Distributed EKF SLAM System with Known Correspondence

Trento, Italy
nicolo.cavalieri@studenti.unitn.it (E. N. 231202)

Trento, Italy
federico.burgio@studenti.unitn.it (E. N. 233177)

***Abstract*—This report presents the design and simulation of a distributed Extended Kalman Filter (EKF) Simultaneous Localization and Mapping (SLAM) system with known correspondence for robotic agents. The objective of the system is to enable multiple robots to collaboratively build a map of an environment while simultaneously localizing themselves within that map. The map consists of a set of landmarks that can be uniquely identified by the robots. The simulation was conducted using Matlab, a widely-used software tool for scientific and engineering applications.**

I. INTRODUCTION

In the field of robotics, SLAM is a fundamental problem that involves constructing a map of an unknown environment while simultaneously estimating the robot's pose within that map. SLAM plays a crucial role in various applications, including autonomous navigation, exploration, and collaborative robotics. The ability to accurately and efficiently perform SLAM is essential for enabling robots to operate in complex and dynamic environments.

Traditionally, SLAM algorithms have been developed for single robots operating in isolation. However, recent advancements in robotics have led to the proliferation of multi-robot systems, where multiple robots collaborate to achieve a common goal. Distributed SLAM, which involves multiple robots collaboratively building a map and localizing themselves within that map, has gained significant attention due to its potential for improved mapping accuracy, robustness, and scalability.

This report focuses on the design and simulation of a distributed EKF SLAM system with known correspondence for multiple robotic agents. The objective is to enable collaborative mapping and localization in complex environments. The system assumes that the robots can uniquely identify a set of landmarks in the environment, facilitating data association and correspondence between the robots. By leveraging the EKF framework, the system can estimate the robots' poses and update the map iteratively.

The motivation for developing a distributed EKF SLAM system lies in the advantages it offers. Collaboration among robots allows for the efficient exploration of the environment and faster map construction compared to individual robots working in isolation [1]. Furthermore, by sharing information and leveraging the known correspondence of landmarks, the system can enhance the accuracy of the individual robots' pose estimates and reduce uncertainty.

To achieve the objectives of the distributed EKF SLAM system, various components need to be considered, including communication systems, system models, control laws, and estimators. The implementation of the system simulation is carried out using Matlab, a widely-used software tool for scientific and engineering applications. The simulation provides a platform to evaluate the system's performance and analyze its behavior in different scenarios.

In the subsequent sections of this report, we will delve into the details of the adopted models, present the proposed solution, discuss implementation details, showcase experimental results, and provide concluding remarks on the benefits, limitations, and potential future directions of the distributed EKF SLAM system.

A. Problem Formulation

The problem addressed in this work is to develop a distributed EKF SLAM system with known correspondence for multiple robotic agents. The system aims to enable collaborative mapping and localization in complex environments by leveraging the capabilities of a team of robots.

The main objectives of the system can be summarized as follows:

1) **Collaborative Mapping**: The system should facilitate the construction of a map of the environment by combining the sensor measurements and robot poses from multiple robots. The map consists of a set of landmarks that can be uniquely identified by the robots. By collaboratively mapping the environment, the system aims to achieve faster and more accurate mapping results compared to individual robots working independently.
2) **Simultaneous Localization**: The system should estimate the poses of the robots within the constructed map. Each robot's pose includes its position (x, y coordinates) and orientation (heading angle). By estimating the poses of the robots simultaneously with map construction, the system enables the robots to localize themselves within the shared map.
3) **Distributed Collaboration**: The system should enable efficient collaboration among the robots while minimizing the communication and computational requirements. The robots should exchange relevant information, such as sensor measurements, robot poses, and map updates, to collectively build an accurate and consistent map. The system should leverage the known correspondence of

landmarks to associate data from different robots and enhance the accuracy of individual robot poses.

To achieve these objectives, the system adopts the EKF framework, which is a widely-used method for state estimation in SLAM problems. The EKF maintains a belief distribution over the robot poses and map states and iteratively updates the estimates based on sensor measurements and control inputs.

The problem formulation involves designing the communication system, defining the system model that includes robot dynamics, sensors, and actuators, and developing control laws and estimators that facilitate collaborative mapping and localization. The implementation of the system is carried out in Matlab, providing a platform for simulation and evaluation of the proposed approach.

B. Notation

Throughout the report, bold lower-case letters ($\mathbf{x}$) represent vectors and bold upper-case letters ($\mathbf{H}$) represent matrices. Scalar will be represented by light lower-case letters (t), functions by light upper-case letters (I).

II. Adopted Models

In this section, we present the adopted models for the communication system and the system itself, including the robot dynamics, sensors, and actuators.

A. Communication System

The communication system plays a crucial role in enabling collaboration and information exchange among the robotic agents. In the implemented distributed EKF SLAM system, a message-passing approach is utilized for communication. Each robot acts as a node in the network and can send and receive messages to/from other robots.

The communication system is implemented in simulation using a message broker server, which serves as a central hub for message routing and distribution. The server receives messages from the robots and forwards them to the intended recipients. It also stores the messages in an inbox, allowing robots to retrieve and process them asynchronously.

The adopted communication system allows the robots to exchange various types of messages, including sensor measurements, robot poses, and map updates. These messages are essential for collaborative mapping and localization, as they enable the robots to share information and enhance their individual estimates.

