use information obtained in form of measurements to correct the prediction \tilde{x}_{k+1}

$$z_k = \mathbf{H}x_k + v_k \text{ with } v_k \sim \mathcal{N}_m(0, \mathbf{R}) , \quad (2)$$

$$\hat{x}_{k+1} = \tilde{x}_{k+1} + \mathbf{K}_{k+1}(z_{k+1} - \mathbf{H}\tilde{x}_{k+1}) , \quad (3)$$

where z_k denotes the m -dimensional measurement, \mathbf{H} the measurement model, v_k the measurement noise, and \mathbf{K}_{k+1} the Kalman gain. We assume v_k to be normally distributed with zero mean and covariance matrix \mathbf{R} . The Kalman gain is computed via

$$\mathbf{K}_k = \Sigma_k \mathbf{H}^T (\mathbf{H} \Sigma_k \mathbf{H}^T + \mathbf{R})^{-1} . \quad (4)$$

The general idea of the EnKF is to propagate a set of $N \ll n$ ensemble members of the ensemble $\mathcal{X} = \{x_k^{(1)}, \dots, x_k^{(N)}\}$ instead of the state \hat{x}_k and the covariance Σ_k , such that the mean of the ensemble members \bar{x}_k approximates the true state \hat{x}_k and the sample covariance $\bar{\Sigma}_k$ approximates the true covariance matrix Σ_k

$$\bar{x}_k = \frac{1}{N} \sum_{i=1}^N x_k^{(i)} \approx \hat{x}_k , \quad (5)$$

$$\bar{\Sigma}_k = \frac{1}{N-1} \sum_{i=1}^N (x_k^{(i)} - \bar{x}_k)(x_k^{(i)} - \bar{x}_k)^T \approx \Sigma_k . \quad (6)$$

Therefore, in the prediction-step of the Kalman filter we can simply propagate each ensemble member $x_k^{(1)}, \dots, x_k^{(N)}$ through the process model and in the update step we update each ensemble member with a simulated measurement

$$\tilde{x}_{k+1}^{(i)} = \mathbf{F}x_k^{(i)} + w_k^i \quad \forall i \in \{1, \dots, N\} , \quad (7)$$

$$\tilde{z}_{k+1}^{(i)} = z_{k+1} + v_{k+1}^i , \quad (8)$$

$$\hat{x}_{k+1}^{(i)} = \tilde{x}_{k+1}^{(i)} + \mathbf{K}_{k+1}(\tilde{z}_{k+1}^{(i)} - \mathbf{H}\tilde{x}_{k+1}^{(i)}) , \quad (9)$$

where an approximation of the Kalman gain is computed using the sample covariance $\bar{\Sigma}_k$

$$\mathbf{K}_{k+1} = \bar{\Sigma}_k \mathbf{H}^T (\mathbf{H} \bar{\Sigma}_k \mathbf{H}^T + \mathbf{R})^{-1} . \quad (10)$$

When dealing with non-linear relationships, we use a non-linear state transition function f to predict the state of each ensemble member and we use a non-linear measurement function h to transform each ensemble member into measurement space

$$\tilde{x}_{k+1}^{(i)} = f(x_k^{(i)}) + w_k^i \quad \forall i \in \{1, \dots, N\} , \quad (11)$$

$$\hat{x}_{k+1}^{(i)} = \tilde{x}_{k+1}^{(i)} + \mathbf{K}_{k+1}(\tilde{z}_{k+1}^{(i)} - h(\tilde{x}_{k+1}^{(i)})) . \quad (12)$$

However, as we want to circumvent the computation of a $n \times n$ covariance matrix or even its approximation in (10), we instead use the sample deviation \mathbf{X} and the measurement deviation \mathbf{Z}

$$\mathbf{X} = \frac{1}{N-1} \left[\tilde{x}_{k+1}^{(1)} - \bar{x}, \dots, \tilde{x}_{k+1}^{(N)} - \bar{x} \right] , \quad (13a)$$

$$\mathbf{Z} = \frac{1}{N-1} \left[\tilde{z}_{k+1}^{(1)} - \bar{z}, \dots, \tilde{z}_{k+1}^{(N)} - \bar{z} \right] , \quad (13b)$$

