Decentralised Cooperative Localisation for Heterogeneous Teams of Mobile Robots

Tim Bailey, Mitch Bryson, Hua Mu*, John Vial, Lachlan McCalman and Hugh Durrant-Whyte
Australian Centre for Field Robotics
University of Sydney, NSW, Australia
Email: tbailey@acfr.usyd.edu.au
*College of Mechatronic Engineering and Automation
National University of Defense Technology, P.R.China

***Abstract*— This paper presents a distributed algorithm for performing joint localisation of a team of robots. The mobile robots have heterogeneous sensing capabilities, with some having high quality inertial and exteroceptive sensing, while others have only low quality sensing or none at all. By sharing information, a combined estimate of all robot poses is obtained. Inter-robot range-bearing measurements provide the mechanism for transferring pose information from well-localised vehicles to those less capable.**

In our proposed formulation, high frequency egocentric data (e.g., odometry, IMU, GPS) is fused locally on each platform. This is the distributed part of the algorithm. Inter-robot measurements, and accompanying state estimates, are communicated to a central server, which generates an* optimal *minimum mean-squared estimate of all robot poses. This server is easily duplicated for full redundant decentralisation. Communication and computation are efficient due to the sparseness properties of the information-form Gaussian representation. A team of three indoor mobile robots equipped with lasers, odometry and inertial sensing provides experimental verification of the algorithms effectiveness in combining location information.

I. Introduction

Estimating the position and heading of each platform in a team of mobile robots is a fundamental capability for autonomous cooperation. However, a heterogeneous team may consist of some robots with high cost, high accuracy localisation sensors, and others with low cost sensors, and others perhaps with no form of exteroception at all. This paper presents an efficient means to compute a joint estimate of all robot poses. The team shares its information allowing more able robots to assist those with lower quality instrumentation.

A centralised solution to the cooperative localisation problem is simple yet expensive. Each robot transmits its raw measurements to a central server, which uses a conventional Kalman filter to compute an optimal estimate of all platform poses. This is expensive because the motion data (e.g., odometry, IMU) of individual robots tends to be high-bandwidth, resulting in high communications costs and computational burden at the server. Also, a centralised solution lacks robustness since the entire system fails if the server fails.

This paper presents a distributed solution, whereby each platform processes the bulk of their sensed data locally. A central server is employed to fuse measurements involving multiple platforms, but this phase deals with relatively low-bandwidth data. It is straightforward to decentralise this server also, in the sense of redundancy, creating multiple duplicate servers to increase system robustness [3]. The algorithm is optimal insofar as it is equivalent to a centralised EKF or EKS given the same data (although communication lag will slightly effect model linearisation). Efficient communication is achieved by local (ie., decentralised) fusion of high-rate sensor data and using a sparse information-form representation to transmit summary state estimates.

The format of the paper is as follows. The next section discusses previous work related to cooperative navigation. Section III presents the details of the new algorithm. Section IV describes an experimental scenario involving three indoor robots that demonstrates how cooperative localisation transfers location information from a well-instrumented robot to one without exteroceptive sensors. Section V gives concluding remarks and future research directions.

II. Related Work

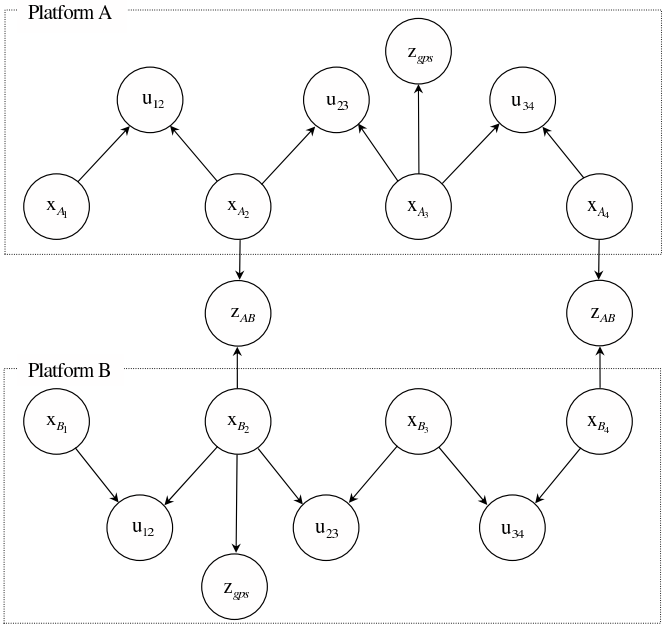
Early work on cooperative localisation required coordination of the motion of the robot team, such that a subset would remain stationary to act as landmarks or beacons as the rest of the team moved forward [9]. This notion of stationary landmarks was later relaxed and robots could track each other while in motion using distributed statistical estimators, such as maximum-likelihood over a finite time-horizon [8] or Monte Carlo filters [7]. The estimators were distributed by having each platform maintain an estimate of its own trajectory subject to inter-robot observations, but with the assumption that the trajectory estimate from one platform is uncorrelated to the estimate of the others; an assumption that eventually causes over-confidence and inconsistency in the estimation process.