B. System Model

The system model encompasses the dynamics, sensors, and actuators of the robotic agents involved in the distributed EKF SLAM system. Our system consists of a fleet of land robots equipped with cameras capable of identifying landmarks and measuring range and bearing to them.

Each robot is equipped with a monocular camera that captures images of the surrounding environment. By processing these images, the robots can detect unique barcodes attached to cylindrical landmarks. These barcodes encode unique identification numbers, allowing the robots to identify and differentiate between landmarks.

Using the information extracted from the barcodes, the robots are able to measure the range and bearing to each landmark. The range measurement represents the distance from the robot to the landmark, while the bearing measurement indicates the direction or angle between the robot's heading and the landmark.

These range and bearing measurements are crucial for mapping and localization within the SLAM system. They provide valuable information about the relative positions of the robots and the landmarks in the environment.

By leveraging these measurements from multiple robots, the distributed EKF SLAM system aims to collaboratively build a global map of the environment. The robots exchange information about their local maps and use consensus mechanisms to merge and refine the maps into a coherent representation of the environment.

In the next section, we will discuss the proposed solution in detail, including the control laws and estimators employed to facilitate collaborative mapping and localization within the system.

III. Solution

The proposed solution makes several key assumptions to address the problem of distributed SLAM for multiple robotic agents. These assumptions serve as foundational principles in developing the system:

1. **Plane Operation**: The robotic agents are assumed to operate in a two-dimensional plane. This implies that the state of each robot, denoted as $\mathbf{x}$, represents the robot's position (x and y) and orientation (ϑ).

2. **Range and Bearing Observations**: The measurements obtained from sensors are expressed using a range and bearing observation model. This model provides information about the distance and angle between the robot and landmarks in the environment.

3. **Velocity-based Motion Model**: The controls or inputs provided to the robots are expressed using a velocity-based motion model. This model incorporates the robot's velocity and angular velocity to estimate its next state.

4. **Known Number of Landmarks**: The number of landmarks in the environment is assumed to be known a priori. This knowledge allows for efficient data association and correspondence between the robots.

A. EKF SLAM

The EKF SLAM algorithm forms the basis for the proposed solution. It maintains two fundamental quantities:

- a combined state vector

$$\begin{aligned}\boldsymbol{\mu} &= \begin{pmatrix}\mathbf{x}^T & \mathbf{m}_1^T & \dots & \mathbf{m}_N^T\end{pmatrix}^T \\ &= \begin{pmatrix}x & y & \vartheta & m_{1,x} & m_{1,y} & \dots & m_{N,x} & m_{N,y}\end{pmatrix}^T\end{aligned} \tag{1}$$

- a covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{xx} & \mathbf{\Sigma}_{xm} \\ \mathbf{\Sigma}_{mx} & \mathbf{\Sigma}_{mm} \end{bmatrix}$$

The combined state vector consists of the estimated robot pose (x, y and ϑ) and the estimated coordinates of the landmarks ($m_{i,x}$ and $m_{i,y}$). The covariance matrix is divided into sub-matrices where:

- $\mathbf{\Sigma}_{xx}$ represents the covariance matrix of the robot pose, capturing the uncertainty in the estimated robot pose.
- $\mathbf{\Sigma}_{xm}$ represents the covariance matrix of the robot pose and the local map.
- $\mathbf{\Sigma}_{mm}$ represents the covariance matrix of the map, containing the uncertainty in the estimated coordinates of the landmarks.

At each point in time the EKF SLAM algorithm with known correspondence updates the aforementioned quantities using two main steps: prediction and correction. The prediction step utilizes the control vector and motion model to estimate the next state of the robot and predict the corresponding covariance matrix. It can be summarized as follows:

$$\begin{aligned} \bar{\boldsymbol{\mu}}_t &= F(\mathbf{u}_t, \boldsymbol{\mu}_{t-1}) \\ \bar{\mathbf{\Sigma}}_t &= \mathbf{G}_t \mathbf{\Sigma}_{t-1} \mathbf{G}_t^T + \mathbf{V}_t \mathbf{R}_t \mathbf{V}_t^T \end{aligned} \tag{2}$$

where:

- $\bar{\boldsymbol{\mu}}_t$ represents the predicted state.
- $F(\cdot)$ represents the motion model that estimates the next state based on the control input $\mathbf{u}_t$ and the previous state $\boldsymbol{\mu}_{t-1}$.
- $\bar{\mathbf{\Sigma}}_t$ represents the predicted covariance matrix.
- $\mathbf{R}_t$ represents the noise matrix associated with the control inputs.
- $\mathbf{G}_t$ and $\mathbf{V}_t$ are the Jacobian matrices of the motion model with respect to the robot pose ($\mathbf{x}$) and the control inputs ($\mathbf{u}$) respectively, i.e.

$$\mathbf{G}_t = \left. \frac{\partial F(\mathbf{u}, \boldsymbol{\mu})}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x}=\mathbf{x}_{t-1} \\ \mathbf{u}=\mathbf{u}_t}} \qquad \mathbf{V}_t = \left. \frac{\partial F(\mathbf{u}, \boldsymbol{\mu})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{x}=\mathbf{x}_{t-1} \\ \mathbf{u}=\mathbf{u}_t}}$$