where

$$z_{k+1}^{(i)} = h(\tilde{x}_{k+1}^{(i)}) , \quad \text{and} \quad (14)$$

$$\bar{z} = \frac{1}{N} \sum_{i=1}^N h(\tilde{x}_{k+1}^{(i)}) . \quad (15)$$

Time subscripts have been omitted for simplicity. This means, we can now compute the Kalman gain as in [8]

$$\begin{aligned} \mathbf{K} &= \bar{\Sigma} \mathbf{H}^T (\mathbf{H} \bar{\Sigma} \mathbf{H}^T + \mathbf{R})^{-1} \\ &= (\mathbf{X} \mathbf{X}^T) \mathbf{H}^T (\mathbf{H} (\mathbf{X} \mathbf{X}^T) \mathbf{H} + \mathbf{R})^{-1} \\ &= \mathbf{X} (\mathbf{H} \mathbf{X})^T (\mathbf{H} \mathbf{X} (\mathbf{H} \mathbf{X})^T + \mathbf{R})^{-1} \\ &= \mathbf{X} \mathbf{Z}^T (\mathbf{Z} \mathbf{Z}^T + \mathbf{R})^{-1} . \end{aligned} \quad (16)$$

The distinct advantage here is that we do not need to compute a matrix of size $n \times n$: \mathbf{X} is of size $n \times N$ and, as we use range-bearing-measurements, \mathbf{Z} is of size $2 \times M$, with M the number of measurements. As $N \ll n$, this reduces the complexity of the Kalman gain computation immensely and sets the EnKF with $\mathcal{O}(NMn)$ between the EKF with $\mathcal{O}(n^2M)$ and FastSLAM with $\mathcal{O}(N \log n)$, where the advantage of FastSLAM stems from the special tree structure for the landmarks. However, EKF and EnKF process all measurements at once, while FastSLAM updates the measurements sequentially. Therefore, for M measurements FastSLAM requires $\mathcal{O}(NM \log n)$, which is still a bit faster than the EnKF approach.

III. ENKF-SLAM

A. Problem Formulation

SLAM describes the problem of simultaneously creating a map of the environment while localizing the robot in this map. To be able to achieve this at the same time, the state that we are estimating with the EnKF must represent the state of the robot as well as the entire map

$$x_k = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} .$$

Let the state of the robot \mathcal{R} be represented by its position in the Euclidean plane $\begin{bmatrix} r_x \\ r_y \end{bmatrix} \in \mathbb{R}^2$ and its heading $r_h \in [0, 2\pi]$. Let the map \mathcal{M} consist of the positions of the observed landmarks $L_{(j)} = \begin{bmatrix} L_{(j),x} \\ L_{(j),y} \end{bmatrix} \in \mathbb{R}^2$, $j = 1, \dots, |\text{landmarks}|$.

The idea to tackle the SLAM problem then is to append each newly observed landmark to the current state estimate, producing a large state vector.

When estimating this state using an EKF, in every update step we can use the measurement z_k to correct the prediction of the robot state \mathcal{R} and of the according landmark position $L_{(j)}$.

However, any solution using the EKF quickly becomes intractable due to the curse of dimensionality, which is why state-of-the-art approaches such as FastSLAM use multiple

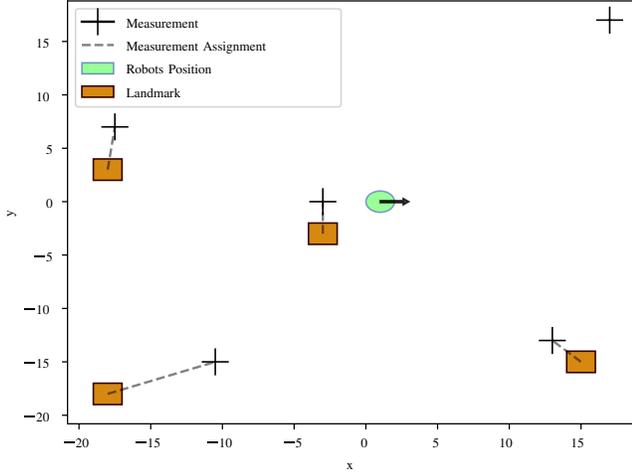


Fig. 1. When there are more measurements than observed landmarks, the OSPA metric is able to assign each measurement to a landmark whilst minimizing the distances. This will leave new measurements such in this example without associaton.

