International Journal on Electrical Engineering and Informatics - Volume 9, Number 4, December 2017

Distributed Multi-Robot Rao-Blackwellized Particle Filtering Simultaneous Localization and Mapping with Consensus Calculation of Particle Weight and Posterior Parameters

Fadhil Hidayat, Bambang Riyanto Trilaksono, and Hilwadi Hindersyah

School of Electrical Engineering and Informatics
Bandung Institute of Technology
INDONESIA
fadhil@stei.itb.ac.id; briyanto@lskk.ee.itb.ac.id; hilwadi@stei.itb.ac.id

Abstract: Extensive research on single-robot simultaneous localization and mapping (SLAM) over the past decade has provided good estimation results in mapping small environment. This raises the idea for building a larger map by using some robots that are assembled within the team, also called multi-robot SLAM. There are two types of multi-robot SLAM, that is, centralized and distributed multi-robot SLAM. Centralized multi-robot SLAM method still has problems in case of system failure on some robots, especially the system failure on the robot central. Distributed multi-robot SLAM was developed to overcome the weaknesses in centralized multi-robot SLAM by: 1) building a map of larger environment by fusing local maps from a group of robots, 2) eliminating the dependence on central processing by using distributed computing, 3) overcoming the vulnerability in case of system failure in some robots, especially the system failure in the robot central, and 4) eliminating centralized communication, in which each robot requires only local communication with its neighboring robots. However, the distributed multi-robot SLAM method has not yielded good results on map estimates and localization estimates.

This paper proposes consensus-based distributed multi-robot SLAM method that can be applied to map common environments. We use FastSLAM algorithm as the main base in developing distributed multi-robot SLAM. We suggest using one of two selection of consensus parameters, that is, particle weight and posterior parameter. We assume that each robot has the same motion model and observation model; therefore, the system noise and observation noise is the same. The aim of this paper is to design consensus based distributed multi-robot SLAM method that provides better map estimation and localization results when applied to common environments. We concentrate to test the proposed methods on aspects: 1) root mean square error (RMSE) of map to see the map estimation performance, 2) RMSE of localization to see the robot's pose estimation performance, 3) mapping coverage to see how long it will take until all features are observed, and 4) processing time to see how long the computation process takes place per timestep.

Keywords: Simultaneous localization and mapping (SLAM), distributed multi-robot SLAM, consensus, particles weight, particles posterior.

1. Introduction

Simultaneous localization and mapping (SLAM) in robotics is a matter of building and updating map on an unknown environment while simultaneously defining and estimating robot's location within the map. SLAM is a very useful technique for robot to navigate on an unknown environment when there is no global positioning sensor, or it cannot connect to strong global positioning sensor signal while indoor, and there is no initial information about the environment. SLAM problem is a combination of two different issues including estimating robot's location (localization) and developing accurate map (mapping). Localization is a computation problem

Received: June 18th, 2017. Accepted: August 31th, 2017
DOI: 10.15676/ijeei.2017.9.4.1

of estimating robot's location on the environment. Meanwhile, mapping is a computation problem of integrating environmental information obtained from sensor installed on the robot into map representation. There are several map representations that can be built using SLAM; those are: (1) grid map, a map that represents the environment into occupied grid [1, 2, 3, 4, 5], (2) topological map, a map that represents the environment in a simplified graph, which only provides vital information such as robot's position and geometric structure of environment remains [1, 6, 7], (3) line map, a map that represents the environment using line and curve model [1, 8, 9, 10, 11], and (4) feature based map, a map that represents the environment into a set of features called landmark [1, 12, 13, 14].

SLAM development which use single robot generates three popular algorithms including Extended Kalman Filter (EKF) SLAM, Graph-Based SLAM, and Rao-Blackwellized Particle Filter (RBPF) SLAM [15]. All of the single-robot SLAM algorithms are statistical computing to find the most probable robot's pose and the most probable map parameter; the combination of these two probabilities is called joint probability distribution SLAM. The result of RBPF SLAM delivers better performance in terms of map convergence, computational complexity, and non-linearity caused by sample-space SLAM which grows with the increase of environmental observation results. Algorithms that implement RBPF SLAM are FastSLAM [12, 13]. This algorithm applies Particle Filter to track robot's location and applies Extended Kalman Filter to develop a map using measurement result obtained from sensor. FastSLAM algorithm represents localization and mapping in a set of weighted particles. The construction of a map is based on the information represented by the particle with the highest weight.

Researches on single-robot SLAM play an important role in the development of robotic mapping and exploration technology. SLAM technique is widely used in mapping planar and non-planar, outdoor and indoor area. SLAM also has applied to car like mobile robot for mapping indoor structure, flying robot for building aerial map, underwater robot for mapping seabed, and underground robot for mapping underground such as subways and tunnels. The performance of SLAM in single-robot mode has been tested in different types of environment and different types of robot; however, it still has limitation in mapping capabilities of only for small environment or small area. The question remains are whether the SLAM technique can be applied to multi-robot mode and whether the results will be approximately as good as the result of single-robot mode.

SLAM development in multi-robot area begins with the implementation of technique for coordinating a team of robots while they are exploring their environment. Team coordination technique on multi-robot SLAM aim to explore the environments efficiently from different unknown locations with the result that the overall exploration time can be minimized [16, 17, 18]. General algorithm of simultaneous localization and mapping for multi-robot using Particle Filtering method is also proposed. This method assumes that each robot in the team is active to broadcast both recent and history control data, and observation data to other teammates [3, 19]. Merging partial maps which have been built by each robot in the team can be done by applying rotation and translation method to find the similarity of overlapping regions against overall partial maps [20]. Sub-mapping method is introduced to reduce computational complexity of building and developing a global map. This method allows each robot to create their local sub-maps which represent the environment in the immediate vicinity of the robot then fuse and merge the overall sub-maps from all robots into the global map [21]. All these researches solve the problem of team coordination, completeness of information, and map merging method in centralized communication. However, distributed approaches are often necessary to address the problem of limited communication range and limited communication capabilities of each robot.

In the condition of initial position of each robot and prior information about the environment has been known, distributed approach to solve multi-robot SLAM problem can be done by disseminating robot's local maps to all robots. Graph optimizer method is used to reduce computational cost and communication load [22]. However, the distributed approach to solve multi-robot SLAM with unknown initial position and no prior information about the environment is still needed for several circumstances. This situation may occur due to a considerable difference in initial position of each robot. Multi-robot SLAM with unknown initial position by applying distributed Rao-Blackwellized Particle Filter SLAM is then proposed in [5]. This method requires a relative measurement position to compute global frame coordinate of global map [5, 23]. This method also introduces acceptance index to represent map similarity which is used to merge local maps into the global map. Nonetheless, this method assumes that the connection among robots are always connected.

A consensus algorithm proposed in [24] aim to find an agreement on map merging in limited communication based on local map information makes each robot own global map without central agent. This method consists of three stages: (1) agreeing with global reference frames of the environment, (2) agreeing with global data association, and (3) agreeing with global map [24, 25]. Nevertheless, this previous work only focuses on map merging scenario under limited communication. Each robot in a team shares their local map information with their local estimation on robot's position. An accurate robot's position is indispensable to the distributed multi-robot SLAM to perform team's coordination and robot's navigation.

Here, we are interested in performing distributed multi-robot SLAM under limited communication with accurate map merging and robot localization by modifying the transfer parameter among robots in a team. Based on the previous work on consensus-based distributed multi-agents target tracking using Particle Filtering method [26], we hypothesize that there are two different transfer parameters that represent other forms of local map which can be used as consensus parameter in multi-agent system. These parameters are particles weight and posterior parameter. In this paper, instead of using local maps information as transfer parameter, we propose distributed algorithm on multi-robot SLAM based on the consensus of Particle Filtering method using particles weight and using particles posterior parameters. Our experiment consists of six steps: (1) using single-robot SLAM based on Particle Filtering during no communication between the robot, (2) defining the type of communication that links several or all robots in the team, (3) building communication graph representation based on communication type, (4) determining agreement on relative position and frame reference among robots, (5) determining agreement on data association, and (6) determining agreement on global map. We found that there are three different types of communication scheme that links several or all robots in the team: (1) each connected robot is currently on the communication range and on the measurement range of neighbors or vice versa; (2) only several connected robots are currently on the communication range and on the measurement range of neighbors, others are currently on the communication range but are not on the measurement range of neighbors; and (3) all connected robots are currently on the communication range and are not on the measurement range of neighbors. Our analysis already considers these three different communications.

This paper is a continuation of our previous paper [27]. The aim of this paper is to proposes consensus-based distributed multi-robot SLAM method that can be applied to map common environments. The proposed method seeks to provide better mapping and localization result on distributed multi-robot SLAM when applied to common environments. Analysis on performance of proposed method focus on four aspects: 1) root mean square error (RMSE) of map to see the map estimation performance, 2) RMSE of localization to see the robot's pose estimation performance, 3) mapping coverage to see how long it will take until all features are observed, and 4) processing time to see how long the computation process takes place per timestep.

In general, every robot in the team starts exploring an unknown environment with unknown initial positions and unknown data association. In this experiment, we assume that the characteristics of all robots are identical. Each robot is equipped with the same range-bearing measurement sensor and the same communication device to build a feature-based map. At the beginning of the exploration, each robot performs single-robot RBPF SLAM and possesses a particle representation of a posterior. To achieve the goal of multi-robot SLAM, all robots are active to communicate one another during the exploration. In our case, the robots require only local communication with neighboring robot and do not require global knowledge about the network of the whole robots. While a robot is connected to one or more neighboring robots, every connected robot tries to measure the relative position of their neighboring robots and broadcast the data to the others. The robot remains performing single-robot SLAM when there is no connection with other robots or when relative position data is unavailable within the network. After receiving relative position data, the robot merges the information of data association and starts to perform multi-robot SLAM based on consensus Particle Filtering.

At the rest of this paper, we describe our research methodology in section 2. The experimental result is given in section 3. Finally, discussion and conclusions are summarized in section 4 and section 5 respectively.

2. Methodology

A. Problem Definition

Considering K as number of robots which explore an unknown environment, each robot k is active in doing communication to their neighbor robot k'. Communications between robots are represented by directed graph $G = (V, A)$, vertices $V = \{1,2,\ldots,K\}$ are the robot, and directed edges or arcs $(k, k') \in A$ are communication across the robots. The direction of the edges depends on the types of communication among robots. The explanations about communication types are described on section 2.4. A set of all robots that are connected to robot k is called neighboring robot N_k, where $N_k = \{k' \in V: (k, k') \in A; k \neq k'\}$. If there is no communication between robot k and their neighboring robots, then, the arcs of node V are (0,0) and the graph representation becomes $G = \{V, (0,0)\}$.