The correction step incorporates the measurements obtained from a range sensor to refine the predicted robot pose. It can be summarized as follows:

$$\begin{aligned} \mathbf{S}_t &= \mathbf{H}_t \bar{\mathbf{\Sigma}}_t \mathbf{H}_t^T + \mathbf{Q}_t \\ \mathbf{K}_t &= \bar{\mathbf{\Sigma}}_t \, \mathbf{H}_t^T \, \mathbf{S}_t^{-1} \\ \boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t(\mathbf{z}_t - H(\bar{\boldsymbol{\mu}}_t)) \\ \mathbf{\Sigma}_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\mathbf{\Sigma}}_t \end{aligned} \tag{3}$$

where:

- $\mathbf{S}_t$ represents the innovation covariance.
- $\mathbf{K}_t$ represents the Kalman Gain, determining the weight of the measurement update.
- $\mathbf{z}_t$ represents the observed measurement.
- $H(\cdot)$ represents the observation model that yields the expected measurement based on the predicted state $\bar{\boldsymbol{\mu}}_t$.
- $\mathbf{H}_t$ represents the Jacobian matrix of the observation model with respect to the state vector $\boldsymbol{\mu}$, i.e.

$$\mathbf{H}_t = \left. \frac{\partial H(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \right|_{\boldsymbol{\mu}=\bar{\boldsymbol{\mu}}_t}$$

- $\mathbf{Q}_t$ represents the sensor noise matrix.
- $\mathbf{I}$ is the identity matrix.

By iteratively performing the prediction and correction steps, the distributed EKF SLAM system achieves collaborative mapping and localization among the robotic agents. The known correspondence of landmarks facilitates efficient data association and enhances the accuracy of individual robot pose estimates while reducing uncertainty. The system leverages communication capabilities for exchanging messages such as sensor measurements, robot poses, and map updates among the robotic agents.

B. *Information consensus filter*

The solution incorporates a consensus mechanism to merge the local maps built by individual robots. The consensus process involves exchanging information among robots to reach a consensus on the global map. This enables the robots to share their local map information and refine the global map collaboratively.

The consensus algorithm assumes that each landmark state is statistically independent, with uncertainty represented by a Gaussian distribution. By leveraging the concept of the Minimum Variance Unbiased Estimator (MVUE), the algorithm combines the information received from other robots with the current estimate of the receiving agent to obtain a consensus estimate of the landmark state.

To perform the consensus operation, the algorithm utilizes the inverse of the 2×2 covariance matrix representing the x and y coordinates of each landmark, resulting in the Fisher Information Matrix. Additionally, the information vector is obtained by multiplying the inverse covariance matrix with the estimated landmark state. This process can be expressed mathematically as follows:

$$\mathbf{Y}_i = \mathbf{\Sigma}_i^{-1} \qquad\qquad \mathbf{y}_i = \mathbf{Y}_i \, \mathbf{m}_i$$

Here, $\mathbf{Y}_i$ represents the information matrix, $\mathbf{\Sigma}_i$ is the covariance matrix (2×2) of the landmark coordinates, $\mathbf{m}_i$ is the state of the *i*-th landmark (estimated $\mathbf{m}_{i,x}$ and $\mathbf{m}_{i,y}$ coordinates), and $\mathbf{y}_i$ is the information vector.

Since the Fisher Information Matrix exhibits an additive property, separate information consensus operations can be performed for each observed landmark. For a specific landmark, the algorithm combines the information vectors received from other robots by averaging them:

$$\mathbf{y}_{MVUE} = \frac{1}{r} \sum_{j=1}^{r} \mathbf{y}_j \qquad Y_{MVUE} = \frac{1}{r} \sum_{j=1}^{r} \mathbf{Y}_j \tag{4}$$

In these equations, r represents the number of participating robots.

The consensus algorithm further refines the estimate of the global map by inverting the equation of the Fisher Information Matrix to obtain the estimated state:

$$\mathbf{\Sigma}_i = \mathbf{Y}_i^{-1} \qquad\qquad \mathbf{m}_i = \mathbf{Y}_i^{-1}\,\mathbf{y}_i$$

By applying this consensus mechanism, the robots can effectively merge the information received from other robots regarding the landmarks and obtain a refined estimate of the global map. This collaborative approach improves the accuracy and completeness of the map representation, leading to enhanced localization and navigation capabilities for the multi-robot system.

IV. Implementation Details

In this section, we provide detailed information about the implementation of the distributed EKF SLAM system, highlighting the utilization of the UTIAS Multi-Robot Cooperative Localization and Mapping (MRCLAM) [2] dataset for simulation and evaluation purposes. The solution was implemented using Matlab. The provided code encompasses functions responsible for executing the prediction and correction steps of the EKF SLAM algorithm, as well as the consensus mechanism for map merging.

A. MRCLAM Dataset

The MRCLAM dataset played a crucial role in providing a comprehensive representation of a multi-robot environment for the simulation and evaluation of the distributed EKF SLAM system. It consists of 9 individual datasets, each featuring 5 robots operating in a 2D indoor environment. The robots in the dataset are constructed using the iRobot Create platform, which employs a differential drive configuration with two wheels. Equipped with a monocular camera and a laptop computer running Player software, each robot captures rectified images at a resolution of 960×720 pixels.

In the MRCLAM dataset, the robots navigate the indoor space by moving towards randomly assigned waypoints. As they traverse the environment, the robots log odometry data, capturing their forward velocity (v) along the x-axis of their body frame and their angular velocity (ω) about the z-axis using the right-hand rule. These odometry measurements contribute to the prediction step of the EKF SLAM algorithm.