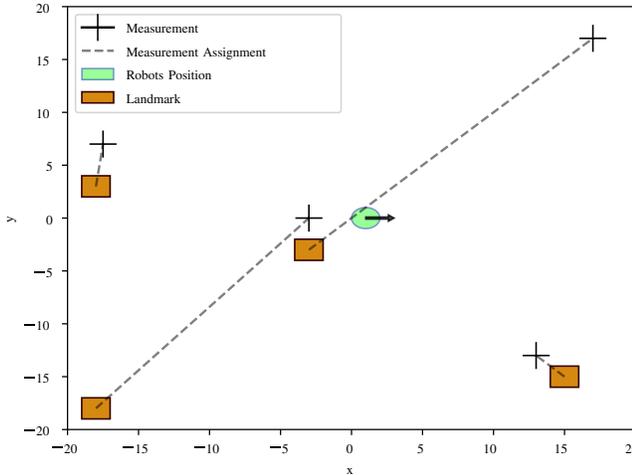


Fig. 2. However, once there are previously observed measurements, that where not observed with a measurement in this current time step, the OSPA metric will fail to recognize the new landmark and will assign it as a noisy measurement.

EKF's to track landmarks individually and use particle filters to track the robot's state [2].

The goal of our approach is to develop a robust, feasible approach to solving the SLAM problem using the EnKF.

B. Robot & World Model

To simulate an environment to which SLAM is applicable, we used a velocity-based motion system for the robot and range-bearing measurements:

- The control input for the state transition function f consists of the velocity v and the turning angle α relative to the robots heading.
- Measurements are given as range, i.e., Euclidean distance to the landmark, and bearing, i.e., angle relative to the robots heading r_h .

C. Challenges

Previous research has shown that the EnKF does not converge to the true solution for nonlinear cases in general [9]. Therefore we need to investigate whether our solution does converge to the correct result.

Furthermore, considering a linear case, the result does converge when $N \rightarrow \infty$. However, we want to focus on the case $N \ll n$, i.e., we only want to use a small number of ensemble members. One of the problems of the EnKF is its tendency to underestimate the covariance when using few ensemble members. As our covariance estimator $\bar{\Sigma}_k$ is the sample covariance, it can at most have rank $N - 1$. This leads to spurious correlations within $\bar{\Sigma}_k$. In addition to that, an underestimated covariance leads to an unrealistic confidence in the predicted state, so that the measurements incorporated in the update step will have insufficient impact on the update [10].

Typically, this is solved using covariance inflation and localization [11], [12]. The covariance can either be inflated using a constant factor ϑ or can be tapered using a localization matrix τ , and define $\Sigma_{tap} := \tau \circ \Sigma$, where \circ is the entry-wise or Hadamard product. The general idea is to eliminate spurious correlations in Σ based on the distance between two observations (landmarks far away from each other should not be correlated). However, as above, we want to avoid computing $n \times n$ matrices. As the Hadamard product is not distributive under multiplication, we cannot use this approach to localization in (16). Other approaches of directly manipulating the estimation of the covariance matrix, for example, leaving out landmarks that are far away from each other only seemed to deteriorate the results, at least for the world sizes we used.

Covariance inflation however can be used directly. To inflate the uncertainty, we sample the initial ensemble distribution from $x_0^{(i)} \sim \mathcal{N}(0, \Sigma_{init})$ and, in the prediction step, add white noise $\rho_{L_j}^{(i)} \sim \mathcal{N}(0, \rho)$ to the j -th landmark of the i -th ensemble member. Currently, we choose these parameters by hand, depending on measurement and control noise. For more realistic scenarios, with unclear or changing noises it might also be handy, to estimate the parameters for inflation, along with the state as described in [13].

Thereby we artificially inflate the uncertainty during the prediction so that the impact of each measurement z_k is meaningful.

D. Data Association

As the EnKF comes with no tools to achieve data association from noisy measurements, we will adapt the Optimal Subpattern Assignment (OSPA) metric [7] to our algorithm.