Consistent distributed estimation built on the EKF has been demonstrated for cooperative localisation with both indoor [17] and outdoor [11] mobile robots. Cross-correlation terms of the joint covariance matrix are shared amongst the platforms after each update step, which is an operation that scales poorly with the number of robots in the team. An alternative EKF-based approach has each platform maintain a bank of EKFs [1]. Each filter corresponds to an interaction with one of the other platforms, and book-keeping of the

cross-correlations between platforms ensures consistent data fusion. This scheme is computationally expensive as the total number of filters grows exponentially with the size of the team, and suboptimal as the book-keeping strategy permits fusion of only a subset of the available information. The algorithm in [10] makes use of all available information, but at the expense of transmitting all odometric data to its neighbours, and duplicating the data fusion effort. Another alternative [14] employs a maximum a posteriori estimator that is distributed amongst the platforms and solved using conjugate gradients. Although iterative methods such as this have potential for cooperative localisation, the presented solution has several drawbacks, primarily the requirement for synchronous communication between platforms and the large number of communications steps performed throughout the estimation process.

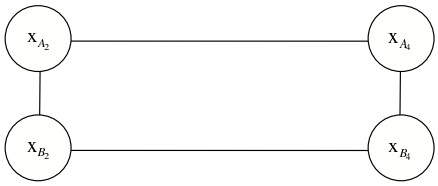
While the Gaussian probability distribution is most commonly represented by its moments—the mean vector and covariance matrix—it may also be represented by another pair of sufficient statistics—the information vector and information matrix (or inverse covariance matrix). The off-diagonal terms of the covariance matrix indicate *marginal* dependency, while the off-diagonals of the information matrix indicate *conditional* dependency. Consequently, the former is typically dense while the latter is often sparse for many temporal and spatial estimation problems. The information-form is exploited in recent simultaneous localisation and mapping (SLAM) solutions, where an information matrix is tractable for large numbers of vehicle poses and map features because it explicitly accounts for the sparsity in the map-to-map and map-to-vehicle relationships [6], [5], [2]. This sparse representation of conditional dependencies is also applied to distributed inference in sensor networks [15] and multi-sensor single-target tracking [12]. These implementations address communication network structures for exchanging information between platforms and methods for data fusion, such as the junction tree algorithm [16].

The cooperative localisation problem is more general than SLAM or conventional distributed sensor networks as the entire system is dynamic; both the sensor platforms and the tracked targets are in motion. A version of information-form cooperative localisation is presented in [13], where sparse Cholesky factors of the joint state estimate are pipelined from platform to platform. This approach has higher lag and computational expense than the current work, but in the current work we apply the idea of incremental Cholesky operations for efficient fusion and moment recovery. The current work extends the concepts in [3], which employs the sparse information-form and has a two-stage partitioning of data fusion—one for egocentric data and another for inter-robot measurements. This strategy facilitates decentralised processing and asynchronous communications, but duplicates the cost of fusing inter-robot measurements on every platform. The current work simplifies the method of generating local state estimates and reconstructing the joint state, and presents a flexible single-server arrangement that easily extends to multiple decentralised servers if required.



(a) Directed graphical model of all states and data

(b) Local Markov chain for platform A



(c) Server estimate

Fig. 1. Graphical models for a two-robot cooperative localisation example. The system (a) is composed of pose states ($\mathbf{x}_k$), egocentric data local to each platform (motion $\mathbf{u}_{k-1,k}$ and position $\mathbf{z}_{gps}$), and inter-robot measurements ($\mathbf{z}_{AB}$). The egocentric data is fused locally to form a Markov chain (b), and relevant pose estimates are sent to a central server, where inter-robot measurements are fused (c).