The monocular cameras mounted on the robots play a crucial role in sensing the environment and extracting valuable information. By processing the rectified images, the cameras detect unique barcodes attached to cylindrical landmarks. Each barcode encodes a unique identification number, enabling the robots to extract range and bearing measurements to the landmarks. This range and bearing information is crucial for mapping and localization within the SLAM system.

To provide groundtruth information, the MRCLAM dataset incorporates a 10-camera Vicon motion capture system. Operating at a frequency of 100Hz, the Vicon system provides accurate pose measurements (x, y, ϑ) for each robot and position measurements (x, y) for the landmarks. The Vicon system's reference frame serves as the inertial reference frame within the datasets, ensuring precise groundtruth information on the order of $1\text{x}10^{-3}$ meters.

By integrating the MRCLAM dataset into the system model, we were able to simulate realistic scenarios and evaluate the performance of the distributed EKF SLAM algorithm under varying conditions. The dataset's detailed information about the robots, sensors, odometry data, landmark measurements, and groundtruth poses and positions allowed for an accurate representation and analysis of the SLAM system.

Moving forward, we will discuss the technical aspects of the implementation.

B. EKF SLAM initialization

The combined state vector of the EKF SLAM algorithm are zero initialized. This means that each robot computes the local map and their own trajectory on a local coordinate frame. On the other hand the covariance matrix is initialized as follows:

$$\Sigma_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \infty & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \infty \end{bmatrix}$$

Since the floating point representation of infinity produces singularities in the correction step of the EKF SLAM algorithm, a natural implementation choice is to set the diagonal values to a large number. In this project a value of 10^{10} was used.

On top of this, the EKF SLAM algorithm implementation also uses a vector of boolean values that keeps track of the observed landmarks.

C. Motion Model

As mentioned in Section III, one of the key assumptions is a velocity based motion model. Such model represents a non-linear function that enables the robot to predict its next pose based on the current estimated pose and the control inputs. This mathematical expression of the motion of the robot corresponds to the function $F(\mathbf{u}_t, \boldsymbol{\mu}_{t-1})$ in equation (2).

One interesting property of the model is that it only affects the robot's motion and not the landmarks. In other terms, the prediction step only updates the first three components of the combined state vector in equation (1), leaving the landmarks' estimated position unmodified, i.e.

$$\bar{\mathbf{x}}_t = F(\mathbf{u}_t, \mathbf{x}_{t-1})$$

Let the control inputs $\mathbf{u}_t$ contain the translation (v_t) and angular (ω_t) velocity recorded at a time step t, and let Δt represent the sampling interval between two adjacent odometry measurements. The velocity based motion model can be formally represented as follows:

$$\bar{\mathbf{x}}_t = \mathbf{x}_{t-1} + \begin{bmatrix} -\frac{v_t}{\omega_t}\sin(\vartheta_{t-1}) + \frac{v_t}{\omega_t}\sin(\vartheta_{t-1} + \omega_t\Delta t) \\ \frac{v_t}{\omega_t}\cos(\vartheta_{t-1}) - \frac{v_t}{\omega_t}\cos(\vartheta_{t-1} + \omega_t\Delta t) \\ \omega_t\Delta t \end{bmatrix} \quad (5)$$

However, such mathematical representation presents a singularity when the angular velocity approaches zero. When a singularity is reached the following expression is employed:

$$\bar{\mathbf{x}}_t = \mathbf{x}_{t-1} + \begin{bmatrix} v_t \Delta t \cos(\vartheta_{t-1}) \\ v_t \Delta t \sin(\vartheta_{t-1}) \\ 0 \end{bmatrix}$$

Since the motion model only affects the robot's estimated pose, one can exploit the sparsity and symmetry of the covariance matrix to apply several optimizations and reduce computation time of the prediction step. In particular, the Jacobian of the motion model with respect to the state vector takes the following form:

$$\mathbf{G}_t = \begin{bmatrix} \mathbf{G}_t^x & 0 \\ 0 & \mathbf{I} \end{bmatrix}$$

where:

$$\mathbf{G}_t^x = \left.\frac{\partial F(\mathbf{u}, \mathbf{x})}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_{t-1} \\ \mathbf{u}=\mathbf{u}_t}}$$

Substituting such expression in equation (2) yields:

$$\begin{aligned} \bar{\mathbf{\Sigma}}_t &= \mathbf{G}_t \mathbf{\Sigma}_{t-1} \mathbf{G}_t^T + \mathbf{V}_t \mathbf{R}_t \mathbf{V}_t^T \\ &= \begin{bmatrix} \mathbf{G}_t^x & 0 \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{xx} & \mathbf{\Sigma}_{xm} \\ \mathbf{\Sigma}_{mx} & \mathbf{\Sigma}_{mm} \end{bmatrix} \begin{bmatrix} (\mathbf{G}_t^x)^T & 0 \\ 0 & \mathbf{I} \end{bmatrix} + \mathbf{V}_t \mathbf{R}_t \mathbf{V}_t^T \\ &= \begin{bmatrix} \mathbf{G}_t^x \mathbf{\Sigma}_{xx} (\mathbf{G}_t^x)^T & \mathbf{G}_t^x \mathbf{\Sigma}_{xm} \\ (\mathbf{G}_t^x \mathbf{\Sigma}_{xm})^T & \mathbf{\Sigma}_{mm} \end{bmatrix} + \mathbf{V}_t \mathbf{R}_t \mathbf{V}_t^T \end{aligned} \tag{6}$$

Equation (6) exploits the symmetry and sparsity of the matrix $\mathbf{G}_t$ to perform optimized computation of the covariance update. In fact, it is only necessary to compute two quantities in order to update the covariance matrix.