Algorithm 1: The prediction step of EnKF-SLAM.

```
Data:  $u = (v, \alpha)$ ,  $\mathbf{Q}$ ,  $\rho$ 
1 for  $\hat{x}_k^{(i)} \in \mathcal{X}$  do
2   | Draw  $w_{k+1,i} \sim \left| \begin{matrix} \mathbf{Q} \\ \rho \end{matrix} \right|$ 
3   |  $\tilde{x}_{k+1}^{(i)} = f(\hat{x}_k^{(i)}, w_{k+1,i})$ 
4 end
5  $\bar{x}_{k|k+1} = \text{mean}(\mathcal{X})$ 
```

This is similar to our previous work [14], where an EnKF and the OSPA metric were used to better match ensemble members to their according tracks. It can be used to find an optimal “global nearest neighbor” association between measurements and landmarks, as it aims to find the permutation over the unassociated measurements that minimizes

$$\mathbf{d}(\tilde{x}_k^{(i)}, z_k) = \min_{\pi \in \Pi_M} \sum_{j=1}^L \|h(\tilde{x}_k^{(i)}, j) - z_{k,\pi(j)}\|_2. \quad (17)$$

Here, Π_M denotes all permutations of lengths M , the number of measurements. Computationally speaking, we can set up a matrix $\mathbf{D}_{j,m}$, where $\mathbf{D}_{j,m} = \|h(\tilde{x}_k^{(i)}, j) - z_{k,m}\|_2$ is the distance between the j -th landmark and the m -th measurement and find the optimal permutation using the Hungarian method.

However, this metric is only partially applicable to the SLAM problem: It is very robust in finding the optimal matching between measurements and previously observed landmarks - yet the metric is not able to detect new measurements.

Fig. 1 and Fig. 2 highlights this problem: Once a measurement for a new landmark is recorded and there are previously observed landmarks that were not observed via a measurement, e.g., through sensor range, the OSPA metric will try to find an assignment that minimizes all distances, as can be seen in Fig. 2. The newly discovered landmark will be assigned to the closest previously discovered landmark.

To detect new landmarks, we use the Mahalanobis distance

$$D_m(h(\tilde{x}_k^{(i)}, j), z_k) = \sqrt{(h(\tilde{x}_k^{(i)}, j) - z_k)^T \mathbf{R}^{-1} (h(\tilde{x}_k^{(i)}, j) - z_k)}, \quad (18)$$

where \mathbf{R} denotes measurement noise covariance matrix.

The result $D_m(h(\tilde{x}_k^{(i)}, j), z_k)$ gives us the distance between a measurement and the expected measurement of an ensemble member with respect to the covariance matrix \mathbf{R} .

As we assume all of our noise to be Gaussian distributed around the true value, i.e., the true position of the landmark, by the given covariance, we can now assume a confidence interval: Under the Chi-squared distribution we assume a 95% confidence interval around the true value, i.e., if for a measurement z_k

$$D_m(h(\tilde{x}_k^{(i)}, j), z_k) > 5.991 \quad \forall j,$$

Algorithm 2: The update step of EnKF-SLAM

```
Data:  $z_k$ ,  $\mathbf{R}$ ,  $\gamma$ 
1  $new\_z_k, obs\_z_k = \text{assign\_measurements}(z_k, \mathcal{X})$ 
2 for  $z_i \in new\_z_k$  do
3   | for  $\tilde{x}_{k+1}^{(i)} \in \tilde{\mathcal{X}}$  do
4     | Add landmark to  $\tilde{x}_{k+1}^{(i)}$  via  $z_i$ 
5   | end
6 end
7 for  $z_i \in obs\_z_k$  do
8   |  $\epsilon = \text{findNearbyLandmarks}(\gamma)$ 
9   | Compute  $\mathbf{X}$  and  $\mathbf{Y}$  using  $\epsilon$ 
10  | Compute  $\mathbf{K} = \mathbf{X}\mathbf{Y}^T(\mathbf{Y}\mathbf{Y}^T + \mathbf{R})^{-1}$ 
11  | for  $\tilde{x}_{k+1}^{(i)} \in \tilde{\mathcal{X}}$  do
12    |  $\hat{x}_{k+1}^{(i)} = \tilde{x}_{k+1}^{(i)} + \mathbf{K}(z_i - h(\tilde{x}_{k+1}^{(i)}))$ 
13  | end
14 end
15  $\bar{x}_{k+1} = \text{mean}(\mathcal{X})$ 
```

it will be declared a new measurement, as it is not in the confidence interval around any landmark. This is similar to other common gating techniques.

If now $\frac{3N}{4}$ ensemble members label a measurement as new, our algorithm will treat it as a new measurement.

After identifying the measurements from newly discovered landmarks, we can use (17) to compute the optimal matching between the remaining measurements and the previously discovered landmarks.

