Fully Decentralized Cooperative Localization of a Robot Team: An Efficient and Centralized Equivalent Solution

Hua Mu, Mengyuan Dai, Ming Gao, and Xianfei Pan

College of Mechatronics and Automation,
National University of Defense Technology, Hunan 410073, P.R. China
huamu08@gmail.com

Abstract. This paper presents an efficient, centralized equivalent and fully decentralized solution to the cooperative localization of mobile robot teams. Formulating the cooperative localization problem in the framework of Bayesian estimation, the decentralized solution is designed by interlacing the calculation steps of *prediction* and *update* in a proper sequence. In the proposed solution, each robot fuses only the sensor data relevant to itself; information is shared among the robots by a chain communication topology. The solution yields linear minimum mean-square error estimates, equivalent to a centralized extended Kalman filter. There is no information redundancy and computation duplication among the robots. The solution can also be viewed from the perspective of implementing inference on a specific junction tree. The performance of the proposed algorithm is evaluated with simulation experiments.

Keywords: cooperative localization, decentralized solution, mobile robot team, Bayesian estimation, junction tree.

1 Introduction

In recent years, the cooperative localization of a robot team has received significant attention from the robotics community[1-3]. When the same external landmarks are used by different robots for self-localization or inter-robot observations are made, the states of the relevant robots are correlated. Cooperative localization is a joint estimation problem state over the time-history of the robot team, allowing the robots to share the localization resources and achieve higher overall performance.

A centralized solution to the cooperative localization problem is straightforward yet expensive and lack of robustness. Various decentralized solutions are proposed by many researchers. A decentralized Kalman filter approach is proposed in [4]. The essence of this method is to distribute the expectation vector and the covariance matrix of the joint probability among the robots; each robot is allocated its state expectation and the covariance block-row related to itself. The prediction of the covariance block-row involves each robot *broadcasting* the state transition matrix. The fusion of each measurement, either a measurement regarding only one robot or an

J. Lee et al. (Eds.): ICIRA 2013, Part I, LNAI 8102, pp. 520–531, 2013.

inter-robot measurement, updates the parameters allocated to each robot, which requires the observer and the observed robot to *broadcast* the measurement information and the covariance block-rows. The operation scales poorly with the number of the robots and the number of the measurements. Actually, the update operation in standard Kalman filter framework is difficult to be distributed since the states are correlated to each other.

The information filter with delayed states [5] is appealing since both the prediction and the update are local operations to each robot and can be easily distributed. The moment recovery, however, manipulates global information and seems to be an obstacle to a decentralized solution. The solution proposed in [6] employs the incremental Cholesky modifications for efficient moment recovery. The sparse Cholesky factors of the joint state estimate are pipelined from robot to robot and the moment recovery for a group of states is accomplished at some robot who accumulates enough Cholesky factors. The solution is communication expensive since the Cholesky factors are duplicated at each robot in fact. The solution presented in [7] reduces the communication cost at the price of lower localization accuracy. In [7], each robot maintains a set of information parameters with the same size of the parameters for the joint delayed states. Each robot only integrates data from its own sensors. When an inter-robot measurement occurs, the two robots exchange their stored data and fuse the data based on convex combination. The moment recovery can be implemented in a centralized manner at each robot. The result is consistent and unbiased, yet not a minimum mean square error estimate.

Very recently another decentralized solution is presented in [8]. It achieves centralized-equivalent estimates, yet requiring a robot to transmit all odometric data to its neighbors, and to duplicate the data fusion effort.

In [9], a decentralized approach based on junction tree algorithm is designed to solve the cooperative localization problem. In the decentralized formulation, the local sensor data at each robot are integrated into potentials of the cliques of junction trees; the information is shared among the robots through message passing between cliques. The information parameterization of Gaussian distributions is utilized. The decentralized junction tree approach actually provides a general framework, within which there are a number of implementations for a fixed problem due to multiple choices of the junction tree.

This paper introduces a decentralized solution to the cooperative localization problem which could be deduced within the framework of the junction tree. Different from the work in [9], this work utilizes the moments form parameterization. The moments parameters have explicit physical meaning and the moments parameterization avoids the troubles brought by the moment recovery. We point out that the solution can be illustrated outside the context of the junction tree, instead from the perspective of Bayesian estimation.