III. An Algorithm for Distributed Cooperative Localisation

Cooperative localisation is a joint estimation problem over the time-history of a robot team. To illustrate, Fig. 1(a) shows a two-robot system. The algorithm presented in this paper distributes evaluation of this graphical model, and is substantially more efficient, in communication and computation, than any existing exact solution. Each platform's egocentric measurements are fused locally to create a Markov chain of robot pose estimates, as in Fig. 1(b), which is independent of all other platforms. These independent estimates become dependent when coupled by inter-platform measurements (e.g., range-bearing), which are fused at the central server, see Fig. 1(c).

All operations are asynchronous, but depend on synchronised clocks on all robots and accurate timestamping of measurements. Notably, inter-robot measurement events

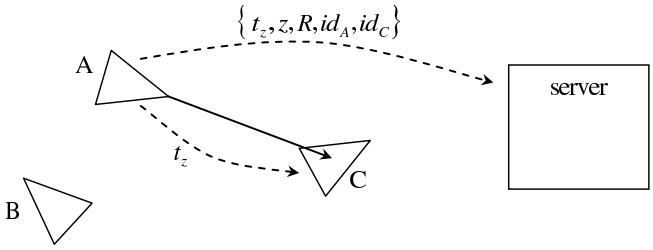


Fig. 2. When robot A makes an inter-robot measurement of C, it sends the measurement timestamp t_z to C, and a measurement packet to the server composed of the timestamp, the measurement and its covariance, and the two platform identifiers. This interaction has no effect or dependence on robot B; all communication and computation is asynchronous.

drive the estimation process, as shown in Fig. 2, wherein if platform A makes an observation of platform C, it sends the event timestamp to the observed platform (C) and sends a packet of measurement data to the central server. The maximum communication lag (t_L) for platform A to inform platform C of the event at time t_z delays the processing of C's egocentric data, and so introduces a time horizon for local fusion, as discussed in Section III-B.

A. Notation for Information-Form Estimation

A random vector $\mathbf{x}$ with Gaussian uncertainty is usually represented in terms of its two moments $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})$, where $\hat{\mathbf{x}}$ is the mean vector and $\mathbf{P}$ is the covariance matrix. In this work, we instead use the *information-form* representation of a Gaussian $\mathcal{N}_I(\mathbf{x}; \hat{\mathbf{y}}, \mathbf{Y})$, where $\hat{\mathbf{y}}$ is the information vector and $\mathbf{Y}$ is the information matrix. These terms are related to the *moment-form* as $\mathbf{Y} \triangleq \mathbf{P}^{-1}$ and $\hat{\mathbf{y}} \triangleq \mathbf{Y}\hat{\mathbf{x}}$. Given a joint state vector,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}, \tag{1}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are sub-vectors, the information-form representation of a Gaussian probability distribution over these states is given by

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{12}^T & \mathbf{Y}_{22} \end{bmatrix}. \tag{2}$$

The information-form possesses additivity and sparsity properties that facilitate cheap distributed estimation. It may be exactly represented as an undirected graph, as shown in Fig. 3. The information vector and the diagonal terms of the information matrix are represented by graph nodes, such that the quantities $\{\hat{\mathbf{y}}_i, \mathbf{Y}_{ii}\}$ are given by node $\mathbf{x}_i$. The off-diagonal terms of the information matrix form edges in the graph, such that a non-zero value for element $\mathbf{Y}_{ij}$ gives an edge connecting nodes $\mathbf{x}_i$ and $\mathbf{x}_j$.

B. Local Augmentation and Fusion Operations

Local processing on each platform involves constructing a Markov chain of robot pose estimates over time. Each odometry measurement ($\mathbf{u}_{k-1,k}$) *augments* the chain, appending a new pose state according to a motion model as follows

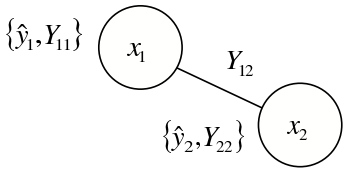


Fig. 3. Graphical model of (2). The nodes represent the relevant parts of the information vector and (block-)diagonal terms of the information matrix, and the edges represent non-zero off-diagonal terms of the information matrix.

$$\mathbf{x}^a = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 = \mathbf{f}(\mathbf{x}_2, \mathbf{u}_{23}) \end{bmatrix}. \tag{3}$$