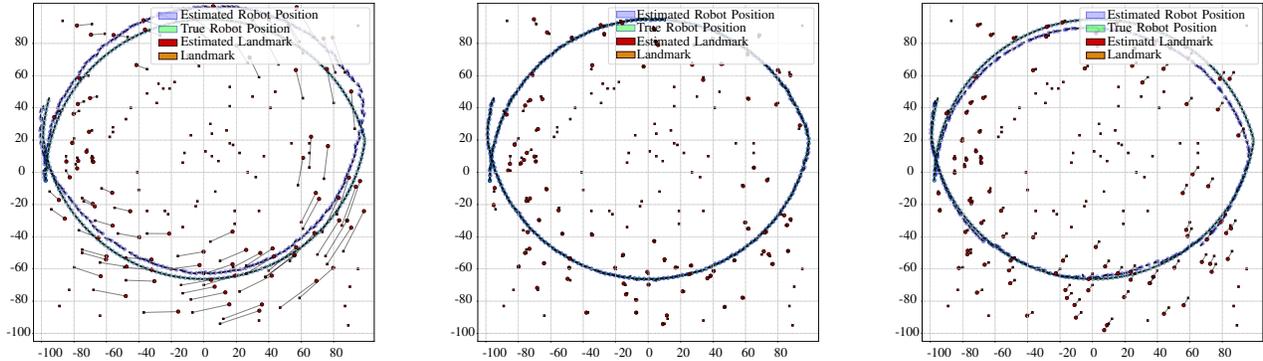
This might lead to extreme outliers being detected as new landmarks. To avoid this problem, we will introduce the cut-off parameter γ : When outliers, i.e., extremely noisy measurements, are detected as new landmarks, we will observe few measurements that correspond to this new landmark, even though it might be in the sensor range. Therefore we will remove landmarks that have not been seen since γ time steps although they should have been observed. In a similar way, false detections can be handled up to a certain degree.

E. EnKF-SLAM

The resulting prediction and update algorithms are described in Alg. 1 and Alg. 2 respectively.

The prediction step for the EnKF is easy to implement and furthermore easy to parallelize. It is important to note the addition of the white noise vector ρ to further inflate the uncertainty, as \mathbf{Q} only refers to the information in the state vector concerning the robot, i.e., \mathcal{R} .

In the update step, we use the procedure for data association introduced in Sec. III-D. After successful association, all new landmarks are linked to the state vector. As the measurements z_i are range-bearing measurements, the position of the landmark must be computed relative to the individual position of each ensemble member. This directly introduces an uncertainty for the new landmarks.



(a) The results for solving SLAM using an EKF (b) The results of our algorithm using an EnKF (c) The results using FastSLAM 2.0

Fig. 5. Our simulation depicts a circular walk through an environment with randomly placed landmarks. Overall, there are $|l| = 150$ landmarks in this world. Noisy robot controls are given to the filters in the prediction-step and noisy measurements are given in the update-step. We implied a sensor range of $r = 30$ Euclidean units, wherefore not all landmarks have yet been observed. We used $N = 75$ ensemble members for EnKF-SLAM and $N = 75$ particles for FastSLAM 2.0. We can clearly see, that the EnKF-SLAM outperforms the EKF in accuracy for the robots position as well as in terms of landmark predictions and shows similar results to those of FastSLAM 2.0.

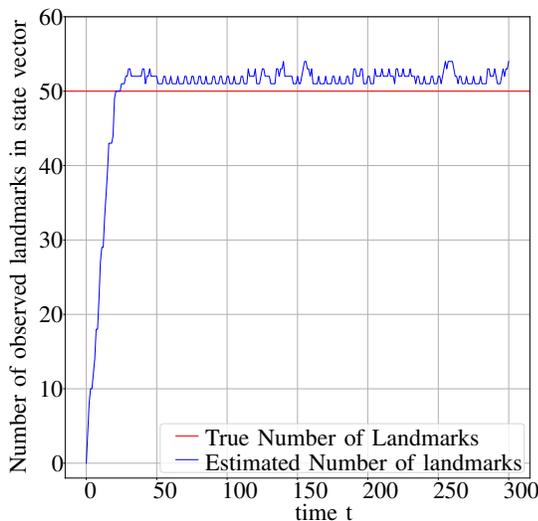


Fig. 6. Visualization of the state vector size for a random walk of 200 steps in a world with 50 landmarks. Each $t = 10$ steps, a random, wrong measurement is generated in addition to the normally observed measurements. When beginning walking, the robot has not yet found all landmarks, wherefore measurements are identified as new landmarks. Later on, the false measurements will first be noted as a new landmark but will be discarded after γ steps, as there have been no further observations. This shows the robustness of our approach to false measurements as well as loop closing.