At the beginning of the process or while there is no communication among robots, each robot performs single-robot SLAM algorithm to calculate joint probability of SLAM. The join probability of SLAM is written as equation (1).

$$p(x_{0:t}, m|z_{0:t}, u_{0:t}) = p(m|x_{0:t}, z_{0:t})p(x_{0:t}|z_{0:t}, u_{0:t}) \tag{1}$$

Referring to the joint probability of SLAM on equation (1), x_t is vector of robot's position at time t; m is map made up from location of landmarks; $z_{0:t}$ is sensor observation; and $u_{0:t}$ is vector of input control given to the robot. Solution for singe-robot SLAM using Rao-Blackwellized Particle Filtering method consists of two phases: prediction and update. Prediction phase tries to find out estimated robot's current position based on its previous position and input control given to the robot. This phase requires robot's motion model $p(x_t|x_{t-1}, u_t)$ to calculate the estimation, so the formulation of prediction phase can be written as equation (2).

$$p(x_t, m|z_{0:t}, u_{0:t}, x_0) = \int p(x_t|x_{t-1}, u_t).p(x_{t-1}, m|z_{0:t-1}, u_{0:t-1}, x_0)dx_{t-1} \tag{2}$$

The update phase of single-robot Rao-Blackwellized Particle Filtering is a computation process which is aimed to correct the prediction results based on sensor measurement result and robot's true position. Since the true position of the robot is never known, this phase believes that all prediction results are the true robot's position. This phase requires sensor model $p(z_t|x_t, m)$ to calculate the degree of confidence of the prediction results and to build the map based on prediction results with the highest degree of confidence. The formulation of update phase can be written as equation (3).

$$\begin{aligned} p(x_t, m|z_{0:t}, u_{0:t}, x_0) &= \frac{p(z_t|x_t, m).p(x_t, m|z_{0:t-1}, u_{0:t}, x_0)}{p(z_t|z_{0:t-1}, u_{0:t})} \\ &\overset{Bayes}{=} \eta p(z_t|x_t, m).p(x_t, m|z_{0:t-1}, u_{0:t}, x_0) \end{aligned} \tag{3}$$

Using FastSLAM algorithm [12], robot's position is represented by weighted samples called particles. Current particles representation S_t on FastSLAM algorithm is notated as equation (4).

$$\begin{aligned} S_t &= \left[w_t^i, x_{0:t}^i, p\left(m|x_{0:t}^i, z_{0:t}\right)\right]_i^N \\ &= \left[w_t^i, x_{0:t}^i, \left\{m_{f,t}^i, \Sigma_{f,t}^i\right\}_{f=1}^F\right]_i^N \end{aligned} \tag{4}$$

Referring to equation (5), N is the number of particles; F is the number of observed landmarks from start $t = 0$ to time t; f is index of landmark or feature observed; i is index of the particles where $i = \{1,2,3,\ldots,N\}$; w_t^i is the weight of particle i; $x_{0:t}^i$ is information of robot's current position held by particle i; and $p\left(m|x_{0:t}^i, z_{0:t}\right)$ is the map representation which consists of landmarks position m and its uncertainty Σ.

During the exploration, the robots can detect the appearances, identify, and measure the relative position of neighboring robots k', and to exchange information within the network. Let d_t^k is the information from robot k that is transferred to all neighboring robots N_k, $d_t^{k'}$ is information that is received from the neighboring robot k', and let $D_t^k = \{d_t^k, \{d_t^{k'}\}\}$ is set of all information owned by robot k at time t. While the robot is connected to their neighboring robots, each robot performs multi-robot SLAM based on the consensus of algorithm to calculate joint probability of SLAM by involving the information that is received from the neighboring robots. Joint probability of multi-robot SLAM based on the consensus of algorithm can be written as equation (5).

$$p\left(x_{0:t}^{k,G}, m^{k,G} | z_{0:t}^{k,T}, u_{0:t}^{k,T}, \{d_t^{k'}\}\right) = p\left(m^{k,G} | x_{0:t}^{k,T}, z_{0:t}^{k,T}, \{d_t^{k'}\}\right) p\left(x_{0:t}^{k,G} | z_{0:t}^{k,T}, u_{0:t}^{k,T}, \{d_t^{k'}\}\right) \quad (5)$$

Referring to equation (5), the superscript k is index of the robot; the superscript k' is index of neighboring robot that are connected to robot k; the superscript T means that the data is in the format of local frame reference; and the superscript G means that the data is in the format of global frame reference. Considering S_k is the particles representation of robot k in multi-robot SLAM, the vertices on the graph representation of multi-robot SLAM holds the particles representation so it can be written as equation (6).

$$V = \{S_1, S_2, \dots, S_k\} \quad (6)$$

Let $\rho_k^{k'}$ is the relative position of neighboring robots k' measured by robot k and let $\mathrm{P}_k^{k'}$ is the uncertaity of the relative position. By including relative position and its uncertainty, particles representation of multi-robot SLAM can be written as equation (7).

$$S_{k,t}^G = [w_{k,t}^{G,i}, x_{k,t}^{G,i}, \{m_{k,f,t}^{G,i}, \Sigma_{k,f,t}^{G,i}\}_{f=1}^{F}, \{\rho_{k,t}^{G,i,k'}, \mathrm{P}_{k,t}^{G,i,k'}\}_{k'=1}^{K}]_{i=1}^{N} \quad (7)$$

In this paper, we use two different consensus parameters to perform multi-robot RBPF SLAM based on consensus particle filtering. They are particles weight and posterior parameters. First, particles weight reflects confidence degree of the data hold by the particle. Second, posterior parameter represents probability posterior of joint probability SLAM including probability of map and probability of robot's position.

The general consensus algorithm is shown in algorithm 1. There are a number of K robots in the agents' network and each member K has certain scalar values ξ_k that can be broadcasted to their neighboring agents.

Algorithm 1. General consensus algorithm
1. $\boldsymbol{Consensus\ Algorithm}(s_k, s_{k'})$:
2. $\quad \xi_k^{(0)} = s_k$
3. $\quad for\ \lambda = 1,2, \dots, \infty\ then$
4. $\quad\quad \xi_k^{(\lambda)} = u\left(\xi_k^{(\lambda-1)}, \left\{\xi_{k'}^{(\lambda-1)}\right\}_{k' \epsilon Nk}\right)$
$\quad\quad$ /* $\left\{\xi_{k'}^{(\lambda-1)}\right\}_{k' \epsilon Nk}$: s_k $neighbooring\ agents$*/
5. $\quad\quad \boldsymbol{broadcast}(\xi_k^{(\lambda)})$
6. $\quad\quad if(\boldsymbol{consensus})\ then\ \boldsymbol{end\ for}.$

Referring to general consensus algorithm in algorithm 1, λ is the number of consensus iterations and $u(.)$ is update function. There are three types of update function $u(.)$ which can be used on consensus algorithms, namely, averaging consensus, maximum consensus, and gossip consensus, written as equation (8), equation (9), and equation (10) respectively.

$$u\left(\xi_k^{(\lambda)}, \left\{\xi_{k'}^{(\lambda)}\right\}_{k' \epsilon Nk}\right) = \omega_{k,k}^{(\lambda)} \xi_k^{(\lambda)} + \sum_{k' \epsilon Nk} \omega_{k,k'}^{(\lambda)} \xi_{k'}^{(\lambda)} \quad (8)$$

$$u\left(\xi_k^{(\lambda)}, \left\{\xi_{k'}^{(\lambda)}\right\}_{k' \epsilon Nk}\right) = max(\xi_k^{(\lambda)}, \xi_{k'}^{(\lambda)}) \quad (9)$$

$$u\left(\xi_k^{(\lambda)}, \left\{\xi_{k'}^{(\lambda)}\right\}_{k' \epsilon Nk}\right) = \frac{1}{K} \xi_k^{(\lambda)} + \sum_{k' \epsilon Nk} \frac{1}{K} \xi_{k'}^{(\lambda)} \quad (10)$$

In the next section, we describe the core concept of our approach on distributed multi-robot SLAM solution based on consensus particle filtering algorithm.

B. Core Concept

In general, our approach to distributed multi-robot SLAM solution based on consensus particle filtering algorithm is shown in figure 2.1.

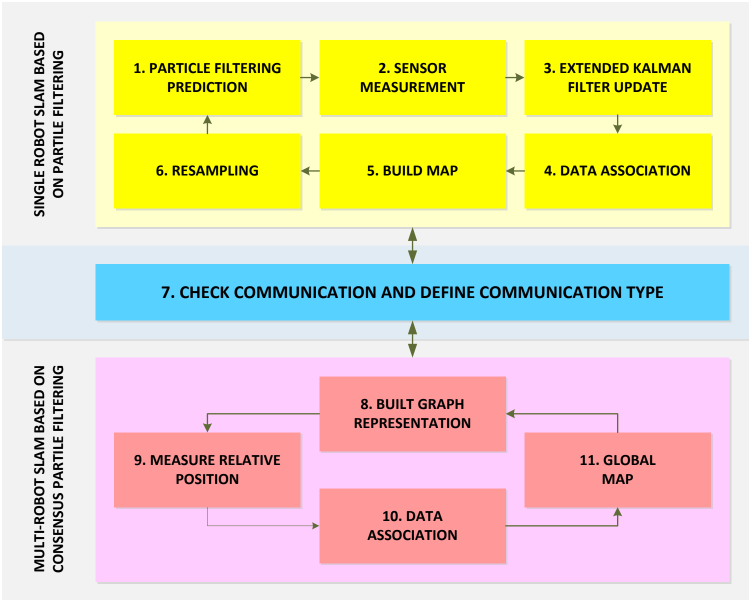


Figure 1. General concept of consensus-based distributed RB PF SLAM.

Algorithm 2. Switching mechanism

$\boldsymbol{Distributed\ SLAM\ Algorithm}(s_{k,t-1})$:
    $\boldsymbol{if}(s_{k,t-1} \leftarrow \emptyset)\ \boldsymbol{then}$
        $s_{k,t} = \boldsymbol{Single\_Robot\_SLAM}(s_{k,t-1})$
    $\boldsymbol{else}$
        $s_{k',\mathrm{t}} = \boldsymbol{Check\_Communication}(k')$
        $\boldsymbol{if}(s_{k',\mathrm{t}}\ \boldsymbol{is\ exist})\ \boldsymbol{then}$
            $s_k = \boldsymbol{consensus\_RBPF\_SLAM}(s_k, s_{k\prime\prime})$
        $\boldsymbol{else}$
            $s_{k,t} = \boldsymbol{single\_robot\_SLAM}(s_{k,t-1})$
        $\boldsymbol{end\ if}$
    $\boldsymbol{end\ if}$

Our approach to distributed multi-robot SLAM based on consensus particle filtering algorithm requires 11 steps which are composed of six steps of single-robot SLAM processing, four steps of multi-robot SLAM processing, and a step of communication among robots. If a robot in a team does not connect to other robots, then, the robot only performs single-robot SLAM processes. Otherwise, if a robot in a team connects to other robots, then, the robot performs all steps. Algorithm 2 shows the switching mechanism from single-robot SLAM to multi-robot SLAM.

C. Single Robot SLAM

The formulation of single robot SLAM in equation is factored into mapping problem $p(m|x_{0:t}, z_{0:t})$ and localization problem $p(x_{0:t}|z_{0:t}, u_{0:t})$, both are processed simultaneously. The single-robot SLAM formulation is resolved in two stages which are prediction step and update step. The prediction step estimates the robot's positions given the input control and robot's previous position. The prediction step is modeled as equation (11).

$$p(x_t, m|z_{0:t}, u_{0:t}, x_0) = \int p(x_t|x_{t-1}, u_t).\, p(x_{t-1}, m|z_{0:t-1}, u_{0:t-1}, x_0)dx_{t-1} \quad (11)$$

The update step is corrections to predicted results using observation results. The update step is modeled as equation (12).

$$p(x_t, m|z_{0:t}, u_{0:t}, x_0) = \frac{p(z_t|x_t, m).\, p(x_t, m|z_{0:t-1}, u_{0:t}, x_0)}{p(z_t|z_{0:t-1}, u_{0:t})} \quad (12)$$

$$\overset{Bayes}{=} \eta p(z_t|x_t, m).\, p(x_t, m|z_{0:t-1}, u_{0:t}, x_0)$$

To solve the single-robot SLAM problem using particle filtering algorithm, we use FastSLAM algorithm at which its details can be found [12]. Prediction step is implemented by spreading N number particles based on proposal distribution shown in equation (13).

$$x_t^i \sim \pi\left(x_t|x_{0:t-1}^i, u_t\right) \quad (13)$$

Importance weight is computed using equation (14).

$$w_t^i = w_{t-1}^i p\left(z_t|x_{0:t}^i, z_{0:t-1}\right)\left(\frac{p\left(x_t^i|x_{t-1}^i, u_t\right)}{\pi\left(x_t^i|x_{0:t-1}^i, u_t\right)}\right) \quad (14)$$

Data association is defined using equation (15).

$$\hat{f}_t^i = \underset{f_t}{\operatorname{argmax}}\, w_t^i(f_t) \quad (15)$$

Map is constructed based on particles with the most significant weight. Map construction can be written as equation (16).

$$p\left(m|x_{0:t}^i, z_{0:t}\right) = \prod_f^D p\left(m^f\middle|x_{0:t}^i, \mathrm{z}_{0:t}^f\right) \quad (16)$$

To reach the goal of multi-robot SLAM, all robots are active to communicate one another during the exploration. Next section shows the mechanism of communication across the robots.