D. Observation Model

Another key assumption of the underlying EKF SLAM algorithm is a range and bearing measurement model. Such non-linear function corresponds to the term $H(\bar{\boldsymbol{\mu}}_t)$ in equation (3). The behaviour of the observation model changes whether or not the landmarks was already observed by the robot or not.

Let $\mathbf{z}_t^i = (\rho_t^i, \phi_t^i)$ be the range and bearing observation of the i-th landmark. If the landmark has not been observed before, the correction step initializes the landmark position using the current predicted state ($\bar{\mathbf{x}}_t$) and the relative measurement, i.e.

$$\begin{bmatrix} m_{i,x} \\ m_{i,y} \end{bmatrix} = \begin{bmatrix} \bar{x}_t \\ \bar{y}_t \end{bmatrix} + \rho_t^i \begin{bmatrix} \cos(\phi_t^i + \bar{\vartheta}_t) \\ \sin(\phi_t^i + \bar{\vartheta}_t) \end{bmatrix}$$

In case the i-th landmark has already been observed, the correction step proceeds to compute the expected observation.

$$\mathbf{z}_t^i = \begin{bmatrix} \sqrt{(m_{i,x} - \bar{x}_t)^2 + (m_{i,y} - \bar{y}_t)^2} \\ \text{atan2}((m_{i,y} - \bar{y}_t), (m_{i,x} - \bar{x}_t)) - \bar{\vartheta}_t \end{bmatrix}$$

E. Map Alignment

Before joining a map with the local maps obtained from neighbouring robots, a map alignment preprocessing step is necessary. This is because the maps constructed by each robot are represented in a different coordinate system. For this reason, whenever a new message is received by an agent, it first finds the common observed landmarks and computes the optimal transformation that best aligns the two sets of paired points. Once the optimal translation and rotation is obtained, the preprocessing step then transforms the received landmarks in the receiving agent coordinate system and rotates the received covariance matrices.

The optimal rotation and translation that aligns the two point clouds is obtained using Kabsch-Umeyama algorithm [4].
Let $\mathbf{A}$ and $\mathbf{B}$ be the local maps of the i-th and j-th robots respectively. To find the transformation that aligns the landmarks in $\mathbf{A}$ with the landmarks in $\mathbf{B}$ the preprocessing step follows these steps:

1) First, the algorithm computes the centroids of the two sets of paired points.

$$\mathbf{c}_A = \frac{1}{n}\sum_{k=1}^{n} \mathbf{a}_k \qquad \mathbf{c}_B = \frac{1}{n}\sum_{k=1}^{n} \mathbf{b}_k$$

2) Then, it subtracts such values from the corresponding landmark position. This way the two sets of points have their centroids in the origin of the same coordinate system.

$$\mathbf{A}' = \mathbf{A} - \mathbf{c}_A \qquad \mathbf{B}' = \mathbf{B} - \mathbf{c}_B$$

3) The algorithm computes the cross-covariance matrix $\mathbf{W}$ between the two paired point sets, i.e.

$$\mathbf{W} = (\mathbf{A}')^T \mathbf{B}'$$

4) Lastly, the Singular Value Decomposition (SVD) of the cross covariance matrix $\mathbf{W}$ is obtained.

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

Finally, the optimal translation $\mathbf{t}$ and rotation matrix $\mathbf{R}$ can be derived using the following equations:

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} 1 & 0 \\ 0 & sign(|\mathbf{W}|) \end{bmatrix} \mathbf{U}^T \qquad \mathbf{t} = \mathbf{c}_B - \mathbf{R}\,\mathbf{c}_A$$

F. Flow of execution

During a simulation step each robot performs the following actions:

1) Perform the prediction step of the EKF algorithm
2) If landmarks measurements are available, perform the correction step of the EKF algorithm.
3) If measurements of neighbouring robots are available, construct a message and send it to the observed agent.
4) If at least one message is received, refine the landmarks estimates using the consensus algorithm.

The message to be sent to the observed agent includes three main quantities: the estimated position of each landmark, the associated covariance matrix and the landmark state (observed or not observed).

For each message received, if any, the agent extract the content, determines the number of common landmarks and aligns the two maps. Please notice, that if the number of common landmarks is not greater than 1, the message is discarded and the content is not considered in the consensus algorithm. Otherwise, the received landmark positions $\mathbf{m}_i$ and covariance matrices $\mathbf{\Sigma}_{ii}$ are transformed in the coordinate system of the recipient ($\mathbf{m}_i^R$ and $\mathbf{\Sigma}_{ii}^R$), i.e.

$$\mathbf{m}_i^R = \mathbf{R}\mathbf{m}_i + \mathbf{t} \qquad \mathbf{\Sigma}_{ii}^R = \mathbf{R}\mathbf{\Sigma}_{ii}\mathbf{R}^T$$

Finally, the information consensus is performed using equation (4), where the factor r takes into consideration the number of robots that: participate in the consensus and have observed the i-th landmark. In other terms, the consensus only refines the landmarks in common between the sender and the recipient.