In information-form the estimate is augmented as

$$\hat{\mathbf{y}}^a = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 - \nabla \mathbf{f}_{\mathbf{x}_2}^T \mathbf{U}^{-1}[\mathbf{f}(\hat{\mathbf{x}}_2, \mathbf{u}) - \nabla \mathbf{f}_{\mathbf{x}_2} \hat{\mathbf{x}}_2] \\ \mathbf{U}^{-1}[\mathbf{f}(\hat{\mathbf{x}}_2, \mathbf{u}) - \nabla \mathbf{f}_{\mathbf{x}_2} \hat{\mathbf{x}}_2] \end{bmatrix}, \tag{4}$$

$$\mathbf{Y}^a = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} & \mathbf{0} \\ \mathbf{Y}_{12}^T & \mathbf{Y}_{22} + \nabla \mathbf{f}_{\mathbf{x}_2}^T \mathbf{U}^{-1} \nabla \mathbf{f}_{\mathbf{x}_2} & -\nabla \mathbf{f}_{\mathbf{x}_2}^T \mathbf{U}^{-1} \\ \mathbf{0} & -\mathbf{U}^{-1} \nabla \mathbf{f}_{\mathbf{x}_2} & \mathbf{U}^{-1} \end{bmatrix}, \tag{5}$$

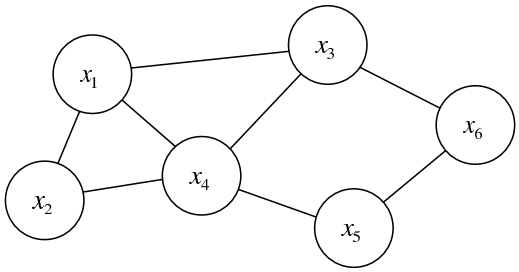
where $\nabla \mathbf{f}_{\mathbf{x}_2} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_2}$ and $\mathbf{U}$ is the uncertainty in $\mathbf{u}$. (The above equations actually represent a simplified augmentation operation, $\mathbf{x}_3 = \mathbf{f}(\mathbf{x}_2) + \mathbf{u}$. The more general form in (3) requires replacing each instance of $\mathbf{U}^{-1}$ with $(\nabla \mathbf{f}_{\mathbf{u}} \mathbf{U} \nabla \mathbf{f}_{\mathbf{u}}^T)^{-1}$.) Notice that augmentation does not alter any elements in (4) or (5) relating to state $\mathbf{x}_1$, since it does not appear in the motion model $\mathbf{f}(.)$.

Other egocentric data (e.g., GPS, IMU) are incorporated as *fusion* operations. For instance, suppose a GPS measurement at time $k = 2$ has a model of the form $\mathbf{z}_2 = \mathbf{h}(\mathbf{x}_2) + \mathbf{r}$, where $\mathbf{r}$ is zero-mean Gaussian noise with covariance $\mathbf{R}$. The information-form fusion operation therefore affects only the elements for $\mathbf{x}_2$,

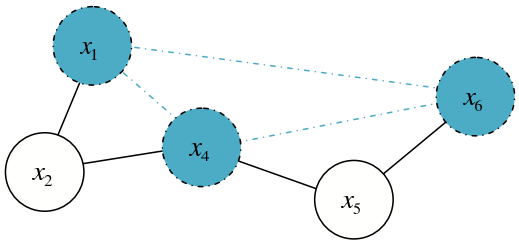
$$\hat{\mathbf{y}}^f = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 + \nabla \mathbf{h}_{\mathbf{x}_2}^T \mathbf{R}^{-1}(\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}_2) + \nabla \mathbf{h}_{\mathbf{x}_2} \hat{\mathbf{x}}_2) \\ \hat{\mathbf{y}}_3 \end{bmatrix}, \tag{6}$$

$$\mathbf{Y}^f = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} & \mathbf{0} \\ \mathbf{Y}_{12}^T & \mathbf{Y}_{22} + \nabla \mathbf{h}_{\mathbf{x}_2}^T \mathbf{R}^{-1} \nabla \mathbf{h}_{\mathbf{x}_2} & \mathbf{Y}_{23} \\ \mathbf{0} & \mathbf{Y}_{23}^T & \mathbf{Y}_{33} \end{bmatrix}. \tag{7}$$

In this way each node in the Markov chain is a pose estimate for a particular moment in time and, for the timestamp of each new egocentric measurement, a new node is added to the chain.[1] Nodes are also added for the timestamps of relevant inter-robot measurements. Thus, for platform A, there is a node generated for each measurement taken *by* A (*of* B or C), and also a node for each measurement taken *by* the other platforms *of* A.