After finding nearby landmarks, we use the approximation of Σ and \mathbf{K} shown in (16).

IV. EVALUATION

To evaluate the performance of our approach, we set up a simulation in a square part of the Euclidean plane. The control inputs u_k used to move the robot as well as the

recorded measurements z_k at each time step k are given to the algorithms with added Gaussian noise.

Both approaches, EKF and EnKF, use the presented algorithm for data association introduced in Sec. III-D. Additionally, we used the FastSLAM 2.0 implementation [15] for comparison.

Fig. 5 shows the result for a circular walk in our simulation. We use $N = 75$ ensemble members and $|l| = 150$ landmarks. The state transition noise and the measurement noise are defined as

$$\mathbf{Q} = \begin{bmatrix} 1.0 & 0. & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.01 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 0.01 \end{bmatrix}.$$

We simulate a sensor with a certain range $r_{max} = 30$, i.e., only landmarks within this range of the robot produce measurements.

Fig. 5a, Fig. 5b, and Fig. 5c show the results of EKF-SLAM, EnKF-SLAM, and FastSLAM 2.0, respectively. Due to the limited number of observations and the linearization, EKF-SLAM produces high errors while EnKF-SLAM and FastSLAM 2.0 achieve similar, better, results. Due to the limited sensor range, some landmarks in the middle of the circle have not been observed by the robot.

Fig. 7 depicts the average error of the landmark predictions for all three algorithm over 100 runs with the same parameters as above. The variance for the single runs is very large for the EKF and similar for EnKF and FastSLAM 2.0. Over all runs, our EnKF-SLAM shows a similar performance as FastSLAM2.0.

However, the performance is directly linked to the noise incorporated into the system. Tab. I and Tab. II depict the average error on the landmark prediction for the three algorithms w.r.t. to different covariances \mathbf{Q} and \mathbf{R} . Here, we see an

TABLE I

AN OVERVIEW OVER THE PERFORMANCE OF EKF-SLAM, ENKF-SLAM AND FASTSLAM IN A SIMULATION WITH $L = 200$ LANDMARKS AND 600 STEPS. FOR EACH ALGORITHM, THE MSE $\varepsilon_{\mathcal{R}}$ W.R.T. THE ROBOTS TRUE POSITION AND THE MSE $\varepsilon_{\mathcal{L}}$ OVER ALL LANDMARK ESTIMATES IS LISTED. THE RESULTS ARE AVERAGED OVER A TOTAL OF $K = 50$ SIMULATIONS. THE CONTROL NOISE IS A FIXED $\sigma_v = 0.1$ FOR THE VELOCITY AND $\sigma_\alpha = 0.001$ FOR THE TURNING ANGLE. THE NOISE FOR THE MEASURED RANGE σ_r AND FOR THE MEASURED BEARING σ_γ IS INCREASED.

Q	σ_v	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	σ_α	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
R	σ_r	0.1	0.25	0.5	0.75	1.0	1.25	1.5	1.75
	σ_γ	0.001	0.0025	0.005	0.0075	0.01	0.0125	0.015	0.0175
EKF-SLAM $\varepsilon_{\mathcal{P}}$		0.261	1.893	1.438	5.239	5.530	6.172	6.459	7.354
EKF-SLAM $\varepsilon_{\mathcal{L}}$		0.282	1.925	1.623	5.055	5.264	5.41	5.632	6.091
EnKF-SLAM $\varepsilon_{\mathcal{P}}$		0.239	0.786	1.046	1.637	1.747	2.53	3.563	4.109
EnKF-SLAM $\varepsilon_{\mathcal{L}}$		0.250	0.646	0.911	1.623	2.112	2.46	3.810	4.89
FastSLAM $\varepsilon_{\mathcal{P}}$		0.197	0.536	1.036	1.419	1.881	1.662	2.032	3.017
FastSLAM $\varepsilon_{\mathcal{L}}$		0.122	0.727	0.833	1.233	1.696	1.592	2.862	3.642