V. RESULTS

In this section, the proposed Distributed EKF SLAM system with known correspondence is compared to the standard EKF SLAM algorithm in order to underline the advantages and drawbacks of the distributed solution. In particular, the results focus on the quantitative analysis of each robots' estimated poses compared to the ground truth and their local map accuracy throughout the simulation interval.

A. Localization

The error in the pose of the robot, when compared to the ground truth, is obtained by computing the deformation energy needed to transform the ground truth pose ($\mathbf{T}_t^*$) with the estimated pose ($\mathbf{T}_t$) up to a specific simulation step t [3]. Such metric measures the error in successive relative poses between the ground truth ($\Delta\mathbf{T}_{i,j}^*$) and the estimated trajectory ($\Delta\mathbf{T}_{i,j}$) of the robot.

$$\mathbf{T}_t = \begin{bmatrix} \cos(\vartheta_t) & -\sin\vartheta_t & x_t \\ \sin\vartheta_t & \cos(\vartheta_t) & y_t \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Formally, given a tuple of three values $\mathbf{x_t} = (x_t, y_t, \vartheta_t)$ representing the position and orientation in the plane, the pose is converted to an homogeneous transformation using equation (7).

$$\Delta\mathbf{T}_{ij} = \mathbf{T}_j\,\mathbf{T}_i^{-1} \qquad \Delta\mathbf{T}_{ij}^* = \mathbf{T}_j^*\,(\mathbf{T}_i^*)^{-1} \tag{8}$$

The relative transformation between a pose $\mathbf{T}_i$ and a pose $\mathbf{T}_j$ is formally represented using the expression in equation (8). Such quantity represents the transformation to move the pose $\mathbf{T}_i$ (or $\mathbf{T}_i^*$) onto $\mathbf{T}_j$ (or $\mathbf{T}_j^*$).

$$\begin{aligned} \varepsilon = \frac{1}{N}\sum_{i,j=1}^{N} & trans(\Delta\mathbf{T}_{ij}\,(\Delta\mathbf{T}_{ij}^*)^{-1})^2 \\ & + rot(\Delta\mathbf{T}_{ij}\,(\Delta\mathbf{T}_{ij}^*)^{-1})^2 \end{aligned} \tag{9}$$

The metric used to evaluate the correctness of the pose estimate with respect to the ground truth is depicted in equation (9), where:

- N is the number of relative relations, i.e. relative transformation between successive poses.
- $trans(\cdot)$ represents the translation component of the obtained transformation.
- $rot(\cdot)$ represents the rotation component of the obtained transformation.

For clarity of the results, it is best to separate the translation and rotation component of the obtained transformation to track the Root Mean Squared Error (RMSE) both in the relative position and orientation of the robot, i.e.:

$$\begin{aligned} RMSE_{trans} &= \sqrt{\frac{1}{N}\sum_{i,j=1}^{N} trans(\Delta\mathbf{T}_{ij}\,(\Delta\mathbf{T}_{ij}^*)^{-1})^2} \\ RMSE_{rot} &= \sqrt{\frac{1}{N}\sum_{i,j=1}^{N} rot(\Delta\mathbf{T}_{ij}\,(\Delta\mathbf{T}_{ij}^*)^{-1})^2} \end{aligned} \tag{10}$$

TABLE I
TRANSLATION AND ROTATION RMSE OF DATASET 5

Robot #	EKF		D-EKF	
	$\sqrt{\varepsilon_{tran}}$ [m]	$\sqrt{\varepsilon_{rot}}$ [rad]	$\sqrt{\varepsilon_{tran}}$ [m]	$\sqrt{\varepsilon_{rot}}$ [rad]
1	0.0777	0.0334	0.0776	0.0335
2	0.0872	0.0377	0.0872	0.0377
3	0.0774	0.0332	0.0772	0.0332
4	0.1582	0.0601	0.1581	0.0601
5	0.1325	0.0484	0.1329	0.0484

Table (I) summarizes the RMSE results for both the translation and rotation components of the error metric. Both the standard EKF and the Distributed EKF SLAM algorithms performed exactly the same, with some negligible differences that are not statistically significant.

This is reasonable given that the consensus algorithm only estimates the position of the landmarks and not the pose of the robots. As a result, the Distributed EKF SLAM algorithm does not alter in any way the estimated trajectory of the robot.

B. Mapping

The MRCLAM datasets provides the ground truth position of the landmarks with respect to the world reference frame. However, the local map constructed by the robots are in the robot reference frame. As such in order to compare the accuracy on the estimated position of the landmarks derived by the EKF SLAM algorithm and the proposed distributed solution of the SLAM problem, the two datasets need to be represented in the same coordinate system.

To derive this rigid transformation, the project makes use of the Kabsch algorithm [4] to find the optimal translation and rotation that aligns the ground truth to the local map constructed by each robot. The resulting rotation matrix and translation vector are then applied to the ground truth coordinates of the landmarks.

Once the two datasets are aligned, it is possible to compute the RMSE of the position of the landmarks since there is a known correspondence between the two point clouds throughout the entire simulation.