2 Problem Formulation

This section formulates the problem of cooperative localization in the framework of Bayesian estimation. Consider a team of mobile robots. Each robot is equipped with dead-reckoning sensors to measure self-motion; odometer or inertial measurement unit for example. Some robots carry sensors that provide relative position measurements (range and/or bearing) among robots or between robots and the environment; such as cameras, laser range-finders. Some robots can correct its dead-reckoning error. In the case that the environment map is known, the measurements between a robot and the environment can be utilized to bound the dead-reckoning error. Otherwise, the sensor like GPS can provide absolution localization information. The robots can communicate with each other. The cooperative localization task is to estimate the state (position, orientation, velocity etc.) of each robot making use of both its own observations and those observations made by and of other robots. The robots are identified by capitals A, B, C, etc .

2.1 State Space Model

The state of robot A at time instance k is denoted by x_k^A . Assume the motion model of robot A is given as

$$x_{k+1}^A = f(x_k^A, u_k) + G_k w_k \tag{1}$$

where u_k is the system input at time step k , w_k is the process noise, $w_k \sim \mathbb{N}_m(w_k; 0, Q)$, and G_k is a matrix with proper dimensions. $\mathbb{N}_m(v; \mu, P)$ represents a Gaussian distribution over v with mean μ and covariance P .

The GPS-like measurement of robot A at time step k is denoted z_k^A . The observation model is given as

$$z_k^A = h_1(x_k^A) + r_{1k} \tag{2}$$

where r_{1k} is the measurement noise, $r_{1k} \sim \mathbb{N}_m(r_{1k}; 0, R_1)$.

The measurement that robot B makes to robot A at time step k is denoted by z_k^{BA} . The observation model is defined as

$$z_k^{BA} = h_2(x_k^A, x_k^B) + r_{2k} \tag{3}$$

where r_{2k} is the measurement noise, $r_{2k} \sim \mathbb{N}_m(r_{2k}; 0, R_2)$.

If functions $f(\cdot)$, $h_1(\cdot)$ and $h_2(\cdot)$ are nonlinear, Taylor expansions are used to linearize the models to make the inference tractable. The linearized system is given in Table 1., where μ_k^A is the best available mean estimate of x_k^A .

Table 1. The linearized space model

model	linearized model	meaning of symbols
(1)	$x_{k+1}^A = f(\mu_k^A, u_k) + F(x_k^A - \mu_k^A) + G_k w_k$	$F = \frac{\partial f}{\partial x_k^A}\Big\vert_{x_k^A = \mu_k^A}$
(2)	$z_k^A = h_1(\mu_k^A) + H_1(x_k^A - \mu_k^A) + r_{1k}$	$H_1 = \frac{\partial h_1}{\partial x_k^A}\Big\vert_{x_k^A = \mu_k^A}$
(3)	$z_k^{BA} = h_2(\mu_k^A, \mu_k^B) + H_2^A(x_k^A - \mu_k^A) + H_2^B(x_k^B - \mu_k^B) + r_{2k}$	$H_2^A = \frac{\partial h_2}{\partial x_k^A}\Big\vert_{x_k^A=\mu_k^A, x_k^B=\mu_k^B}$, $H_2^B = \frac{\partial h_2}{\partial x_k^B}\Big\vert_{x_k^A=\mu_k^A, x_k^B=\mu_k^B}$

Assume the initial state of each robot is a Gaussian variable. The linearized system in Table 1. defines the following conditional distributions

$$p(x_{k+1}^A \,\big|\, x_k^A) = \mathbb{N}_m(x_{k+1}^A; F x_k^A + f(\mu_k^A, u_k) - F\mu_k^A, \bar{Q}) \tag{4}$$

$$p(z_k^A \,\big|\, x_k^A) = \mathbb{N}_m(z_k^A; H_1 x_k^A + h_1(\mu_k^A) - H_1\mu_k^A, R_1) \tag{5}$$

$$p(z_k^{BA} \,\big|\, x_k^A, x_k^B) = \mathbb{N}_m(z_k^{BA}; H_2^A x_k^A + H_2^B x_k^B + h_2(\mu_k^A, \mu_k^B) - H_2^A \mu_k^A - H_2^B \mu_k^B, R_2) \tag{6}$$

where $\bar{Q} = G_k Q G_k^T$.