(a) Before marginalisation



(b) After marginalisation

Fig. 4. Elimination of node x_3 alters the links and nodes of its immediate neighbours. Other parts of the graph are unaffected.

C. Marginalisation and Packet Formation

The reason for adding nodes to the Markov chain is to permit *deferred* data fusion of measurements obtained at those recorded moments. Egocentric data is fused locally soon after it is obtained (depending on time horizon t_h, see below) while inter-robot data is fused later at the central server. Therefore, nodes in the Markov chain recorded for egocentric data are obsolete *before* the chain is transmitted to the server, and can be removed by *marginalisation*.

Given the joint state vector in (1), marginalisation to eliminate $\mathbf{x}_2$ has the following information form expression,

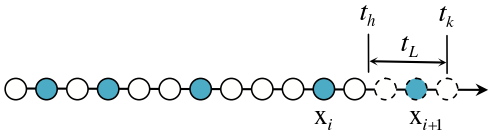
$$\hat{\mathbf{y}}_1^m = \hat{\mathbf{y}}_1 - \mathbf{Y}_{12}\mathbf{Y}_{22}^{-1}\hat{\mathbf{y}}_2, \tag{8}$$

$$\mathbf{Y}_{11}^m = \mathbf{Y}_{11} - \mathbf{Y}_{12}\mathbf{Y}_{22}^{-1}\mathbf{Y}_{12}^T. \tag{9}$$

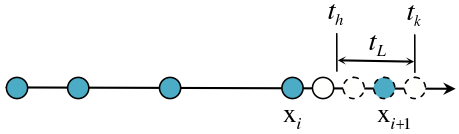
While it is not immediately apparent from these equations, marginalisation is a sparse operation. A graphical depiction of a sparse system with several states (see Fig. 4) shows that eliminating a node has local effect, creating a clique between its former neighbours.

At time k, a local platform (e.g., A) has all of its local information up to t_k, but may not have all the relevant data from its neighbouring platforms (e.g., B and C) due to communication lag. Platform A must wait to be informed of any timestamps of being observed by platforms B or C. This delays augmentation, local fusion, marginalisation, and the generation of packets to send to the server. If we presume a maximum communication delay of t_L, then we may process pose estimates for all timestamps $t_i < t_h$ $(= t_k - t_L)$, as shown in Fig. 5(a). Once fusion of egocentric data has taken place, all nodes not associated with inter-robot measurements

[1]In practice IMU and GPS data are not sychronised with odometry. We perform time-alignment, by interpolating over the odometry data, to generate pose estimates at specific timestamps. These and other implementation issues will be detailed in a forthcoming article.



(a) Lagged local fusion



(b) Marginalise

Fig. 5. Local construction of a Markov chain of pose estimates for the timestamps of egocentric (clear) and inter-robot (shaded) data. For platform A, the inter-robot timestamps include measurements by A and of A. The latter incur a delay t_L. Known timestamps that cannot yet be processed are shown dashed. Before sending states to the central server, the non-shaded states are eliminated by marginalisation. (Note, the last node before t_h must be retained to permit augmentation of future nodes.)

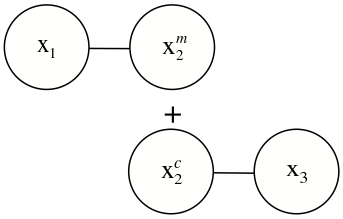


Fig. 6. A three-element Markov chain decomposed about node $\mathbf{x}_2$ into marginal and conditional components.

are eliminated by marginalisation, as shown in Fig. 5(b). At this point, the Markov chain up to $\mathbf{x}_i$ may be sent to the server.

Since estimation is performed online, and the Markov chain grows indefinitely, it is not transmitted to the server in a single block, but is sent incrementally in packets. Packets are formed by separating the Markov chain into segments according to "product rule decomposition" as follows. A Markov chain $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ can be separated into $p(\mathbf{x}_1, \mathbf{x}_2)\, p(\mathbf{x}_3|\mathbf{x}_2)$, as shown in Fig. 6. In information-form, denote the original Markov chain as