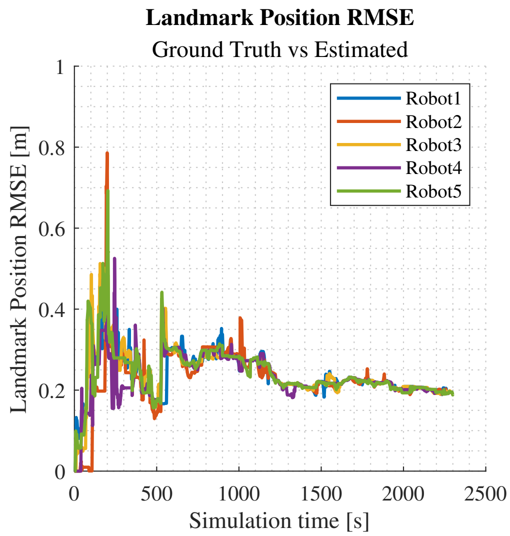

Fig. 1. Mapping accuracy of Dataset 5 throughout the simulation with the proposed distributed solution.

Figure 1 depicts the RMSE error of the local map when compared to the ground truth for each of the 5 robots. In this figure we can see the effect of the consensus algorithm in action: all of the three robots converge to a common value as suggested by the monotonically decreasing shape of the RMSE.

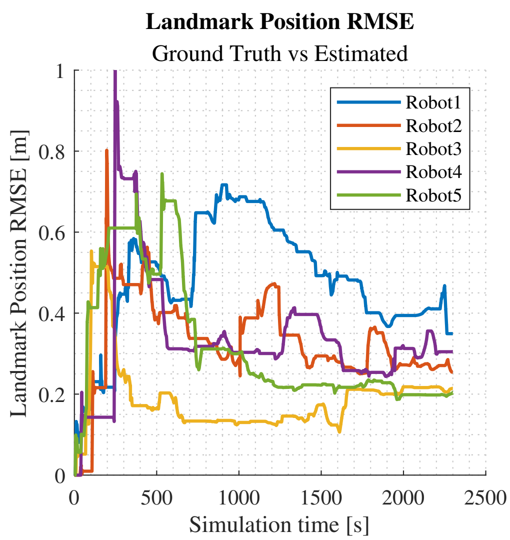

Fig. 2. Mapping accuracy of Dataset 5 throughout the simulation with the standard EKF SLAM algorithm.

Figure 2, on the contrary depicts the mapping accuracy of the standard EKF SLAM algorithm. In such situation no message is exchanged between robots and therefore no consensus is guaranteed to be reached.

An interesting detail that has emerged during the result gathering process is the effect of the measurement uncertainty on the position of the landmarks. In figure 1 and 2 the measurement noise matrix is:

$$\mathbf{Q} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix} = \begin{bmatrix} 11.8 & 0 \\ 0 & 0.18 \end{bmatrix}$$

This guess of the covariance matrix is over estimated as it assumes a standard deviation in the range measurement of $\sigma_r \approx 3.43$ meters and a standard deviation in the bearing measurement of $\sigma_b \approx 24$ degrees.

A more realistic measurement noise matrix was obtained from a naive analysis of the measurements of the dataset:

$$Q = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix} = \begin{bmatrix} 0.2025 & 0 \\ 0 & 0.011 \end{bmatrix} \tag{11}$$

Such quantity considers a range measurement uncertainty of $\sigma_r = 0.45$ meters and a bearing measurement uncertainty of $\sigma_b = 6$ degrees.

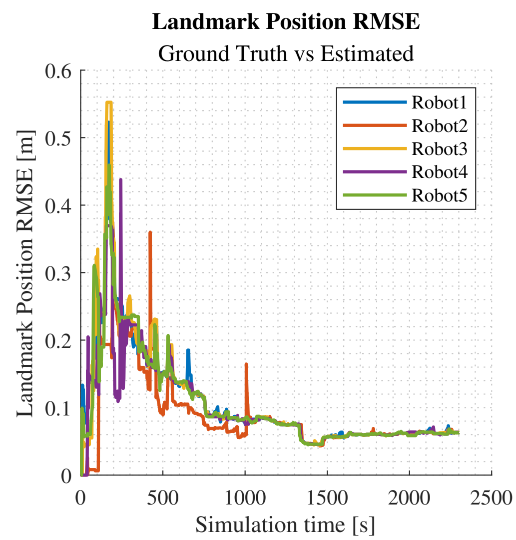

Fig. 3. Mapping accuracy of Dataset 5 throughout the simulation with the distributed EKF SLAM algorithm, with lower measurement uncertainty.

With a "small" measurement uncertainty in the range and bearing, the standard EKF SLAM algorithm for all 5 robots was able to match the map accuracy of the Distributed solution as depicted in figure 3 and 4.

Hence, the distributed solution is not only able to match the mapping and localization performance of the standard EKF SLAM algorithm, but also provide more accurate local maps combining the knowledge of neighbouring agents.

Moreover, the effect of the measurement noise matrix on the local map accuracy confirms the main advantage of the distributed approach. In fact, a more accurate map can be obtained by employing a team of robot equipped with less accurate sensors.

In the next section, the main conclusion drawn from the results discussed in this section are proposed, as well as the limitations of the algorithm and some pointers regarding the further opportunity and improvement that this method provides.