TABLE II

AN OVERVIEW OVER THE PERFORMANCE OF EKF-SLAM, ENKF-SLAM AND FASTSLAM IN A SIMULATION WITH $L = 200$ LANDMARKS AND 600 STEPS. FOR EACH ALGORITHM, THE MSE $\varepsilon_{\mathcal{R}}$ W.R.T. THE ROBOTS TRUE POSITION AND THE MSE $\varepsilon_{\mathcal{L}}$ OVER ALL LANDMARK ESTIMATES IS LISTED. THE RESULTS ARE AVERAGED OVER A TOTAL OF 50 SIMULATIONS. THE MEASUREMENT NOISE IS A FIXED $\sigma_r = 0.1$ FOR THE RANGE AND $\sigma_\gamma = 0.001$ FOR THE BEARING. THE NOISE FOR THE VELOCITY σ_v AND FOR THE TURNING ANGLE σ_α IS INCREASED.

Q	σ_v	0.1	0.25	0.5	0.75	1.0	1.25	1.5	1.75
	σ_α	0.001	0.0025	0.005	0.0075	0.01	0.0125	0.015	0.0175
R	σ_r	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	σ_γ	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
EKF-SLAM $\varepsilon_{\mathcal{P}}$		0.261	1.390	5.251	6.817	9.812	11.234	15.345	15.211
EKF-SLAM $\varepsilon_{\mathcal{L}}$		0.282	1.339	5.160	6.828	9.810	13.132	15.664	18.233
EnKF-SLAM $\varepsilon_{\mathcal{P}}$		0.239	0.505	0.387	1.507	2.360	3.212	5.983	6.129
EnKF-SLAM $\varepsilon_{\mathcal{L}}$		0.250	0.417	0.332	1.367	2.313	2.931	4.325	6.745
FastSLAM $\varepsilon_{\mathcal{P}}$		0.197	0.485	1.295	3.330	4.859	5.477	6.567	8.001
FastSLAM $\varepsilon_{\mathcal{L}}$		0.122	0.337	1.289	3.279	5.284	5.456	7.312	8.120

increase in the error related to increased noise. While the EKF-SLAM performance suffers strongly, both FastSLAM 2.0 and our EnKF-SLAM sustain a smaller increase in the observed error.

In Tab. I, the control noise is fixed and the measurement noise is increased in each scenario. In this case, FastSLAM 2.0 outperforms the EnKF by a small margin. Results for the switched scenario, with fixed measurement noise and an increasing control noise, can be seen in Tab. II. Here, the EnKF produces notably smaller errors. This might be a consequence of the assumption that FastSLAM makes on the path. In order for the landmarks to be uncorrelated, it requires good knowledge of the robot path. Hence, estimates become worse for growing control noises faster than for the EnKF, which makes no assumption on the robot path.

Furthermore, Fig. 6 highlights the performance of our data association. In an environment with 50 landmarks, we perform a random walk of 200 steps. In addition to the already noisy

measurements, we add a false measurement every 5 steps. The figure shows the total amount of observed features in the state vector: In the beginning, more and more features are found. Yet later, when converging to the true number of landmarks, incoming measurements are correctly associated with the already observed landmarks. Extremely noisy measurements or false measurements lead to the temporary discovery of a new landmark which are dropped after γ steps due to no more incoming measurements. Although we add false measurements every five steps, the total amount of landmarks in the state vector hardly rises over 55. This shows that our data association approach is capable of dealing with false measurements. Additionally, this also tackles the problem of loop closure: Through our gating approach, incoming measurements are first matched against previously discovered measurements and we are therefore able to detect a closed loop.

ACKNOWLEDGEMENTS

This work was partially supported by the Simulation Science Center Clausthal-Göttingen.