D. Communication across the Robot

In order to achieve the goal of multi-robot SLAM that shown in equation (5), all robots are active to communicate one another during the exploration. Communications among robots are represented by directed graph $G = (V, A)$ in which vertices $V = \{1,2,\ldots,K\}$ are the robot and directed edges or arcs $(k, k') \in A$ are communication across the robots. Set of neighboring robots are $N_k = \{k' \in V: (k, k') \in A; k \neq k'\}$. While a robot is connected to one or more neighboring robots, every connected robot tries to measure the relative position of their neighboring robots $\rho_k^{k'}$ including its uncertainty $\mathrm{P}_k^{k'}$ and to broadcast the data to the others $(\rho, P \in D)$. There are three types of communication that determine the direction of the graph, that are:

1) Type 1, each of the connected robots is currently on the communication range and on the measurement range of its neighbors or vice versa. In this type, all connected robots share the current relative positions of other robots $\{\rho_{k,t}^{k'}, P_{k,t}^{k'}\}$ and receive their own relative positions measured by other robots $\{\rho_{k',t}^{k}, P_{k',t}^{k}\}$.
2) Type 2, only several connected robots are currently on the communication range and on the measurement range of their neighbors; others are currently on the communication range but are not on the measurement range of their neighbors. In this case, only robot k shares the current relative positions of other robots $\{\rho_{k,t}^{k'}, P_{k,t}^{k'}\}$.
3) Type 3, all connected robots are currently on the communication range and are not on the measurement range of their neighbors. Communication type 3 can only occur when the robots

have ever communicated with type 1 and 2 in the previous communication. In this case, no robot shares the current relative positions of other robots.
These communications type are illustrated as figure 2.2.

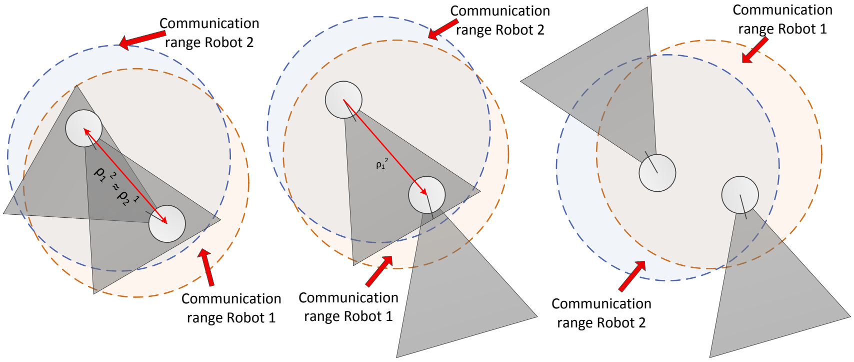


Figure 2. Communication type 1 (left), communication type 2 (middle), and communication type 3 (right)

Adjacency matrix or transition matrix of the communication graph A is built based on signal strength among communicated robots. To simplify computation and without loss of generality, we define the signal strength is at a value between 0 and 1. The value of signal strength represents the priority of process. The distributed multi-robot SLAM processing procedure starts from involving the data from the neighboring robots that are connected with the highest signal strength value. This processing procedure is developed based on type of communication between robots, shown in algorithm 3.

Algorithm 3. Processing procedure

$A = (\boldsymbol{Check\_Communication}(k'))$
$\boldsymbol{Processing\ Procedure}\ (S_k, k, A, D_{k'}^{k})$
  $\boldsymbol{for\ each}\ (k)\ \boldsymbol{do}$
    $B = sort\_descending\left(c_{k'}^{k} \in A\right)$
    $\boldsymbol{for\ each}\ (B)\ \boldsymbol{do}$
      $\boldsymbol{if}\ (B(\Upsilon)\ \boldsymbol{is}\ 3\ \boldsymbol{or}\ 2)\ \boldsymbol{then}$ /*Communication type 1 or 2
        $\boldsymbol{Calculate\_Global\_Frame}(S_k, D_{k'}^{k})$
      $\boldsymbol{end\ if}$
      $\boldsymbol{Global\_Data\_Association}(S_k, D_{k'}^{k})$
      $process\ one\ iteration\ of\ consensus\ algorithm\ using\ D_{k'}^{k}$
    $\boldsymbol{end\ for}$
  $\boldsymbol{end\ for}$

Next section shows how to define global frame reference and global data association.

E. Global Frame and Global Data Association

Since robot k is connected to the neighboring robots k' with connection type 1 or type 2, both robot k and robot k' perform calculation on global frame. In the communication type 1, robot k sends the relative position information $\left\{\rho_{k,t}^{k'}, P_{k,t}^{k'}\right\} \in D_t^k$ of the neighboring robot k' and receives its own relative position information $\left\{\rho_{k',t}^{k}, P_{k',t}^{k}\right\} \in D_t^k$ from the neighboring robot k'. In the communication type 2, robot k sends the relative position $\left\{\rho_{k,t}^{k'}, P_{k,t}^{k'}\right\} \in D_t^k$ of robot k' and does not receive its own relative position information from the neighboring robot k' or vice versa. Based on the assumption that all robots are equipped with same characteristics of range-bearing sensor, in the communication type 1, both robot k and robot k' will send and receive identical relative position $\rho_k^{k'} \approx \rho_{k'}^{k}$. In the communication type 2, robot k does not receive $\rho_{k'}^{k}$ from robot k'. Therefore, $\rho_{k'}^{k}$ is generated by transforming $\rho_k^{k'}$ geometrically $\rho_{k'}^{k} \doteqdot \rho_k^{k'}$ based on the position of robot k'.

Considering robot k starts single-robot SLAM in its own local frame referenceW_k^T, robot k' starts single-robot SLAM in its own local frame reference $W_{k'}^T$, using $\{\rho_{k,t}^{k'}, P_{k,t}^{k'}\} \in D_t^k$ and$\{\rho_{k',t}^{k}, P_{k',t}^{k}\} \in D_t^k$, and using J as the number of neigbor robots, the (0,0) global frame W_k^G coordinates of robot k are determined by calculating the center of mass method shown as equation (17).

$$W_k^G = \frac{1}{J} W_k^T \sum_{k'=1}^{J} W_{k'}^T \tag{17}$$

We refer to [5] in order to calculate global frame W_k^G. Based on global frame calculation in [5], robot k calculates local frame reference $W_{k'}^T$ of robot k' and applies roto-translation using formulation that is written as equation (18).

$$W_k^G = W_k^T \oplus W_k^{k'} \ominus W_{k'}^T \tag{18}$$

Each robot processes the global frame locally; thus, in every consensus iteration λ the robots try to find the convergence of the global frame $W_k^G(t)$ which can be expressed as equation (19).

$$\lim_{\lambda\to\infty} W_k^G(\lambda) = W_k^G = \frac{1}{J} W_k^T \sum_{k'=1}^{J} W_{k'}^w \tag{19}$$

Due to the change from local frame reference to global frame reference, then, any information of every connected robot must be transformed into global frame including landmarks information. The robot k position and relative position can be transformed directly using roto-translation. However, on the landmarks transformation, we will find some landmarks that are in close position, but they have different labels; considering $F_{k,t}^T \in D_t^k$ is data association table on robot k, and $F_{k',t}^T \in D_t^{k'}$ is data association table on robot k' connected to robot k. There are two steps of procedure for determining global data association: 1) detection of overlapping features, and 2) global labeling on all features including the overlapping features. In the detection step, the data association tables from both robot k and k' are combined, so the data association table of robot k can be expressed as equation (20).

$$F_{k,t}^G = F_{k,t}^w \cup F_{k',t}^w \tag{20}$$

Furthermore, each element is sorted in $F_{k,t}^G$ by its position in global coordinate frame using merge sort method. The asymptotic time complexity of merge sort is $O(n\ log\ n)$. The nearest neighbor method is used to detect the overlapping feature. Considering $f \in F$, th_b and th_c are the threshold range of overlapping features, the formulation of overlapping features detection is written as equation (21).

$$F_{k,t}^G = \begin{cases} f_{k,t}^G, \ if \ \dfrac{f_{k,t}^T(y) - f_{k',t}^T(y)}{f_{k',t}^T(x) - f_{k',t}^T(x)} \le th_b \\ \qquad and \ \dfrac{f_{k,t}^T(x,y,\theta) - x_{k,t}^G(x,y,\theta)}{f_{k',t}^T(x,y,\theta) - x_{k,t}^G(x,y,\theta)} \le th_c \\ f_{k,t}^w, \ otherwise \end{cases} \tag{21}$$

Each robot processes the global association data locally; thus, in every consensus iteration λ the robots try to find the convergence of the global data association table data $F_{k,t}^G(t)$ which can be expressed as equation (22).

$$\lim_{\lambda\to\infty} \{f_{k,t}^G \in D_t^{k,G}(\lambda)\} - \{f_{k',t}^G \in D_t^{k',G}(\lambda)\} = 0 \tag{22}$$

Next, the consensus calculation on particle weights and particle posterior parameters are shown in section 2.6 and section 2.7 respectively.