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \hat{\mathbf{y}}_3 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} & \mathbf{0} \\ \mathbf{Y}_{12}^T & \mathbf{Y}_{22} & \mathbf{Y}_{23} \\ \mathbf{0} & \mathbf{Y}_{23}^T & \mathbf{Y}_{33} \end{bmatrix}, \tag{10}$$

and the decomposed components are

$$\hat{\mathbf{y}}^m = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2^m \end{bmatrix}, \quad \mathbf{Y}^m = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{12}^T & \mathbf{Y}_{22}^m \end{bmatrix}, \tag{11}$$

$$\hat{\mathbf{y}}^c = \begin{bmatrix} \hat{\mathbf{y}}_2^c \\ \hat{\mathbf{y}}_3 \end{bmatrix}, \quad \mathbf{Y}^c = \begin{bmatrix} \mathbf{Y}_{22}^c & \mathbf{Y}_{23} \\ \mathbf{Y}_{23}^T & \mathbf{Y}_{33} \end{bmatrix}, \tag{12}$$

where the marginal part $\{\hat{\mathbf{y}}_m, \mathbf{Y}_m\}$ is obtained by marginalising away $\mathbf{x}_3$; this operation only affects $\{\hat{\mathbf{y}}_2^m, \mathbf{Y}_{22}^m\}$. The conditional part $\{\hat{\mathbf{y}}_c, \mathbf{Y}_c\}$ is obtained by subtracting the marginal from the original, such that $\hat{\mathbf{y}}_2^c = \hat{\mathbf{y}}_2 - \hat{\mathbf{y}}_2^m$ and $\mathbf{Y}_{22}^c = \mathbf{Y}_{22} - \mathbf{Y}_{22}^m$.

For the example in Fig. 5(b), suppose that the last packet sent to the server ended with node $\mathbf{x}_{i-N}$. The next packet

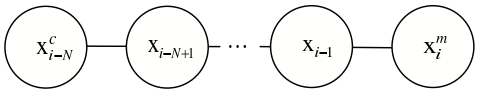
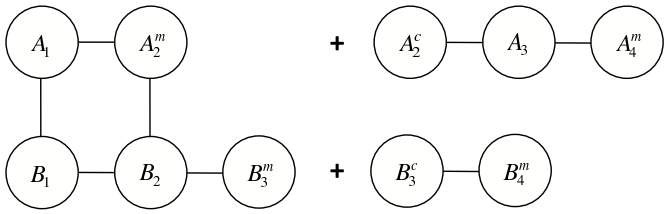
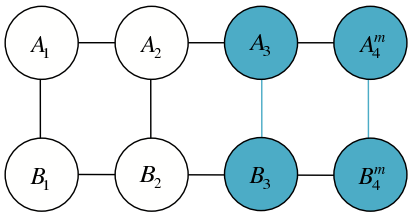


Fig. 7. A packet is composed of a sequence of pose nodes, where the first has the same timestamp as the end-state of the previous packet. The first and last nodes are altered according to the conditional and marginal components, respectively, of product rule decomposition.



(a) Connect new states



(b) Fuse inter-robot measurements

Fig. 8. Undirected graphical model of server-side state reconstruction and fusion of inter-robot measurements.

will begin with $\mathbf{x}_{i-N}$ and end with the latest node $\mathbf{x}_i$. The nodes at the beginning and end of the packet are modified to form conditional and marginal components, as shown in Fig. 7. The marginal component $\mathbf{x}_i^m$ is computed by applying product rule decomposition to the local Markov chain at $\mathbf{x}_i$. This component is also preserved in a separate local cache to be used in calculating the next packet, and to later facilitate server feedback, as discussed in Section III-E. The conditional component $\mathbf{x}_{i-N}^c$ is computed from the current value of node $\mathbf{x}_{i-N}$ by subtracting the $\mathbf{x}_{i-N}^m$ component that was cached when the previous packet was created.

Keep in mind that this set of states $\{\mathbf{x}_{i-N}^c, \ldots, \mathbf{x}_i^m\}$ is actually a sparse information-form estimate, which we shall denote $\{\hat{\mathbf{y}}^p, \mathbf{Y}^p\}$. The packet sent to the server, therefore, consists of this state segment $\{\hat{\mathbf{y}}^p, \mathbf{Y}^p\}$, the node timestamps $\{t_{i-N}, \ldots, t_i\}$ and the platform identifier, e.g., $\{\mathrm{A}\}$. (Note, these packets may be sent or forwarded to multiple servers, permitting decentralised estimation in terms of redundancy and robustness. Packet forwarding is most cheaply accomplished by a having a spanning tree over the server platforms, and propagating packets once along each path in the tree, although schemes with only local topological knowledge [3], [10] are also viable.)