2.2 Bayesian Estimation

Let Z_k denote the collection of the measurements made at time step k, including the inter-robot measurements and the intro-robot measurements. Let $\mathbb{Z}_k$ denote the collection of the measurements up to time step k, namely, $\mathbb{Z}_k = \{Z_1, Z_2, \cdots Z_k\}$. The task of cooperative localization is to estimate the posterior state for each robot, which can be given by $p(x_k^A, x_k^B, x_k^C, \cdots \big| \mathbb{Z}_k)$. The estimation is implemented iteratively as follows:

prediction: $$p(x_{k+1}^{R1}, \cdots, x_{k+1}^{RN} \big| \mathbb{Z}_k) = \int p(x_k^{R1}, \cdots, x_k^{RN}) p(x_{k+1}^{R1} \mid x_k^{R1}) \cdots p(x_{k+1}^{RN} \mid x_k^{RN}) dx_k^{R1} \cdots dx_k^{RN} \tag{7}$$

update: $$p(x_{k+1}^{R1}, \cdots, x_{k+1}^{RN} \mid \mathbb{Z}_{k+1}) = \frac{p(x_{k+1}^{R1}, \cdots, x_{k+1}^{RN}) \prod_{i=1}^{N} p(z_{k+1}^{Ri} \mid x_{k+1}^{Ri}) \prod_{j \neq i} p(z_{k+1}^{RiRj} \mid x_{k+1}^{Ri}, x_{k+1}^{Rj})}{\prod_{i=1}^{N} p(z_{k+1}^{Ri}) \prod_{j \neq i} p(z_{k+1}^{RiRj})} \tag{8}$$

In each cycle, the computation is composed of two parts: *prediction* and *update*. In the *prediction* part, the joint state probability is propagated from time step $k-1$ to time step k . In the *update* part, the inter-robot measurements and the intro-robot measurements are fused. With the Gaussian assumption, the inference for the linearized system can be accomplished by the extended Kalman filter (EKF), providing the expectation and covariance of the probability distribution of the joint states.

Inspecting that both *prediction* and *update* are actually multi-step calculations, a fully decentralized solution is proposed in the paper by arranging the calculation steps of *prediction* and *update* in a proper sequence. In the decentralized solution, *prediction* is accomplished by each robot predicting its own state and *update* is accomplished by each robot fusing the measurements relevant to itself; the state predictions and the measurements fusions are implemented by turns. The messages passed between robots are the joint probabilities where only part of robots doing predictions and measurement updates. The posterior joint probability is obtained by a certain robot.

3 A Fully Decentralized Solution: An Example

Example. Consider the cooperative localization of three robots which are labeled as *A, B* and *C*. Each robot is equipped with an odometry. Robot *A* is equipped with a GPS receiver to obtain its position. Robot *B* has a laser range-finder which can make range measurements to the other two robots (See Fig.1). The three robots can communicate with each other by a local wireless network.

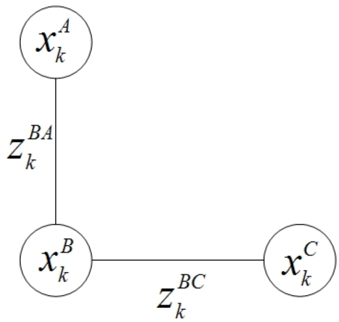

Fig. 1. The inter-robot measurements

3.1 Estimation at Time Step $k = 1$

In the beginning, the robot states are irrelevant to each other. The state prediction and the fusion of the intro-robot measurement are implemented in the same way at each robot. Take robot *A* for example.

$$\textit{prediction}: \quad p(x_1^A) = \int p(x_0^A)\, p(x_1^A \mid x_0^A)\, dx_0^A$$

$$\textit{update}\text{(if needed)}: \quad p(x_1^A \,|\, z_1^A) = \frac{p(x_1^A)\, p(z_1^A \,|\, x_1^A)}{p(z_1^A)}$$

To fuse the inter-robot measurements, the joint state probability should be constructed and this involves the communication of the state estimation between robots. The joint state can be constructed incrementally if the robots communicate in a chain. An inter-robot measurement can be fused either at the observer or the observed when the two relevant states are included into the joint state. The order of the robots on the communication chain is not critical. Assume robot *B* sends a message of $p(x_1^B)$ to robot *C* and robot *C* constructs a joint state as follows,

$$p(x_1^B, x_1^C) = p(x_1^B)p(x_1^C) \tag{9}$$

Then robot *C* can fuse the observation z_1^{BC}:

$$p(x_1^B, x_1^C \,|\, z_1^{BC}) = \frac{p(x_1^B, x_1^C)p(z_1^{BC} \,|\, x_1^B, x_1^C)}{p(z_1^{BC})} \tag{10}$$