F. Consensus-Based Calculation of Particles Weight in Distributed Multi-Robot SLAM

Particle weight or importance weight represents the significance index of a particle; the greater the particle weight, the greater the probability that the particle is reused at the later stage. Based on the FastSLAM algorithm [12], the particle weight is calculated based on the equation (23).

$$w^i = \frac{target(x^i)}{proposal(x^i)} = \eta \int_{f_j}^{M} p\left(z_t \middle| x_{1:t}^i, f_j\right). p(f_j | x_{1:t}^i, z_{1:t-1}) d_{f_j} \quad (23)$$

Assuming that the observed noise distribution is zero-mean Gaussian, then, the equation (24) can be computed in Gaussian representation shown on equation (24).

$$w^i \simeq |2\pi Q|^{-\frac{1}{2}}. \exp\{-\frac{1}{2}(z_t - z^i)^T. Q^{-1}. (z_t - z^i)\} \quad (24)$$

If each interconnected robot performs calculations on the particle weight independently, the consensus algorithm aims to find the convergence on the average particle weight of all robots. Assuming robot k and k' are in a close position when communicating, both robots are observing identical environments. It is also assumed that robot k and robot k' already have the same frame reference and the same data association. The implementation steps of consensus-based calculation of particles weight in distributed multi-robot SLAM are as follows:

1) At the beginning of time t, each robot k and k' predicts robot pose x_t by spreading a number of N sample based on the proposal distribution. First, robot k samples a number of N samples that represent $x_{k,t}^i \sim p(x_{k,t} | x_{k,t-1}^i, u_{k,t})$ and robot k' samples a number of N sample that represent $x_{k\prime,t}^i \sim p(x_{k\prime,t} | x_{k\prime,t-1}^i, u_{k\prime,t})$ independently. It is assumed that the number of particle on robot k and k' is the same. Every sample particle consists of three data, those are, particle weight$w_{k,t}^i$, robot pose $x_{k,t}^i = \left[x_{k,t}^i \quad y_{k,t}^i \quad \theta_{k,t}^i\right]^T$, and map $m_{k,t}^i = \left\{\mu_{k,f}^i, \Sigma_{k,f}^i\right\}_{f=1}^{M}$. At this stage, the value of particle weight is $1/_N$, and map representation on time t is using map representation of previous step. Hence, a set of particle owned by robot k is $S_{k,t} = \left\{w_{k,t}^i,\ x_{k,t}^i, m_{k,t-1}^i\right\}_{i=1}^{N}$ and a set of particle owned by robot k' is $S_{k\prime,t} = \left\{w_{k\prime,t}^i,\ x_{k\prime,t}^i, m_{k\prime,t-1}^i\right\}_{i=1}^{N}$.
2) The robots move based on the input control, and then they simultaneously observe the environment using range bearing sensor that is attached to them. Robot k gets the measurement results $z_{k,t} = \left\{z_{k,t}^1, z_{k,t}^2, \dots, z_{k,t}^{f_k}\right\}$ and robot k' gets the measurement results$z_{k\prime,t} = \left\{z_{k\prime,t}^1, z_{k\prime,t}^2, \dots, z_{k\prime,t}^{l_{k\prime}}\right\}$. the number of landmarks and index landmarks observed by robots k and k' may be different.
3) Each robot k and k' performs calculation on likelihood distribution $p(z_t | x_t^i)$. In Gaussian representation, it is shown in equation (25).

$$p\left(z_t \middle| x_t^i\right) \sim w_{k,f,t}^i \quad (25)$$

$$w_{k,f,t}^i \simeq \left|2\pi Q_{k,f}^i\right|^{-\frac{1}{2}}. \exp\{-\frac{1}{2}\left(z_{k,f,t} - z_{k,f,t}^i\right)^T. {Q_{k,f}^i}^{-1}. (z_{k,f,t} - z_{k,f,t}^i)\}$$

Here, $z_{k,l,t} - z_{k,l,t}^i$ is computed based on equation (26).

$$z_{k,l,t} - z_{k,l,t}^i = \begin{bmatrix} m_{k,f,x} - x_{k,t,x}^i \\ m_{k,f,y} - x_{k,t,y}^i \\ atan2\left(m_{k,f,x} - x_{k,t,x}^i, m_{k,f,y} - x_{k,t,y}^i\right) - x_{k,t,\theta}^i \end{bmatrix} \quad (26)$$

Computation of $Q_{k,f}^i$ is shown on equation (27).

$$Q_{k,f}^i = \nabla H_{k,f}^i. \Sigma_{k,f}^i. \left(\nabla H_{k,f}^i\right)^T + Q \text{ where}$$
$$\nabla H_{k,f}^i = \begin{bmatrix} \frac{dx}{d} & \frac{dy}{d} \\ \frac{dy}{d2} & \frac{dx}{d2} \end{bmatrix}; \quad (27)$$
$$dx = m_{k,f,x}^i - x_{k,t,x}^i \text{ and } dy = m_{k,f,y}^i - x_{k,t,y}^i \ ;$$

$d2 = dx^2 + dy^2$ and $d = \sqrt{d2}$

From equation (27), $\nabla H^i_{k,f}$ is Jacobian of the feature set, $\Sigma^i_{k,f}$ is covarian matrix of the feature, and Q is the observed noise. This stage updates the set of particle owned by robot k becoming $S_{k,t} = \left\{\left(\frac{1}{N}.\prod_{f=1}^{M} w^i_{k,f,t}\right),\ x^i_{k,t}, m^i_{k,t-1}\right\}_{i=1}^{N}$ and robot k' becoming $S_{k\prime,t} = \left\{\left(\frac{1}{N}.\prod_{f=1}^{M} w^i_{k\prime,f,t}\right),\ x^i_{k\prime,t}, m^i_{k\prime,t-1}\right\}_{i=1}^{N}$.

4) Each robot k and k' performs map update from m^i_{t-1} to m^i_t using Extended Kalman Filter (EKF) [12] and calculates importance weight of the particles using its previous importance weight $w^i_{k,t} = w^i_{k,t-1}.\left(\frac{1}{N}.\prod_{f=1}^{M} w^i_{k,f,t}\right)$. Now, the set of particles owned by robot k is $S_{k,t} = \left\{w^i_{k,t},\ x^i_{k,t}, m^i_{k,t}\right\}_{i=1}^{N}$ and the set of particles owned by robot k' is $S_{k\prime,t} = \left\{w^i_{k\prime,t},\ x^i_{k\prime,t}, m^i_{k\prime,t}\right\}_{i=1}^{N}$. Each robot k and k' transmits its set of particles to their teammates.
5) Each robot k and k' receives a set of particles from their teammates and the number of sets of particles now becomes two times larger than before $S_{k,t} = \left\{w^i_{k,t},\ x^i_{k,t}, m^i_{k,t}\right\}_{i=1}^{N} \cup \left\{w^i_{k\prime,t},\ x^i_{k\prime,t}, m^i_{k\prime,t}\right\}_{i=1}^{N}$. It is assumed that $m^i_{k,t}$ and $m^i_{k\prime,t}$ have some landmarks index that are overlapping each other $m^i_{k,t} \cap m^i_{k\prime,t} \neq \{\emptyset\}$. Next step, local particles weight $w^i_{k,t}$ and local map $m^i_{k,t}$ needs to be updated into global weight $w^{i,G}_{k,t}$ and global map $m^{i,G}_{k,t}$ by using $S_{k\prime,t}$.
6) For any overlapping landmarks f, each robot computes equation (28) locally.

$$w^{i,G}_{k,t} = w^i_{k,t}.\left(\frac{1}{\sum_{i=1}^{N}\left(w^i_{k,t}+w^i_{k\prime,t}\right)}\right).\prod_{f=1}^{M} w^{i,G}_{k,f,t} \tag{28}$$

Computation of $w^{i,G}_{k,f,t}$ in Gaussian representation is based on equation (29).

$$w^{i,G}_{k,f,t} \simeq \left|2\pi Q^{i,G}_{k,f}\right|^{-\frac{1}{2}}.\exp\{-\frac{1}{2}\left(m^i_{k,f} - m^i_{k\prime,f}\right)^T.{Q^{i,G}_{k,f}}^{-1}.(m^i_{k,f} - m^i_{k\prime,f})\} \tag{29}$$

Here, $m^i_{k,f} - m^i_{k\prime,f}$ is computed based on equation (30).

$$m^i_{k,f} - m^i_{k\prime,f} = \begin{bmatrix} m^i_{k,f,x} - m^i_{k\prime,f,x} \\ m^i_{k,f,y} - m^i_{k\prime,f,y} \\ atan2\left(m^i_{k,f,x} - m^i_{k\prime,f,x}, m^i_{k,f,y} - m^i_{k\prime,f,y}\right) - x^i_{k,t,\theta} \end{bmatrix} \tag{30}$$

Computation of $Q^{i,G}_{k,f}$ is shown on equation (31).

$$Q^{i,G}_{k,f} = \nabla H^{i,G}_{k,f}.\Sigma^{i,G}_{k,f}.\left(\nabla H^{i,G}_{k,f}\right)^T + Q \text{ where}$$

$$\nabla H^{i,G}_{k,f} = \begin{bmatrix} \frac{dx}{d} & \frac{dy}{d} \\ \frac{dy}{d2} & \frac{dx}{d2} \end{bmatrix}; \tag{31}$$

$dx = m^i_{k\prime,f,x} - x^i_{k,t,x}$ and $dy = m^i_{k\prime,f,y} - x^i_{k,t,y}$;

$d2 = dx^2 + dy^2$ and $d = \sqrt{d2}$;

$$\Sigma^{i,G}_{k,f} = \begin{bmatrix} \sqrt{\sigma^2_{k,x}.\sigma^2_{k\prime,x}} & \sqrt{\sigma_{k,x}\sigma_{k,y}.\sigma_{k\prime,x}\sigma_{k\prime,y}} \\ \sqrt{\sigma_{k,x}\sigma_{k,y}.\sigma_{k\prime,x}\sigma_{k\prime,y}} & \sqrt{\sigma^2_{k,y}.\sigma^2_{k\prime,y}} \end{bmatrix}$$

However, the calculations that are given in equations (28) to (31) require rearrangement of the particle index owned by robot k and k'. The set of particles of both robot k and k' are sorted based on their local significant index. Thus, the particle with a high significant index on robot k will be computed with the particle with a high significant index on robot k', and so on.

7) Currently, particle i has two weight values $w_{k,f,t}^{i}$ and $w_{k,f,t}^{i,G}$. Computation of the average of $w_{k,f,t}^{i}$ and $w_{k,f,t}^{i,G}$ uses equation (32).
$$w_{k,f,t}^{i} = \frac{1}{2}\left(w_{k,f,t}^{i} + w_{k,f,t}^{i,G}\right) \quad (32)$$
The computation of equation (32) causes the particle weight ratio in robot k to be disproportionate; therefore, normalized coefficients γ on equation (33) are used to restore the proportionality of particle weights.
$$w_{k,f,t}^{i} = \left(\frac{1}{2}\left(w_{k,f,t}^{i} + w_{k,f,t}^{i,G}\right)\right).\gamma \quad (33)$$
8) Performing map update from $m_{k,ft}^{i}$ to $m_{k,ft}^{i,G}$ using Extended Kalman Filter (EKF) [12]. At the end of this step, robot k owns the global particle representation $S_{k,t}^{G} = \{w_{k,t}^{i,G}, x_{k,t}^{i}, m_{k,t}^{i,G}\}_{i=1}^{N}$. By using computations in equations (28) to (33), the approximation of the average particle weight of robot k and robot k' can be computed in a distributed way.
9) Repeating steps 5 through 8 for other robots that are connected to robot k until some loop of λ. At the end of λ, robot k performs resampling particle [12] locally.