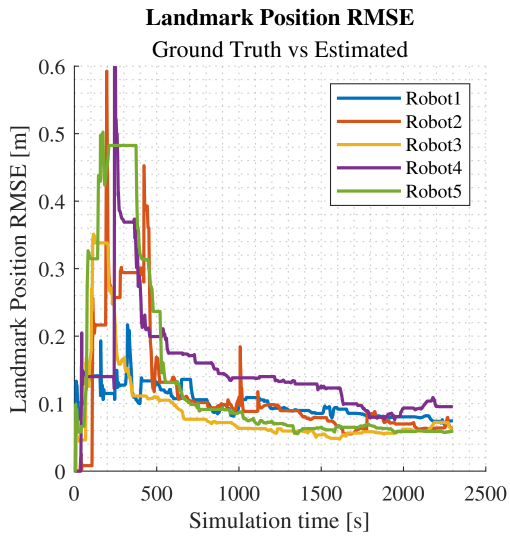

Fig. 4. Mapping accuracy of Dataset 5 throughout the simulation with the standard EKF SLAM algorithm, with lower measurement uncertainty.

VI. Conclusions

In this study, a Distributed EKF SLAM algorithm with known correspondence was proposed and compared to the standard EKF SLAM algorithm. The proposed algorithm aimed to improve the mapping and localization accuracy by enabling robots to exchange information and reach a consensus on the estimated map.

The results of the evaluation demonstrated that the Distributed EKF SLAM algorithm performed on par with the standard EKF SLAM algorithm in terms of localization accuracy. The pose estimates of the robots showed similar RMSEs for both translation and rotation components.

In terms of mapping accuracy, the Distributed EKF SLAM algorithm outperformed the standard EKF SLAM algorithm when considering larger measurement errors. The local maps constructed by the robots using the distributed solution showed lower RMSE compared to the standard algorithm, indicating improved accuracy in landmark positions.

However, it should be noted that the advantage of the distributed approach was more pronounced when the measurement uncertainties were larger. When smaller measurement uncertainties were considered, the standard EKF SLAM algorithm was able to achieve similar mapping accuracy. The results highlighted the potential of the distributed approach to leverage the collective knowledge of neighbouring robots and improve the accuracy of local maps. This could be particularly valuable in scenarios where robots with less accurate sensors can benefit from the information shared by more accurate robots.

A. Limitations and Future Work

While the proposed Distributed EKF SLAM algorithm with known correspondence has shown promising results, there are several limitations and areas for future improvement:

- **Unknown Correspondence**: The algorithm assumes known correspondence between features in the environment. However, in real-world scenarios, feature association can be challenging and may not be known beforehand. Future work should explore techniques to handle unknown correspondence, such as data association algorithms or feature matching techniques
- **Unknown Number of Landmarks**: The algorithm assumes that the number of landmarks is known a priori, which is often not the case in practice. In many scenarios, the number of landmarks is unbounded or dynamically changing. Future research should focus on developing algorithms that can handle an unknown number of landmarks, such as techniques based on sparsity or online feature selection.
- **Scaling with Landmark Count**: The EKF SLAM algorithm scales poorly with an increasing number of landmarks. As the number of landmarks grows, the computational complexity and memory requirements of the algorithm increase significantly. Future work should investigate alternative SLAM formulations, such as the Sparse Extended Information Filter (SEIF) [6] or other factor graph-based approaches, which can provide better scalability with a large number of landmarks.
- **Communication Constraints**: The proposed distributed algorithm assumes reliable communication and information exchange between robots. However, in real-world scenarios, communication may be subject to constraints, limitations, or failures. Future reseach should focus on developing robust distributed SLAM algorithms that can handle intermittent or unreliable communication, as well as study the impact of communication constraints on mapping accuracy.
- **Planar Environment**: The proposed algorithm focused on 2D SLAM in a planar environment. Extending the algorithm to handle 3D SLAM in complex 3D environments could be an interesting direction for future research.
- **Numerical Stability**: The EKF can suffer from numerical stability issues, especially when dealing with highly nonlinear systems or large uncertainties. The Extended Information Filter (EIF) generally provides a numerically more stable solution in most applications, with the additional advantage of having immediate access to the information matrix of the submap [5].

Addressing these limitations and exploring these research directions will contribute to the advancement of distributed SLAM algorithms, making them more robust, scalable, and applicable to real-world scenarios.

References

[1] Rajnikant Sharma, Clark N. Taylor, David W. Casbeer, "Distributed Cooperative SLAM using Information Consensus Filter," AIAA Guidance, Navigation, and Control Conference, 2010, DOI:10.2514/6.2010-8334

[2] Leung K. Y. K., Halpern Y., Barfoot T. D., Liu H. H. T. "The UTIAS Multi-Robot Cooperative Localization and Mapping Dataset." International Journal of Robotics Research, 30(8):969–974, July 2011

[3] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss and A. Kleiner, “On Measuring the Accuracy of SLAM Algorithms”, Autonomous Robots, 2009
[4] S. Umeyama, “Least-squares estimation of transformation parameters between two point patterns”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, DOI:10.1109/34.88573.
[5] S. Thrun, W. Burgard, D. Fox, “Probabilistic Robotics”, MIT Press, 2005
[6] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, A. Y. Ng, “Simultaneous Mapping and Localization With Sparse Extended Information Filters: Theory and Initial Results” In: Boissonnat, JD., Burdick, J., Goldberg, K., Hutchinson, S. (eds) Algorithmic Foundations of Robotics V. Springer Tracts in Advanced Robotics, vol 7. Springer, Berlin, Heidelberg., 2004