Robot *C* then sends its estimation to Robot *A*. After receiving the estimation message, robot *A* carries on the joint state construction and the inter-robot measurement fusion, just like robot *C*.

$$p(x_1^A, x_1^B, x_1^C \,|\, z_1^A, z_1^{BC}) = p(x_1^A \,|\, z_1^A)p(x_1^B, x_1^C \,|\, z_1^{BC}) \tag{11}$$

$$p(x_1^A, x_1^B, x_1^C \,|\, z_1^A, z_1^{BC}, z_1^{BA}) = \frac{p(x_1^A, x_1^B, x_1^C \,|\, z_1^A, z_1^{BC})p(z_1^{BA} \,|\, x_1^B, x_1^A)}{p(z_1^{BA})} \tag{12}$$

3.2 Estimation with Correlated States

The robot states become correlated due to the fusion of the inter-robot measurement, which is clearly shown in the above subsection. Without loss of generality, we assume the states are correlated to each other when estimating the states of time step $k+1$ ($k \geq 1$). The prior joint state is represented by $p(x_k^A, x_k^B, x_k^C \,|\, \mathbb{Z}_k)$.

The robot team satisfies the following assumptions before implementing the $k+1$ estimation:

1. The prior is held by a robot, say robot *A*.
2. The intro-robot measurements are held by the robot who makes the measurements. That is to say, there is no communication among the robots about the intro-robot measurements.
3. The inter-robot measurements are held by the two relevant robots. For example, the measurement z_k^{BA} is stored at both robot *A* and robot *B*. This requires a measuring robot to send the inter-robot measurement information to the observed robot.

Assume $Z_k = \{z_k^A, z_k^{BA}, z_k^{BC}\}$. Again the robots need to communicate in a chain. It's better for robot A to start the estimation since it holds the prior joint state. The decentralized solution places no constraint on the locations of the other two robots at the communication chain. Take the communication topology of $A \to B \to C$ for example. The computations implemented by each robot in the fully decentralized solution are as follows.

Robot A:

prediction: $$p(x_k^B, x_k^C, x_{k+1}^A \mid \mathbb{Z}_k) = \int p(x_k^A, x_k^B, x_k^C \mid \mathbb{Z}_k) p(x_{k+1}^A \mid x_k^A) dx_k^A \tag{13}$$

update: $$p(x_k^B, x_k^C, x_{k+1}^A \mid \mathbb{Z}_k, z_{k+1}^A) = \frac{p(x_k^B, x_k^C, x_{k+1}^A \mid \mathbb{Z}_k) p(z_{k+1}^A \mid x_{k+1}^A)}{p(z_{k+1}^A)} \tag{14}$$

Robot B:

prediction: $$p(x_{k+1}^B, x_k^C, x_{k+1}^A \mid \mathbb{Z}_k, z_{k+1}^A) = \int p(x_k^B, x_k^C, x_{k+1}^A \mid \mathbb{Z}_k, z_{k+1}^A) p(x_{k+1}^B \mid x_k^B) dx_k^B \tag{15}$$

update: $$p(x_{k+1}^B, x_k^C, x_{k+1}^A \mid \mathbb{Z}_k, z_{k+1}^A, z_{k+1}^{BA}) = \frac{p(x_{k+1}^B, x_k^C, x_{k+1}^A \mid \mathbb{Z}_k, z_{k+1}^A) p(z_{k+1}^{BA} \mid x_{k+1}^B, x_{k+1}^A)}{p(z_{k+1}^{BA})} \tag{16}$$

Robot C:

prediction: $$p(x_{k+1}^B, x_{k+1}^C, x_{k+1}^A \mid \mathbb{Z}_k, z_{k+1}^A, z_{k+1}^{BA}) = \int p(x_{k+1}^B, x_k^C, x_{k+1}^A \mid \mathbb{Z}_k, z_{k+1}^A, z_{k+1}^{BA}) p(x_{k+1}^C \mid x_k^C) dx_k^C \tag{17}$$

update: $$p(x_{k+1}^B, x_{k+1}^C, x_{k+1}^A \mid \mathbb{Z}_{k+1}) = \frac{p(x_{k+1}^B, x_{k+1}^C, x_{k+1}^A \mid \mathbb{Z}_k, z_{k+1}^A, z_{k+1}^{BA}) p(z_{k+1}^{BC} \mid x_{k+1}^B, x_{k+1}^C)}{p(z_{k+1}^{BC})} \tag{18}$$

At the end of $k+1$ estimation, robot C achieves the posterior state estimation for each robot. It can send the results to the other robots. To estimate the states at next time step, it's better to start the calculations from robot C to avoid unnecessary communication.