G. Consensus-Based Calculation of Posterior Parameters in Distributed-Multi Robot SLAM

Posterior parameter or joint probability density function (PDF) in SLAM represents the probability of map and robot's position given all set of measurement and all set of input control. Joint PDF in single-robot SLAM is shown in equation (1). In distributed multi-robot SLAM, the global PDF are computed using the local PDF in distributed way by applying consensus algorithm. Assuming robot k and k' are in a close position when communicating, both robots are observing identical environments. It is also assumed that robot k and robot k' already have the same frame reference and the same data association. The implementation steps of consensus-based calculation of posterior parameters in distributed multi-robot SLAM are as follows:

1) At the beginning of time step t, each robot k and k' performs single-robot SLAM locally including resampling particle in order to calculate local posterior of joint distribution SLAM. After performing single-robot SLAM, both robot k and k' own their SLAM local posterior $S_{k,t} = \{w_{k,t}^{i}, x_{k,t}^{i}, m_{k,t}^{i}\}_{i=1}^{N}$ and $S_{k\prime,t} = \{w_{k\prime,t}^{i}, x_{k\prime,t}^{i}, m_{k\prime,t}^{i}\}_{i=1}^{N}$ respectively.
2) Each robot transmits its own local posterior to their teammates. Each robot k and k' receives SLAM local posterior of their teammates and the number of sets SLAM posterior now becomes two times larger than before $S_{k,t} = \{w_{k,t}^{i}, x_{k,t}^{i}, m_{k,t}^{i}\}_{i=1}^{N} \cup \{w_{k\prime,t}^{i}, x_{k\prime,t}^{i}, m_{k\prime,t}^{i}\}_{i=1}^{N}$. It is assumed that $m_{k,t}^{i}$ and $m_{k\prime,t}^{i}$ have some landmarks index that are overlapping each other $m_{k,t}^{i} \cap m_{k\prime,t}^{i} \neq \{\emptyset\}$. Next step, the local SLAM posterior $S_{k,t}$ needs to be updated into global SLAM posterior $S_{k,t}^{G}$ by fusing its own local SLAM posterior and its teammate local SLAM posterior.
3) Calculation on global posterior parameters $S_{k,t}^{G}$ which is executed locally by robot k using set of all information owned by robot k at time t can be written as equation (34).
$$S_{k,t}^{G} = \begin{cases} A, \ if\ f \epsilon\ m\ is\ overlaping \\ B, \ otherwise \end{cases} \quad (34)$$
From equation (34), A and B are computed based on equation (35) and (36) respectively.
$$A = S_{k,t}^{G} = \left(\left[\frac{1}{K}\left(w_{k,t}^{i} + \sum_{k'=1}^{J} w_{k',t}^{i}\right), \frac{1}{K}\left(x_{k,t}^{i} + \sum_{k'=1}^{J} \max_{1\le i\le N} \rho_{k',t}^{i,k}\right), \left\{\frac{1}{K}\left(m_{k,f,t}^{i} + \sum_{k'=1}^{J} m_{k',f,t}^{i}\right), \frac{1}{K}\left(\Sigma_{k,f,t}^{i} + \sum_{k'=1}^{J} \Sigma_{k',f,t}^{i}\right)\right\}_{f=1}^{D}, \left\{\frac{1}{K}\left(\rho_{k,t}^{i,k'} + \sum_{k'=1}^{J} \max_{1\le i\le N} x_{k',t}^{i,k}\right), \frac{1}{K}\left(P_{k,t}^{i,k'} \sum_{k'=1}^{J} P_{k',t}^{i,k}\right)\right\}_{k'=1}^{J}\right]_{i=1}^{N}\right) \quad (35)$$

$$B = S^{G}_{k,t} = \left(\left[\frac{1}{K}\left(w^{i}_{k,t} + \sum_{k'=1}^{J} w^{i}_{k',t}\right), \frac{1}{K}\left(x^{i}_{k,t} + \sum_{k'=1}^{J} \max_{1\le i\le N} \rho^{i,k}_{k',t}\right),\ \left\{m^{i}_{k,f,t}, \Sigma^{i}_{k,f,t}\right\}_{f=1}^{D}, \left\{\frac{1}{K}\left(\rho^{i,j}_{k,t} + \sum_{k'=1}^{J} \max_{1\le i\le N} x^{i,k}_{k',t}\right),\ \frac{1}{K}\left(P^{i,k'}_{k,t} \sum_{k'=1}^{J} P^{i,k}_{k',t}\right)\right\}_{k'=1}^{J}\right]_{i=1}^{N}\right) \tag{36}$$

4) Repeating step three as much as λ times from the consensus iteration. Based on equation (35) and (36), if $\lambda > 1$ then t is replaced with λ.

Next, section three describes how these two consensus algorithms are simulated.

3. Experimental Result

In order to show the performance of the method, we built a simulator of distributed multi-robot SLAM using 4 robots. The robots start from an unknown location and begin the single-robot SLAM using their own local frame reference. The robots explore 100x100 m^2 large of environment that consists of 100 landmarks. The environment that is used in this experiment is shown on figure 3.1.

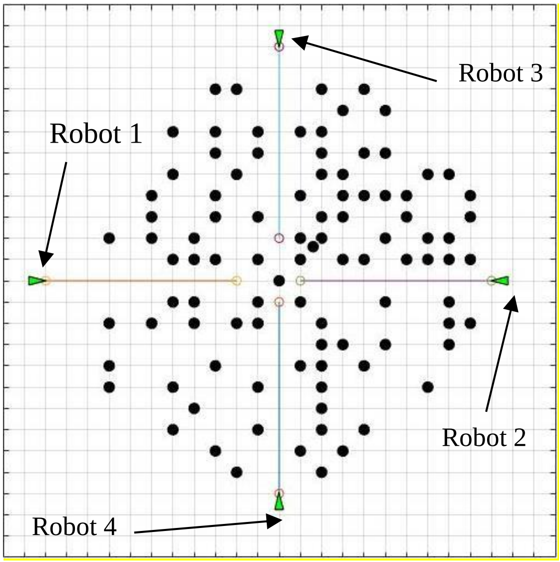


Figure 3. Four robots explore 100x100 m^2 large environment that consists of 100 landmarks.

Robot moves based on its waypoint; the waypoint of each robot is already defined. We assume that each robot uses same motion model and range-bearing observation model; thus, observation noise and system noise are also the same. The robot's motion model and range-bearing sensor model used in the simulator are shown in equation (37) and (38) respectively. The simulation uses velocity v and steering angle ω as the control variable that can be written as equation (39).

$$x' = \begin{bmatrix} x + v\,\Delta t\,cos(\omega + \theta) \\ y + v\,\Delta t\,sin(\omega + \theta) \\ \theta + v\,\Delta t\,sin(\omega)\,/\,h \end{bmatrix} + R_t \tag{37}$$

$$z = \begin{bmatrix} range \\ bearing \end{bmatrix} = \begin{bmatrix} r^{i}_{t} \\ \phi^{i}_{t} \end{bmatrix} = \begin{bmatrix} \sqrt{\left(m_{k,x} - x\right)^2 + \left(m_{k,y} - y\right)^2} \\ atan2\left(m_{k,y} - y, m_{k,x} - x\right) - \theta \end{bmatrix} + Q_t \tag{38}$$

$$u = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{39}$$

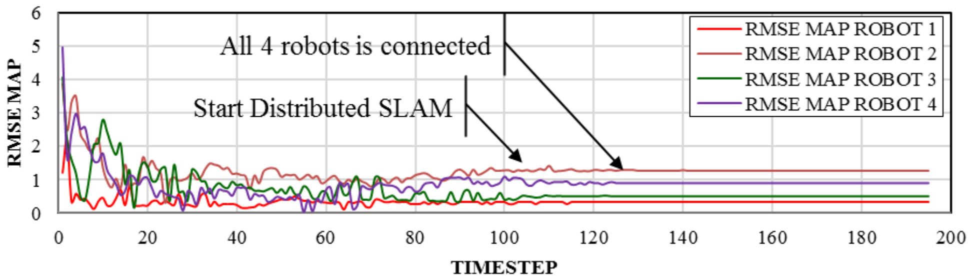
Robot's wheel based h is 4 meters. System noise R_t that is added to the motion model consists of noise to velocity and noise to steering angle. The initial noise to velocity is 1.0 m/s and the initial noise to steering angle is $10\pi/180$ radian. Measurement noise Q_t that is added to the sensor model consists of noise to range and noise to bearing. The maximum observation range is 30 m and the initial noises to observation are 1 m for range and $10\pi/180$ radian for bearing. Time interval between control signals Δt is 0.025 second and time interval between observation is 0.2 second. The maximum distance of communication range between robots is 30 meters. Number of particles in each robot are 25 particles.

A. *RMSE Map*

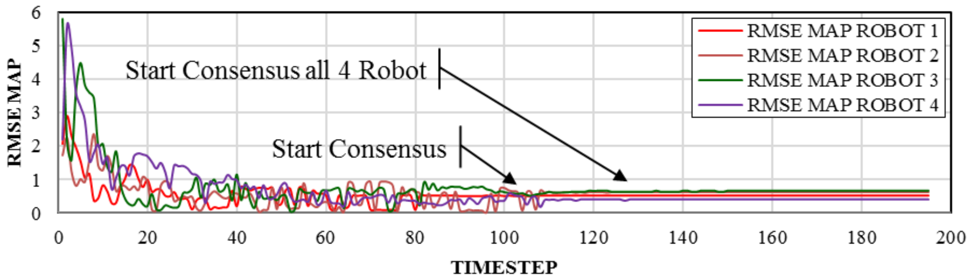
We use root mean square error (RMSE) to show mapping performance of the algorithm. Consider that lm_f is true landmark position and x_f^i is estimated landmark position, RMSE map can be written as equation 40.

$$RMSE\ Map = \frac{1}{F}\sum_{f=1}^{F}\sqrt{\left(lm_f(x) - \max_w x_f^i(x)\right)^2 + \left(lm_f(y) - \max_w x_f^i(y)\right)^2} \qquad (40)$$

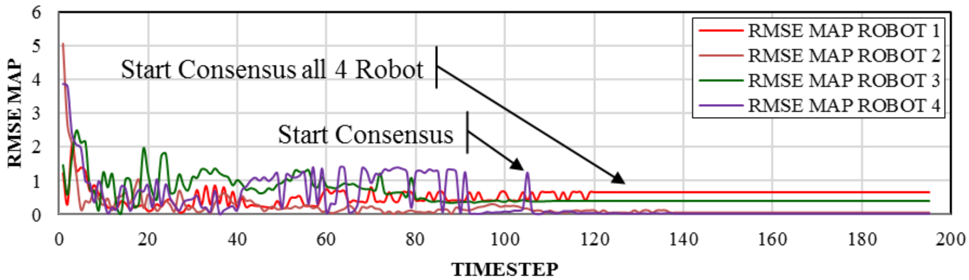
Mapping performance of the consensus particle filtering distributed multi-robot SLAM compared to distributed multi-robot SLAM without consensus are shown in figure 3.2.



a) Distributed Multi Robot SLAM without consensus

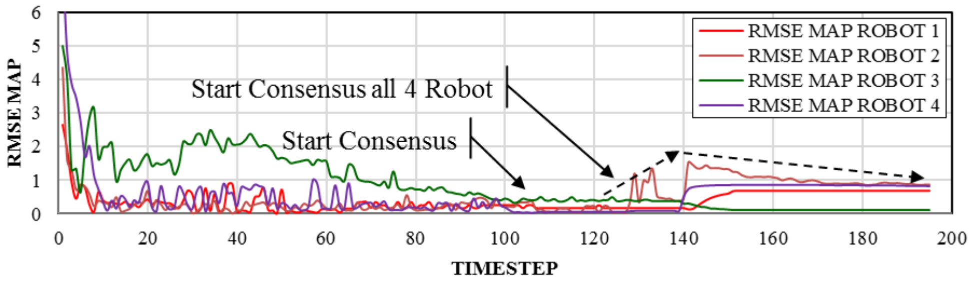


b-i) Distributed Multi Robot SLAM based on consensus particle weight

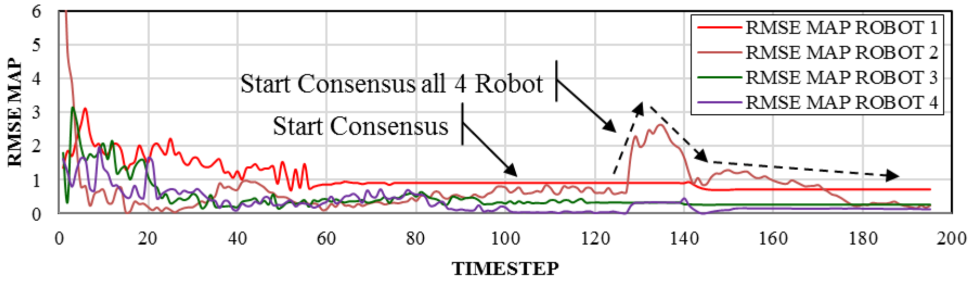


b-ii) Distributed Multi Robot SLAM based on consensus particle posterior

b) Consensus Particle Filtering Distributed Multi Robot SLAM using 5 iterations.



c-i) Distributed Multi Robot SLAM based on consensus particle weight



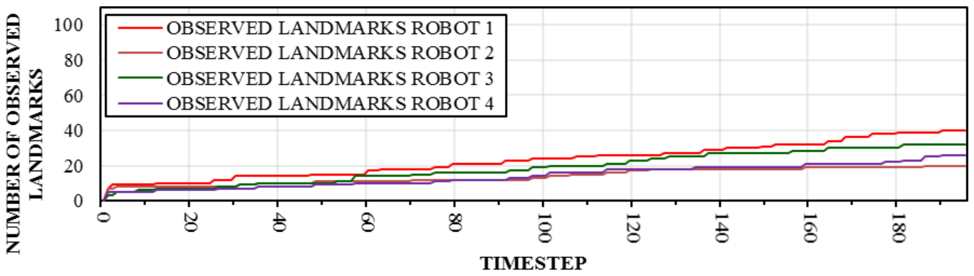
c-ii) Distributed Multi Robot SLAM based on consensus particle posterior

c) Consensus Particle Filtering Distributed Multi Robot SLAM using 20 iterations.