D. Fusion at the Central Server

Packets from all platform accumulate at the central server, which constructs a joint state estimate and fuses inter-robot measurements. An information-form description of state reconstruction is shown in Fig. 8. New state packets connect to states in the existing joint estimate by a trivial addition of connecting nodes (e.g., $A_2 = A_2^m + A_2^c$) and extension of the state vector for the new nodes (e.g., A_3 and A_4^m). Inter-robot measurement fusion is carried out via the standard fusion equations, see (6,7); this is also a small additive operation.

In practice, we implement reconstruction and fusion using the incremental sparse Cholesky operations described in [13, Section 3.4]. The Cholesky-form shares the sparse additive properties on the information-form, but also permits efficient sparse recovery of mean and covariance components. Furthermore, the entire marginal information content of recent platform states is contained in the bottom right of the Cholesky factor, enabling obsolete past states to be eliminated by simply extracting out the relevant lower submatrix.

Sparse moment recovery is performed to compute the latest estimate for each platform, which allows platforms to improve their local linearisation operations based on server feedback. Correlation terms between platforms are also cheaply attainable via the sparse inverse [4, Section 6.7.4] to facilitate data association.

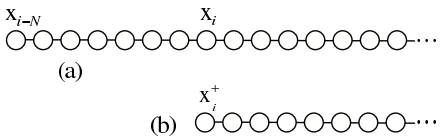


Fig. 9. Splicing server feedback into the local platform Markov chain.

E. Server Feedback

The latest server estimate for a given platform may be spliced into the local platform Markov chain as illustrated in Fig. 9. Suppose the local chain currently extends from $\mathbf{x}_{i-N}$ onwards, and a new estimate $\{\hat{\mathbf{y}}_i^s, \mathbf{Y}_{ii}^s\}$ for pose $\mathbf{x}_i$ is received from the server. The portion of the Markov chain for $\mathbf{x}_i^+$ is altered as

$$\hat{\mathbf{y}}_i^+ = \hat{\mathbf{y}}_i - \hat{\mathbf{y}}_i^m + \hat{\mathbf{y}}_i^s, \tag{13}$$

$$\mathbf{Y}_{ii}^+ = \mathbf{Y}_{ii} - \mathbf{Y}_{ii}^m + \mathbf{Y}_{ii}^s, \tag{14}$$

where $\{\hat{\mathbf{y}}_i^m, \mathbf{Y}_{ii}^m\}$ refers to the component $\mathbf{x}_i^m$ that was cached previously during packet generation. With this alteration, the Markov chain now extends from $\mathbf{x}_i$ onwards, and all older states are disconnected and discarded. The revised chain contains all recent server information for improved local model linearisation. Note, this server feedback is an exact operation; it is consistent and does not double count information.

F. Cost Analysis

The key costs of this algorithm are (i) local Markov chain construction, (ii) incremental Cholesky construction and fusion, (iii) communication bandwidth, and (iii) communication lag. For the following analysis, let n be the number of states to define a *momentary* pose, let N be the number of platforms in the team, and let d be the total number of inter-robot measurements per second.

Fig. 10. Experimental robots.

a) Markov chain construction: This cost is isolated to each platform and so is entirely decentralised. If platform A obtains k motion measurements per second, and observes or is observed by d_A inter-robot measurements per second, the per-second cost is $O((k + d_A)n^2)$.

b) Server-side reconstruction and moment recovery: The cost of building the sparse Cholesky joint state, and fusing inter-robot measurements, is not immediately dependent on N. Rather it is a function of d, which typically increases with N. This cost depends on the frequency and detection pattern of inter-robot measurements, and requires non-trivial analysis. Example-based costs will be detailed in a forthcoming article.

c) Communication bandwidth: Each inter-robot measurement is sent to the server, and is followed by two state estimates for the two robots involved. Thus, the per second cost is $O(dn^2)$.

d) Communication Lag: The first lag is t_L, the maximum delay for one platform to inform another of its timestamp. The next is t_p, the time to compute a state packet and send it to the server. The last is t_s, the time to compute the server estimate and send it back to the platform. Thus, the total time for a platform to obtain an optimal estimate is $t_L + t_p + t_s$.