4 A Fully Decentralized Solution: General Cases

4.1 Perspective of the Bayesian Estimation

The solution for the example of three robots in Section 3 can be generalized to a team of N robots. The observer who made an inter-robot measurement is required to send a measurement message to the observed robot, which ensures the measurement can always be fused by one of the two robots. Except the communication of the inter-robot measurements, the communication of the state estimation requires a chain topology. The only thing a robot should know about other robots is the IDs (identifications) of its neighbors. In the applications where there exists a fixed communication chain, the neighbors' IDs can be downloaded to each robot as prior information. In the cases where it is hard to find a fixed communication chain due to the movements of the robots, the communication chain can be built dynamically for different estimation cycles. Each robot acts in a modular way described as follows.

1. For time step $k=1$:
-- Predicting its own state since the states are irrelevant at the moment.
-- Updating its own state by fusing intro-robot measurements if there is any.
-- If there is an estimation message sending from a neighbor:
---- receiving the message and then calculates the joint state probability;
---- updating the joint state by fusing the inter-robot measurements that the relevant states appear in the joint state.
-- Sending an estimation message to a neighbor with whom it has not communicated any estimation message in this estimation cycle. If a robot can not find such a neighbor, the estimation finishes up and the robot is identified as the starting robot for the next estimation cycle.
2. For time step $k+1$ ($k=1,2,\cdots$):
If the robot is the starting robot or if the robot receives an estimation message from its neighbor:
--Predicting its own state in the joint state probability.
--Updating the joint state by fusing intro-robot measurements if there is any.
--Updating the joint state by fusing the inter-robot measurements that the relevant states appear in the joint state.
-- Sending an estimation message to a neighbor with whom it has not communicated any estimation message in this estimation cycle. If a robot can not find such a neighbor, the estimation finishes up and the robot is identified as the starting robot for the next estimation cycle.

In the Bayesian estimation framework, both the *prediction* and the *update* are composed of multiple steps. The decentralized solution interlaces these steps to make each robot fuse its local sensor data and share its information across the team with proper communication strategy. For a linearized system model, the estimates generated by the decentralized solution achieve the same accuracy as a centralized EKF, if given the same data. In this sense, the decentralized solution is optimal since it yields minimum mean-square error estimation.

4.2 Perspective of the Junction Tree Algorithm

The proposed solution is actually a special implementation of the junction tree solution in [9]. The junction tree used for inference is specific. Firstly, the cliques of the junction tree are formulated using the elimination order as follows. The measurement variables are eliminated before the state variables and the state variables at time step $k-1$ are eliminated before the states at time step k. The elimination order of the state variables at time step $k-1$ determines the communication chain of robots. The construction of a junction tree is still not unique from the cliques. An optimal junction tree in terms of the minimum communication cost among the robots can be determined by making the initialization of cliques local to each robot. The junction tree for the cooperative localization at time step k for the example in Section 3 is depicted in Fig.2.

The standard Hugin strategy is adopted as the message passing strategy for the inference on the junction tree. Assume the states of robots are irrelevant with each other at the beginning. The overall decentralized solution is implemented in terms of Hugin algorithm as follows.

1. *Initialization*

Each robot is informed the IDs of its neighbors in the communication chain. The potentials of the measurement cliques are initialized as the measurement likelihoods.

For $k=1$, the state potentials of all the robots are initialized in the same way. Take robot A for example. The state potential of robot A is initialized as $p(x_0^A)p(x_1^A \mid x_0^A)$.