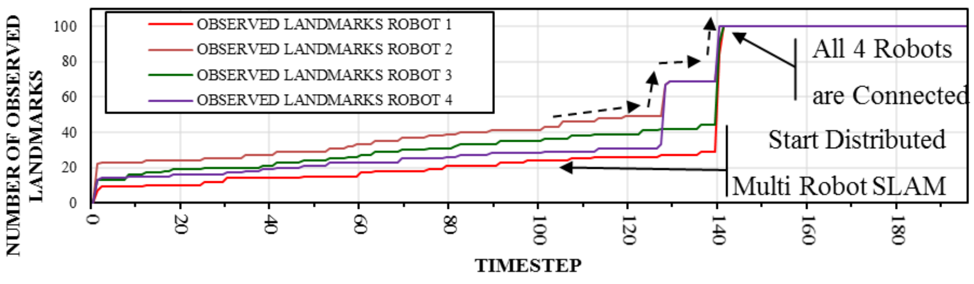
Figure 4. Map RMSE of consensus distributed multi-robot SLAM compared to distributed multi-robot SLAM without consensus (in meter).

B. *Mapping Coverage*

Mapping coverage is the number of landmarks successfully mapped by each robot. Figure 3.3 shows the comparison of the mapping coverage between single-robot SLAM and distributed multi-robot SLAM.



a) Mapping coverage of single-robot SLAM.



b) Mapping coverage of distributed multi robot SLAM.

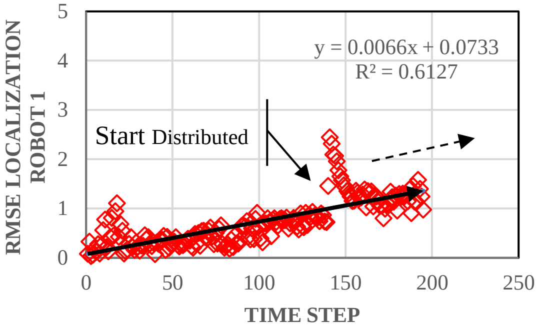
Figure 5. Mapping coverage of distributed multi-robot SLAM compared to single-robot SLAM.

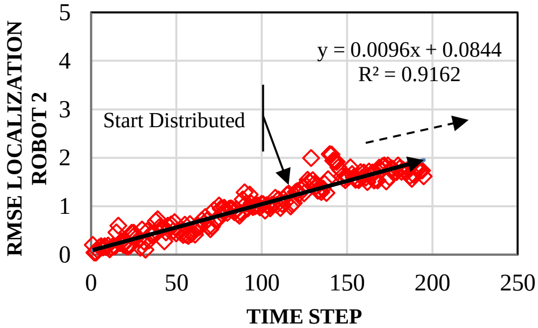
C. RMSE Localization

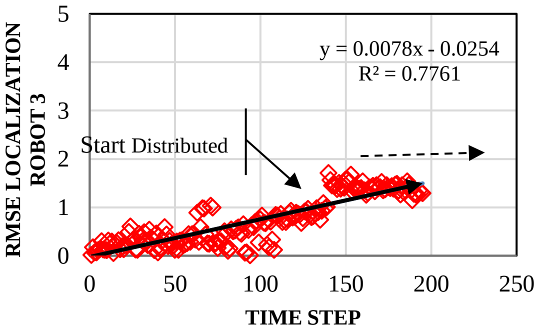
We used root mean square error (RMSE) to show localization performance of the algorithm. Consider that xv_t is true robot position and x_t^i is estimated robot position, RMSE localization can be written as equation 41.

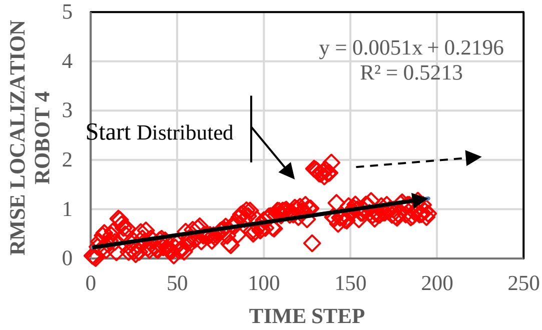
$$RMSE\ Localization = \sqrt{\left(xv_t(x) - \max_w x_t^i(x)\right)^2 + \left(xv_t(y) - \max_w x_t^i(y)\right)^2} \qquad (41)$$

Localization performance of consensus particle filtering distributed multi-robot SLAM compared to distributed multi-robot SLAM without consensus are shown in figure 3.4.

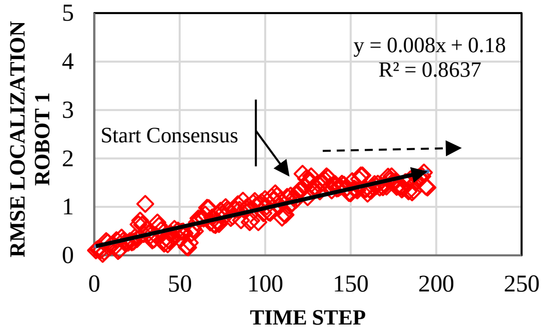


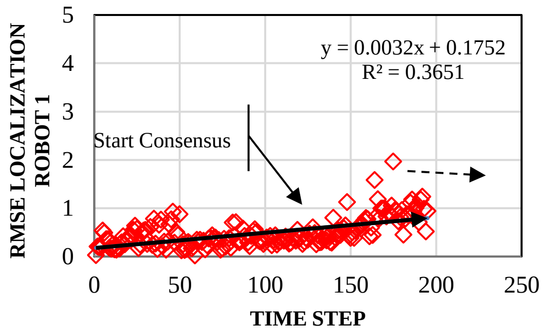


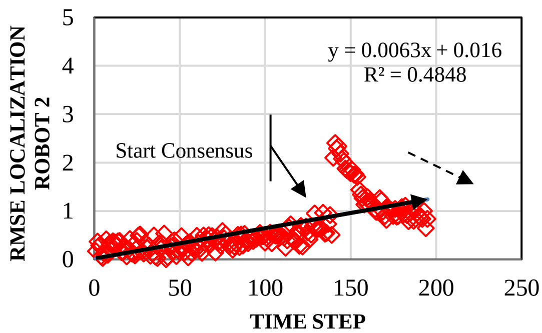


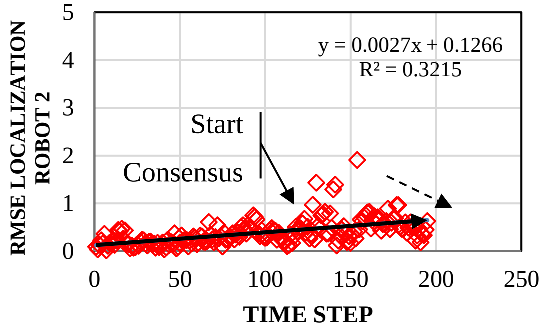


a) Distributed multi robot SLAM without consensus

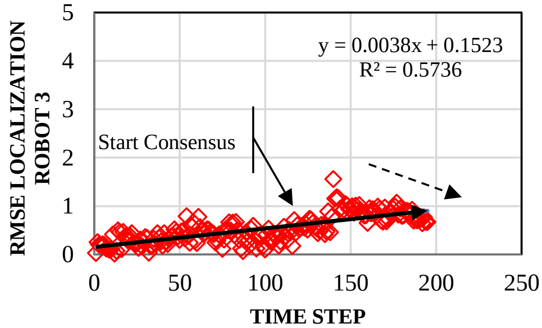




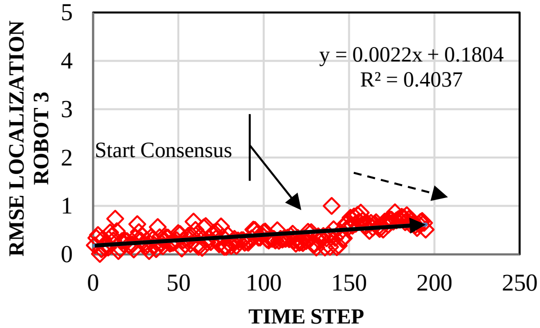




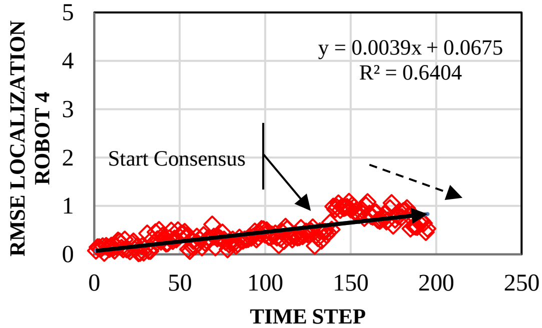
RMSE LOCALIZATION ROBOT 2
TIME STEP
y = 0.0027x + 0.1266
R² = 0.3215
Start Consensus
0 1 2 3 4 5
0 50 100 150 200 250



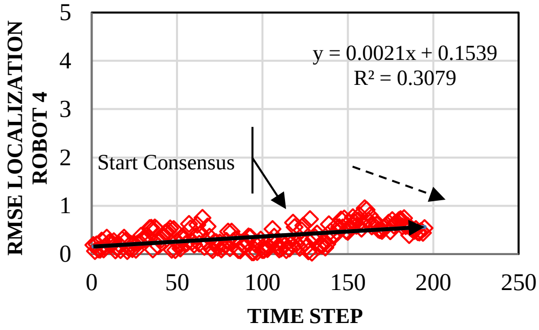
RMSE LOCALIZATION ROBOT 3
TIME STEP
y = 0.0038x + 0.1523
R² = 0.5736
Start Consensus
0 1 2 3 4 5
0 50 100 150 200 250



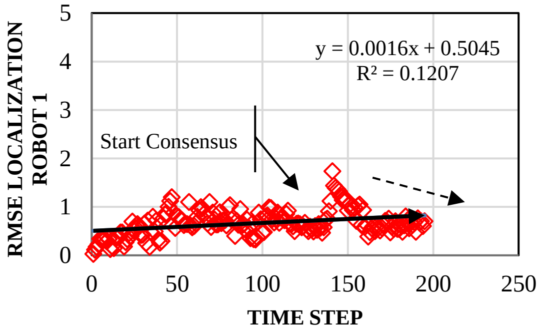
RMSE LOCALIZATION ROBOT 3
TIME STEP
y = 0.0022x + 0.1804
R² = 0.4037
Start Consensus
0 1 2 3 4 5
0 50 100 150 200 250