REFERENCES

- [1] S. Huang and G. Dissanayake, "Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, Oct. 2007.
- [2] M. Montemerlo and S. Thrun, "Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA 2003)*, Taipei, Taiwan, Sept. 2003.
- [3] M. Roth, C. Fritsche, G. Hendeby, and F. Gustafsson, "The Ensemble Kalman Filter and its Relations to Other Nonlinear Filters," in *Proceedings of the 2015 European Signal Processing Conference (EUSIPCO 2015)*, ser. European Signal Processing Conference. Institute of Electrical and Electronics Engineers (IEEE), 2015, pp. 1236–1240.
- [4] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, "The Ensemble Kalman filter: a signal processing perspective," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, p. 56, Aug. 2017.
- [5] J. Wang and Z. Liu, "Improved FastSLAM based on EnKF proposal distribution for AUV," in *Proceedings of the 2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017)*, vol. 74, July 2017.
- [6] J. Čurn, D. Marinescu, and V. Cahill, "On the Monte Carlo Representation of Uncertain Spatial Constraints," *Procedia Engineering*, vol. 41, pp. 37–46, 2012.
- [7] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A Consistent Metric for Performance Evaluation of Multi-Object Filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.
- [8] Y. Tang, J. Ambandan, and D. Chen, "Nonlinear measurement function in the ensemble Kalman filter," *Advances in Atmospheric Sciences*, vol. 31, no. 3, pp. 551–558, May 2014.
- [9] F. Le Gland, V. Monbet, and V.-D. Tran, "Large sample asymptotics for the ensemble Kalman filter," INRIA, Research Report RR-7014, 2009. [Online]. Available: <https://hal.inria.fr/inria-00409060>
- [10] G. Evensen, "The Ensemble Kalman Filter: Theoretical Formulation and Practical Implementation," *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 2003. [Online]. Available: <http://dx.doi.org/10.1007/s10236-003-0036-9>
- [11] J. L. Anderson, "Localization and Sampling Error Correction in Ensemble Kalman Filter Data Assimilation," *Monthly Weather Review*, vol. 140, no. 7, pp. 2359–2371, 2012.
- [12] K. Bergemann and S. Reich, "A localization technique for ensemble Kalman filters," *Quarterly Journal of the Royal Meteorological Society*, vol. 136, no. 648, pp. 701–707, 2010.
- [13] J. L. Anderson, "Ensemble Kalman filters for large geophysical applications," *IEEE Control Systems Magazine*, vol. 29, no. 3, pp. 66–82, June 2009.
- [14] F. Siggés and M. Baum, "A Nearest Neighbour Ensemble Kalman Filter for Multi-Object Tracking," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Daegu, South Korea, Nov. 2017, pp. 227–232.
- [15] T. Bailey, "Slam simulations," http://www-personal.acfr.usyd.edu.au/tbailey/software/slam_simulations.htm, accessed: 2010-09-30.

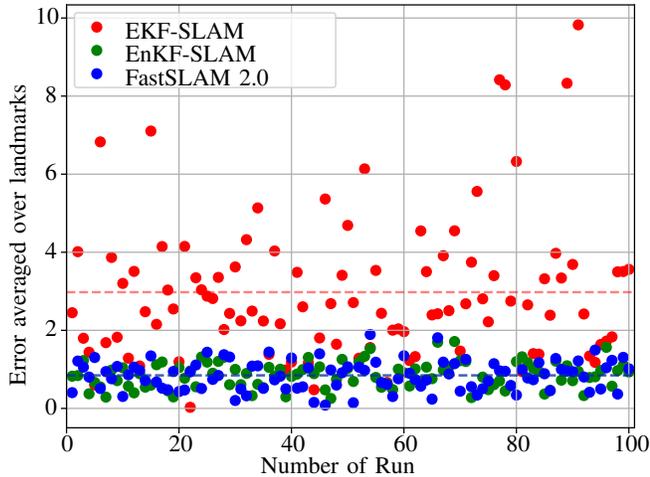


Fig. 7. The average error of all three approaches sampled over 100 runs. We used $|l| = 150$ randomly placed landmarks and a sensor range of 30. While the EnKF-SLAM and FastSLAM 2.0 show an almost constant error, the EKF-SLAM shows a wider error range.

V. CONCLUSION & OUTLOOK

Our work has shown that the Ensemble Kalman Filter serves as a viable approach for tackling the SLAM problem. Adapting various techniques from other research focused on the improvement of the EnKF itself. We were able to improve the performance of the filter in our conducted simulations. The presented solution outperforms the traditional algorithm using an EKF and achieves similar, and in some cases even better results, to those of the state-of-the-art algorithm FastSLAM 2.0.

Our approach for data association for the Ensemble Kalman Filter in the context of SLAM has shown to be reliable and also robust to false measurements as well as being able to deal with loop closure.

We would like to focus future work on a detailed analysis on different possible scenarios and evaluate the performance of FastSLAM 2.0 and our EnKF-SLAM w.r.t. the specified noise to provide a detailed overview of the strength and weaknesses of both approaches.