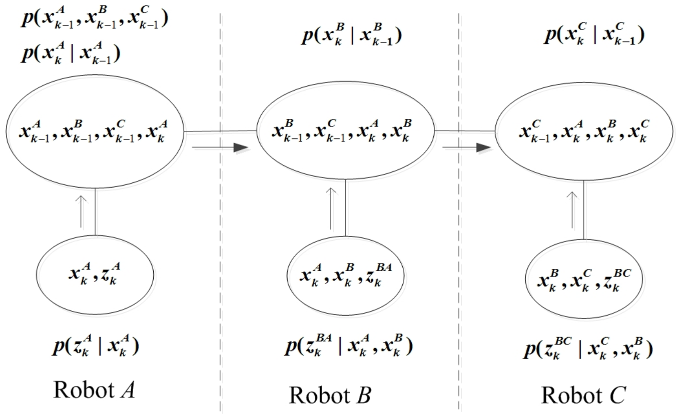

Fig. 2. The junction tree for inference for time step k ($k \geq 1$) for the example in Section 3, with the initial potentials of the cliques given for $k \geq 2$. The dotted lines illustrates the allocation of the cliques to the robots. The arrows with dark head stand for the message passing involving communication between robots, while the arrows with simple head represent the message passing between cliques held by the same robot.

For $k \geq 2$, the state potential of the robot who holds the prior, take robot A for example, is initialized as $p(x_{k-1}^A, x_{k-1}^B, \cdots \mid \mathbb{Z}_{k-1}) p(x_k^A \mid x_{k-1}^A)$. The state potential for other robots, take robot B for example, is initialized as $p(x_k^B \mid x_{k-1}^B)$.

2. *Inference,* $k \geq 1$

When a robot makes an inter-robot measurement, it sends a measurement message to the observed robot. Each robot manipulates its state potential by the following operations.

(1) Multiplying in the estimation message passed to it if there is any, and denoting the ID of the robot who passed the message to it as X.

(2) Incorporating the measurement likelihoods if there is any, including intro-robot measurements and inter-robot measurements.

(3) Eliminating the older own state from the state potential. If there is another neighbor different from X, formulating an estimation message from the resultant potential and transmitting to that neighbor. Otherwise, the resultant potential gives the required state estimates of time instance k for all the robots. The state of this robot should be eliminated first when constructing junction tree for next cycle inference.

5 Performance Evaluation and Analysis

In this section, we perform simulations to validate our approach and analyze the performance in terms of accuracy, modularity, scalability and robustness.

5.1 Simulation

A conventional two-wheel veihcle model is used and the trajectories of robots are generated by setting waypoints. Three robots (labeled with *A, B* and *C*) are considered. Assume each robot has odometer for motion measurement. Robot *A* and robot *B* each has a range finder which can make range measurements to other robots lying within 150m. Robot *B* also has a GPS receiver. The standard deviations are: initial positions, 8m; velocity measurements, 0.6m/s; steering angle measurements, 10degree; GPS measurements, 2m; range measurements, 0.1m. Data association for the range observations is assumed known.

Fig. 3 provides the robot trajectories with and without cooperative localization, together with the ground truth. The cooperative localization errors for robot *A* in x-axis and y-axis are given in Fig.4, together with the 3σ uncertainty. There is no inter-robot range measurement relevant to robot A before 15s, since the distances to the other robots are larger than 150m, beyond the limit of the range finder. After 15s, the localization error for robot *A* is reduced sharply and maintained small due to the fusion of the range measurements. The benefit of the cooperative localization is obvious. The accuracy of our decentralized solution is the same with a centralized EKF.

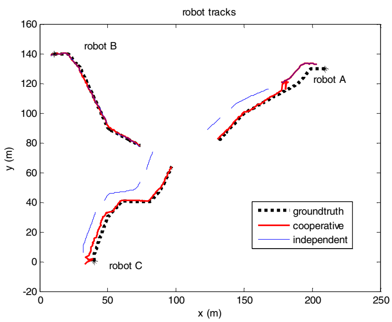

Fig. 3. The robot trajectories

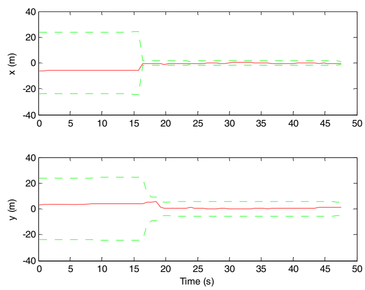

Fig. 4. The cooperative localization error for robot *A*

5.2 Analysis

The proposed solution has following advantages.

(1) Accuracy. The proposed solution is optimal in terms of the linearized system model of the cooperative navigation problem. However, the problem itself is actually a nonlinear problem, especially the observation equations of the inter-robot measurements. The loss of the accuracy due to the linearization can be reduced utilizing the relinearization technique. Each iteration of relinearization repeats the whole estimation process except the linearization points are updated as estimates of last iteration.