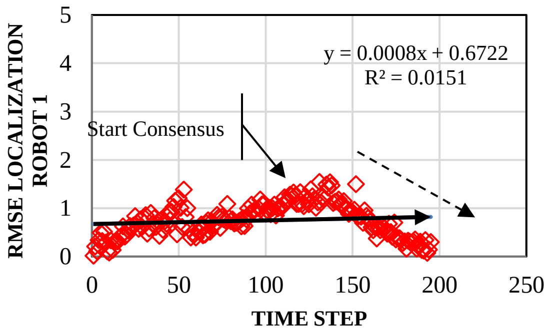


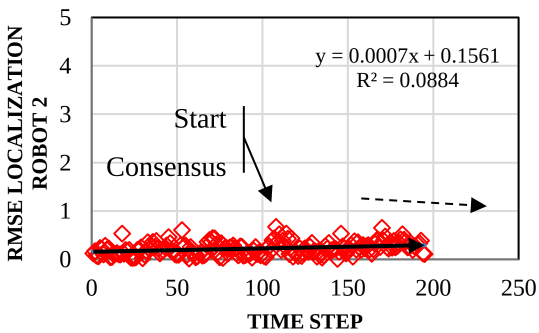
RMSE LOCALIZATION ROBOT 4
TIME STEP
y = 0.0039x + 0.0675
R² = 0.6404
Start Consensus
0 1 2 3 4 5
0 50 100 150 200 250

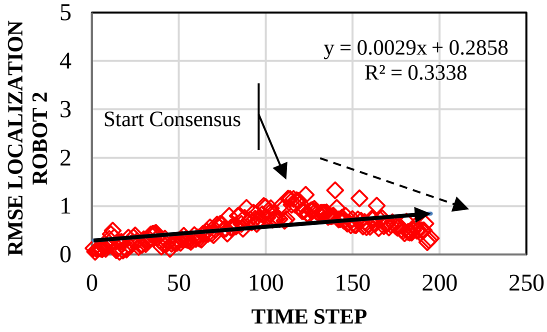


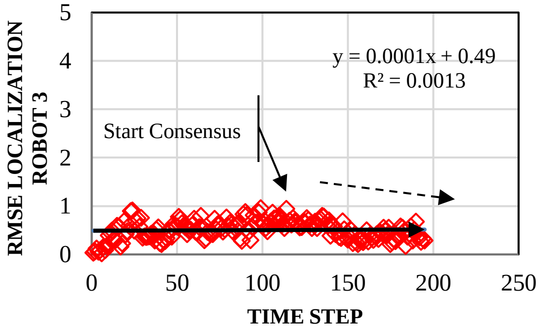
b) Consensus Particle Filtering Distributed Multi Robot SLAM using 5 Iterations. Particle Weight-Based Consensus (left), and Particle Posterior Based Consensus (right).

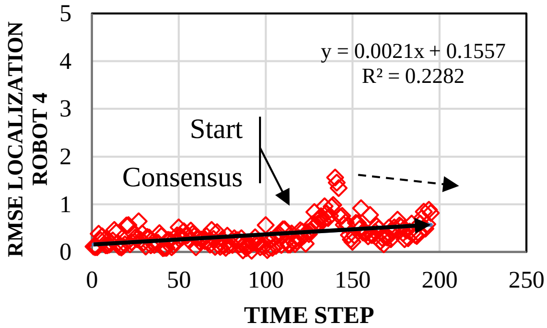


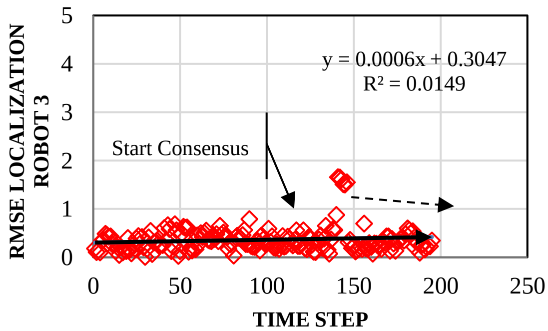


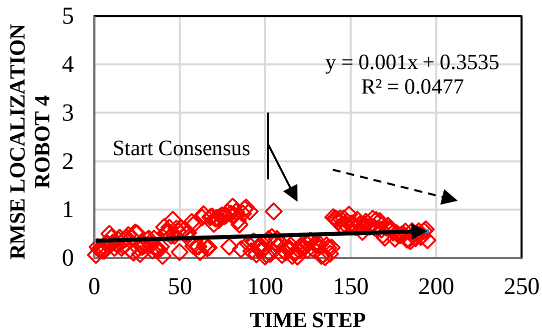



RMSE LOCALIZATION ROBOT 2
y = 0.0029x + 0.2858
R² = 0.3338
Start Consensus
TIME STEP
0
1
2
3
4
5
0
50
100
150
200
250


RMSE LOCALIZATION ROBOT 3
y = 0.0001x + 0.49
R² = 0.0013
Start Consensus
TIME STEP
0
1
2
3
4
5
0
50
100
150
200
250


RMSE LOCALIZATION ROBOT 4
y = 0.0021x + 0.1557
R² = 0.2282
Start Consensus
TIME STEP
0
1
2
3
4
5
0
50
100
150
200
250


RMSE LOCALIZATION ROBOT 3
y = 0.0006x + 0.3047
R² = 0.0149
Start Consensus
TIME STEP
0
1
2
3
4
5
0
50
100
150
200
250

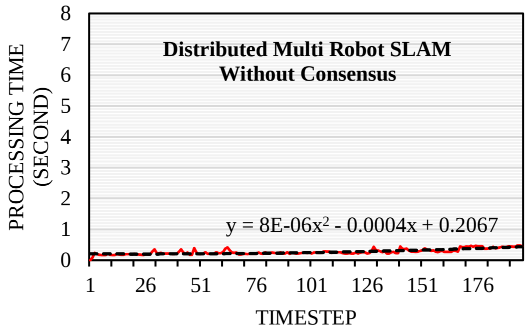


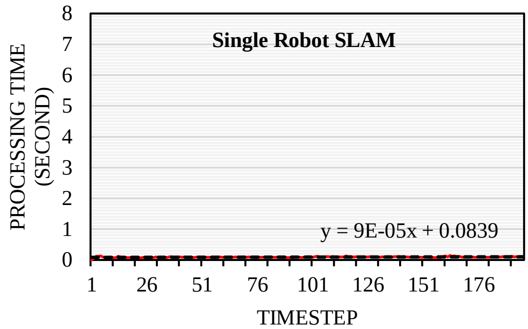
c) Consensus Particle Filtering Distributed Multi Robot SLAM using 20 iterations. Particle weight-based consensus (left), and particle posterior based consensus (right).

Figure 6. Localization RMSE of consensus distributed multi-robot SLAM compared to distributed multi-robot SLAM without consensus (in meter).

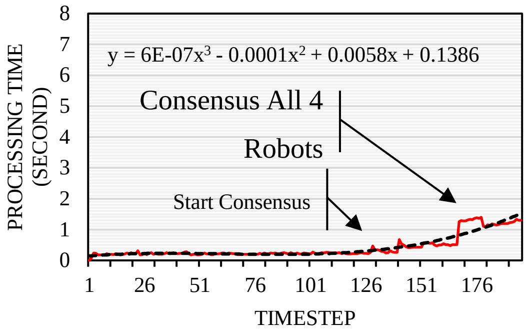
D. Processing Time

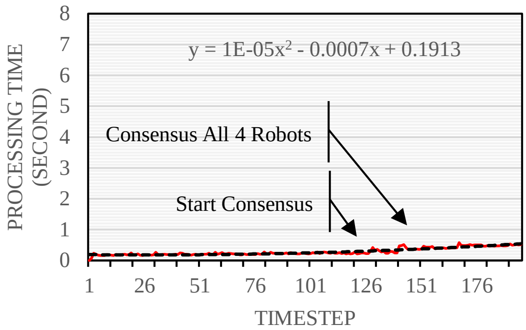
We also measure the processing time to show the performance of the algorithms. In this simulation, we use computer with specifications 2 GHz quad core processor, 12 GB memory, and 2 GB graphic processor. The processing time of the consensus particle filtering distributed multi-robot SLAM compared to distributed multi-robot SLAM without consensus and single robot SLAM are shown in figure 3.5.



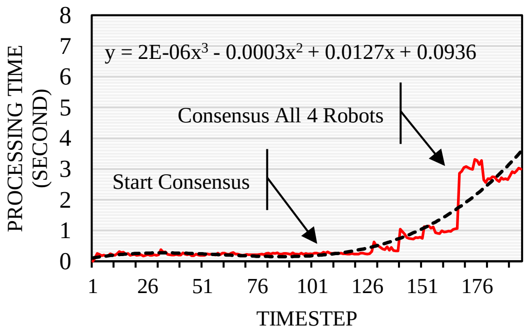


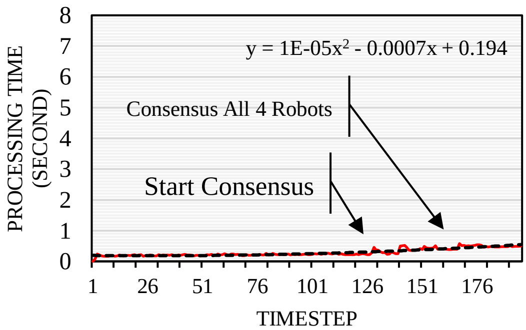
a) Processing time of Distributed Multi-Robot SLAM without consensus (left) and processing time of Single-Robot SLAM (right)



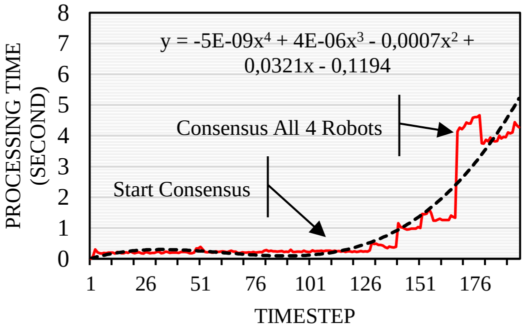


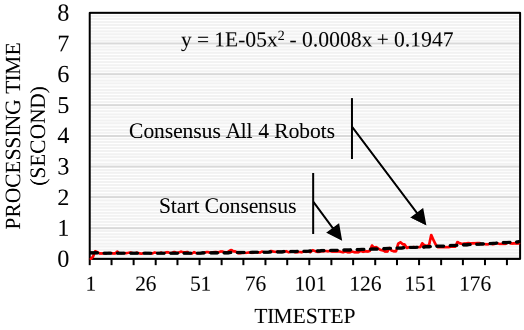
b) Processing time of Consensus Based Distributed Multi-Robot SLAM using 5 iterations Particle weight-based consensus (left), and particle posterior based consensus (right)



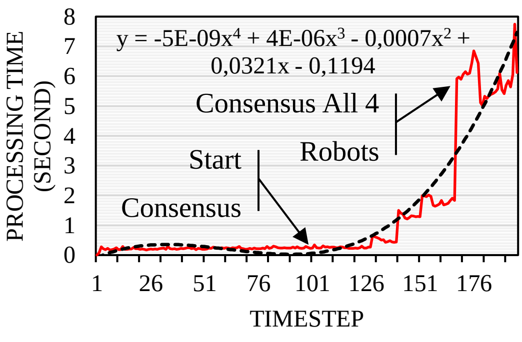


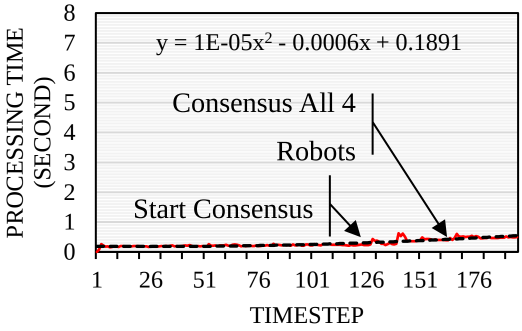
c) Processing time of Consensus Based Distributed Multi-Robot SLAM using 10 iterations. Particle weight-based consensus (left), and particle posterior based consensus (right).





d) Processing time of Consensus Based Distributed Multi-Robot SLAM using 15 iterations. Particle weight-based consensus (left), and particle posterior based consensus (right).





e) Processing time of Consensus Based Distributed Multi-Robot SLAM using 20 iterations.
Particle weight-based consensus (left), and particle posterior based consensus (right).
Figure 7. Processing time of consensus distributed multi-robot SLAM compared to distributed multi-robot SLAM without consensus and single-robot SLAM (in second).

.

4. Discussion

We concentrated on testing the proposed methods on four aspects: 1) root mean square error (RMSE) of map to see the map estimation performance, 2) RMSE of localization to see the robot's pose estimation performance, 3) mapping coverage to see how quickly all features can be observed, and 4) processing time to see how long the computation process takes place per timestep.

A. RMSE Map

Although not significantly different, consensus-based distributed multi-robot SLAM gives a smaller error than distributed multi-robot SLAM without consensus. An increase in error may occur at the beginning of the consensus process, but the longer the robot is connected the error will decrease. RMSE map of each robot on both consensus distributed multi-robot SLAM is under 1 meter, while RMSE map of each robot on distributed multi-robot SLAM without consensus is slightly higher.

We also examined the effect of the number of consensus iterations on maps estimation result. Both consensus distributed multi-robot SLAM are executed using two different number of iteration, that is, 5 iterations and 20 iterations. After applying higher iteration values, we found that not all robots earn a significant decrease in map error compared to the result of using a smaller iterations number. An increase in the number of consensus iterations influences the reduction of map errors on some robots.

B. Mapping Coverage

Mapping coverage of distributed multi-robot SLAM is larger than single-robot SLAM. Each robot on distributed multi-robot SLAM is able to map all 100 landmarks after timesteps number 140, while on single robot SLAM there is not a single robot able to map all 100 landmarks. Even though all four maps obtained by each robot on a single-robot can be combined with the merging map method, but this requires a further processing to determine the starting point of the merging.

C. RMSE Localization

A significant decrease in localization error appears in both consensus-based distributed multi-robot SLAM compared with distributed multi-robot SLAM without consensus. On distributed multi-robot SLAM without consensus we found that after a robot is connected to other robot, the localization error still has an upward trend. It is contrary to the results in both consensus-based distributed multi-robot SLAM that have a downward trend. We also found that the higher iteration number used affected to lower localization error. Based on these results, it can be concluded that the both consensus-based distributed multi-robot SLAM method is able to provide better localization estimation result compared with distributed multi-robot SLAM without consensus.

D. Processing Time

Distributed multi-robot SLAM requires a longer processing time compared to single-robot SLAM, especially when a robot is connected to another robot. The more number of robots that communicate at the same time require longer processing time. Distributed multi-robot SLAM based on consensus particle weight requires the longest processing time. The processing duration on distributed multi-robot SLAM based on consensus particle weight influenced by not only the number of robots communicating at the same time but also the number of landmarks observed by each communicating robot. Distributed multi-robot SLAM based on consensus posterior parameter and distributed multi-robot SLAM without consensus deliver better result on processing time compared with distributed multi-robot SLAM based on consensus particle weight.

5. Conclusion

In this paper, we have presented two different methods of consensus-based distributed multi robot SLAM. Both methods can be used to map the common environment on condition motion model and observation model each robot is the same. We also test the proposed methods on four aspects: 1) root mean square error (RMSE) of map to see the map estimation performance, 2) RMSE of localization to see the robot's pose estimation performance, 3) mapping coverage to see how long it will take until all features are observed, and 4) processing time to see how long the computation process takes place per timestep. The result is both of our proposed consensus-based distributed multi-robot SLAM method deliver better performance on map estimation and pose estimation compared with distributed multi robot SLAM without consensus. However, distributed multi-robot SLAM based on consensus particle weight requires longer processing time compared to other methods. Both of our proposed consensus-based distributed multi-robot SLAM methods require active local communication on each robot. The analysis of communication loads and computational loads is still an issue that should be investigated further.

6. Acknowledgement

This work was funded by Ministry of Research, Technology and Higher Education of the Republic of Indonesia. The authors gratefully acknowledge them for providing the funding.

7. References

[1]. S. Bruno dan K. Oussama, Handbook of Robotics, Secaucus, NJ: Springer-Verlag New York, Inc., 2007.
[2]. S. Thrun, W. Burgard dan D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, 2005.
[3]. A. Howard, "Multi-robot Simultaneous Localization and Mapping using Particle Filters," *International Journal of Robotics Research,* vol. 25, no. 2, pp. 1243--1256, December 2006.
[4]. Grisetti, C. Stachniss dan W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics,* vol. 23, no. 1, pp. 34 - 46, February 2007.
[5]. L. Carlone, M. K. Ng, J. Du, B. Bona dan M. Indri, "Rao-Blackwellized Particle Filters multi robot SLAM with unknown initial correspondences and limited communication," dalam *Robotics and Automation (ICRA), 2010 IEEE International Conference on,* Anchorage, Alaska, USA, 2010.
[6]. G. Grisetti, R. Kummerle, C. Stachniss dan W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intelligent Transportation Systems Magazine,* vol. 2, no. 3, pp. 31 - 43, 31 January 2011.

[7]. A. Angeli, S. Doncieux, J.-A. Meyer dan D. Filliat, “Visual topological SLAM and global localization,” dalam *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009.
[8]. A. P. Gee, D. Chekhlov, A. Calway dan W. Mayol-Cuevas, “Discovering Higher Level Structure in Visual SLAM,” *IEEE Transactions on Robotics,* vol. 25, no. 5, pp. 980 - 990, 2008.
[9]. G. Zhang, J. H. Lee, J. Lim dan I. H. Suh, “Building a 3-D Line-Based Map Using Stereo SLAM,” *IEEE Transactions on Robotics,* vol. 31, no. 6, pp. 1364 - 1377, 2015.
[10]. S.-Y. Hwang dan J.-B. Song, “Monocular Vision-Based SLAM in Indoor Environment Using Corner, Lamp, and Door Features from Upward-Looking Camera,” *IEEE Transactions on Industrial Electronics,* vol. 58, no. 10, pp. 4804 - 4812, 2011.
[11]. W. Y. Jeong dan K. M. Lee, “Visual SLAM with Line and Corner Features,” dalam *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.
[12]. M. Montemerlo, S. Thrun, D. Koller dan B. Wegbreit, “FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem,” dalam *Eighteenth National Conference on Artificial Intelligence*, Edmonton, Alberta, Canada, 2002
[13]. M. Montemerlo, S. Thrun, D. Roller dan B. Wegbreit, “FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges,” dalam *Proceedings of the 18th international joint conference on Artificial intelligence (IJCAI'03),* San Francisco, CA, USA, 2003.
[14]. M. Adams, B.-N. Vo, R. Mahler dan J. Mullane, “SLAM Gets a PHD: New Concepts in Map Estimation,” *IEEE Robotics & Automation Magazine,* vol. 21, no. 2, pp. 26 - 37, 2014.
[15]. H. Durrant-Whyte dan T. Bailey, “Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms,” *IEEE ROBOTICS AND AUTOMATION MAGAZINE,* vol. 2, pp. 99-108, June 2006.
[16]. W. Burgard, M. Moors, D. Fox, R. Simmons dan S. Thrun, “Collaborative multi-robot exploration,” dalam *Robotics and Automation Proceedings, ICRA*.
[17]. D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz dan B. Stewart, “Distributed Multirobot Exploration and Mapping,” *Proceedings of the IEEE,* vol. 94, no. 7, pp. 1325 - 1339, July 2006.
[18]. S. R. u. N. Jafri, L. Brayda dan R. Chellali, “Distributed Feature based RBPF Multi Robot SLAM,” dalam *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics*, Phuket, Thailand, 2011.
[19]. A. Gil, Ó. Reinoso, M. Ballesta dan M. Juliá, “Multi-robot visual SLAM using a Rao-Blackwellized particle filter,” *Elsevier Robotics and Autonomous Systems,* vol. 58, no. 1, pp. 68 - 80, January 2010.
[20]. S. Carpin, A. Birk dan V. Jucikas, “On map merging,” *Robotics and Autonomous Systems,* vol. 53, pp. 1-14, 2005.
[21]. S. B. Williams, G. Dissanayake dan H. Durrant-Whyte, “Towards multi-vehicle simultaneous localisation and mapping,” dalam *Proceedings 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, USA, 2002.
[22]. A. Cunningham, M. Paluri dan F. Dellaert, “DDF-SAM: Fully Distributed SLAM using Constrained Factor Graphs,” dalam *The 2010 IEEE/RSJ International Conference*, Taipei, Taiwan, 2010.
[23]. L. Carlone, M. K. Ng, J. Du, B. Bona dan M. Indri, “Simultaneous Localization and Mapping Using Rao-Blackwellized Particle Filters in Multi Robot Systems,” *Springer Journal of Intelligent & Robotic Systems,* vol. 63, no. 2, pp. 283 - 307, 2011.
[24]. R. Aragues, J. Cortes dan C. Sagues, “Distributed consensus algorithms for merging feature-based maps with limited communication,” *Elsevier Robotics and Autonomous Systems,* vol. 59, no. 3 - 4, pp. 163 - 180, 2011.

[25]. R. Aragues, J. Cortes dan C. Sagues, “Distributed Consensus on Robot Networks for Dynamically Merging Feature-Based Maps,” *IEEE TRANSACTIONS ON ROBOTICS,* vol. 28, no. 4, pp. 840 - 854, 2012.
[26]. O. Hlinka, F. Hlawatsch dan P. M. Djuric, “Distributed Particle Filtering in Agent Networks: A survey, classification, and comparison,” *IEEE Signal Processing Magazine,* vol. 30, no. 1, pp. 61 - 81, 2013.
[27]. F. Hidayat, B. R. Trilaksono dan H. Hindersah, “Distributed Multi Robot Simultaneous Localization and Mapping with Consensus Particle Filtering,” *Journal of Physics: Conference Series,* vol. 801, no. 1, 2017.

Fadhil Hidayat is a Ph.D. student in the School of Electrical Engineering and Informatics at the Bandung Institute of Technology, Indonesia. He received his M.S. degree in Information Technology from the Bandung Institute of Technology in 2011. Since December 2012, he has been with the Bandung Institute of Technology as a teacher and researcher. His current research interests include distributed multi-robot system, smart city, smart transportation, robot for smart city, and management of information security.

Bambang Riyanto Trilaksono was born in Banyuwangi, Indonesia on November 1962. He received his Bachelor degree in Electrical Engineering from ITB, and Master and Doctorate degree from Waseda University, Japan. He is now a Professor at School of Electrical and Informatics Engineering, ITB. His research interests include robust control, intelligent control & intelligent systems, discrete event systems, control applications, telerobotics, embedded control systems, and robotics.

Hilwadi Hindersah, He graduated his Doctoral program from Institut Teknologi Bandung. His employment as a lecturer and researcher in School of Electrical Engineering and Informatics ITB. His fields of interest are vision-based robot control, design interaction, and adaptive control.