(2) Modularity. No global knowledge of the robots team is required a prior, each robot can be constructed and programmed in a modular fashion, allowing dynamic system configuration. In the decentralized cooperative navigation solution, the communication requirements between robots are allowed to change dynamically, as long as the whole team can communicate in a chain structure.

(3) Scalability. The proposed solution has good scalability especially among the centralized-equivalent solutions. The computation of data fusion is distributed to each robot pretty much evenly without any duplication. No broadcast is involved. The communication load is not only lower than most of the other centralized-equivalent solutions, but also is comparable with the solution in [7]. The communication of the inter-robot measurements in the two solutions is the same. The bulk of the communication of estimates is $2m(Nd)^2$ for the solution in [7] and $(N-1)(Nd)^2$ for the solution in this work, where m is the number of the inter-robot measurements, N is the number of the robots, and d is the state dimension for single robot. When it satisfies $m > (N-1)/2$, which are common cases, our solution has lower communication costs.

(4) Robustness. The communication chain can be built dynamically in each estimation cycle, imposing the least requirements on the connectivity of the robots. Each robot has a joint state estimate for the team, at most one estimation cycle earlier. As a result the cooperative localization could be restarted at any other robot when a robot fails.

6 Conclusions

In the paper, a novel decentralized solution to the cooperative localization problems of robot teams is investigated. The solution is designed by interlacing the calculation steps of *prediction* and *update* in a proper sequence. In the proposed solution, each robot fuses only the sensor data relevant to itself and shares estimation information with its neighbors on a chain communication topology. The solution is centralized-equivalent, in the sense that it yields linear minimum mean-square error estimates, equivalent to a centralized extended Kalman filter given the same data. The solution has good scalability since there is no information redundancy and computation duplication among the robots. Each robot acts in a modular manner and the solution is robust to robot failure. The solution can also be viewed from the perspective of implementing inference on a specific junction tree.

A simulation scenario of three ground robots is designed to verify our approach. Our simulated system has time-synchronous measurements. However, the decentralized algorithm applies equally to real-world asynchronous systems, in which case time-alignment is performed by projecting forward the platform states (according to the motion model) to match the observation time-stamps. This is the same approach as is routinely applied in centralized filtering systems. The approach is not limited to the scenario described here; it can be applied to multiple aerial vehicles and multiple underwater vehicles as well. Also, the decentralized structure appears to be a general paradigm for decentralized estimation, like cooperative target tracking, cooperative exploration, etc.

References

1. Bahr, A., Leonard, J.J., Maurice, F.F.: Cooperative Localization for Autonomous Underwater Vehicles. The International Journal of Robotics Research 28(6), 714–728 (2009)
2. Sharma, R.: Bearing-only Cooperative-Localization and Path-Planning of Ground and Aerial Robots. PhD thesis, Brigham Young University (2011)
3. Savic, V., Zazo, S.: Cooperative localization in mobile networks using nonparametric variants of belief propagation. Ad Hoc Networks 11(1), 138–150 (2013)
4. Roumeliotis, S.I., Bekey, G.A.: Distributed Multi-Robot Localization. IEEE Transactions on Robotics and Automation 18(5), 781–795 (2002)
5. Eustice, R.M., Singh, H., Leonard, J.J.: Exactly Sparse Delayed-State Filters for View-Based SLAM. IEEE Trans. on Robotics 22(6), 1100–1114 (2006)
6. Mu, H., Bailey, T., Durrant-Whyte, H., et al.: Decentralised Solutions to the Cooperative Multi-Platform Navigation Problem. IEEE Transactions on Aerospace and Electronic Systems 47(2), 1433–1449 (2011)
7. Cristofaro, A., Renzaglia, A., Martinelli, A. (eds.): Distributed Information Filters for MAV Cooperative Localization. Distributed Autonomous Robotic Systems. Springer, Heidelberg (2013)
8. Leung, K.Y.K., Barfoot, T.D., Liu, H.H.T.: Decentralized localization of sparsely-communicating robot networks: A centralized equivalent approach. IEEE Transactions on Robotics 26(1), 62–77 (2010)
9. Mu, H., Wu, M., Ma, H., Wu, W.: A Decentralized Junction Tree Approach to Mobile Robots Cooperative Localization. In: Liu, H., Ding, H., Xiong, Z., Zhu, X. (eds.) ICIRA 2010, Part I. LNCS, vol. 6424, pp. 736–748. Springer, Heidelberg (2010)