IV. Experimental System

An experimental evaluation of the proposed algorithm was carried out on a team of three Pioneer indoor robots (see Fig. 10). Each robot is equipped with wheel encoders for odometry and SICK scanning lasers for range-bearing measurements. One robot also has an ISIS inertial measurement unit. However, for the purposes of emulating a heterogeneous system, the robot sensing was processed and limited as follows.

1) Platform A uses its laser to perform scan-matching-based localisation from an a priori map. It also uses this laser to make range-bearing measurements of the other robots.
2) Platform B uses its laser solely to make range-bearing measurements of the other robots.
3) Platform C does not use its laser. It uses the yaw gyroscope of its IMU to measure heading.

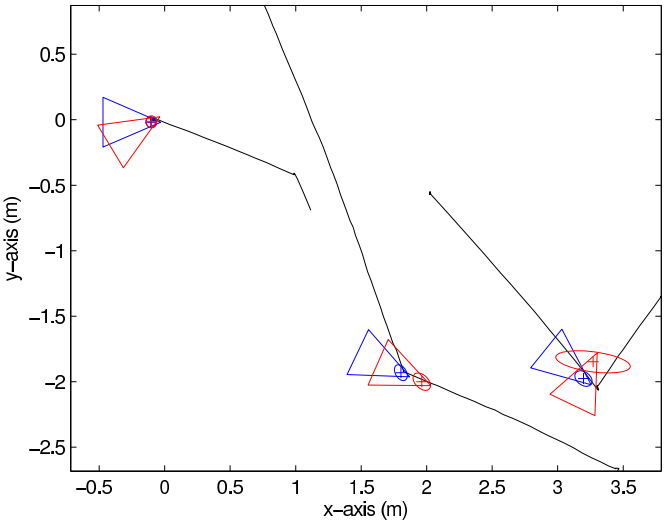


Fig. 11. Most recent pose estimates at server (blue) and on each platform (red) for platforms A, C and B (left-to-right), shown with 2σ uncertainty ellipses. The ground-truth vehicle paths for a short period before and after these poses is shown in black.

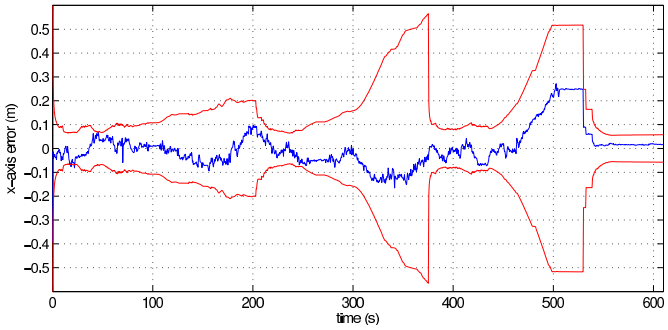
All robots use their odometry for motion measurement. Platforms A and B perform feature extraction on their laser data to generate range-bearing observations of other robots. Data association is managed by auxiliary means and is assumed known. Platform C is effectively blind, but has a higher quality motion model due to the gyroscope. While ground-truth is not available, it is approximated by recording scan-matching-based poses of platforms B and C, which provides sub-centimetre accuracy.

The aim of this experiment is to demonstrate the transfer of pose information from a well-localised robot to one that has no localisation capability at all. We do not discuss tuning parameters, consistency or estimate quality since our filter is optimal, and the resultant estimate is the same as for a centralised EKF given the same information.

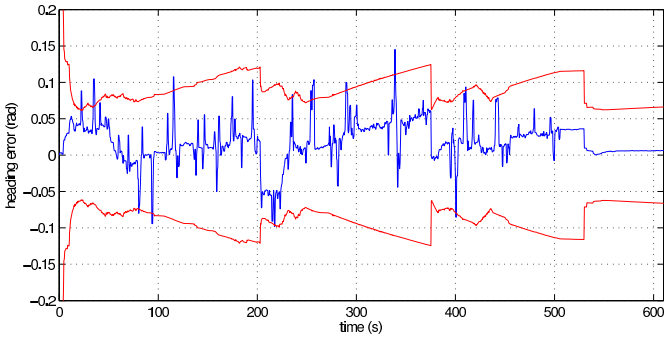
A. Results

The estimate held by the server incorporates all available information received so far, both egocentric and inter-robot data for all platforms. However, this information is lagged, and each platform can generate a more recent pose estimate for itself, albeit without integration of the latest inter-robot measurements. An example of lagged optimal estimates and current local estimates, for a given moment in time, is shown in Fig. 11.

The estimate errors for platform C are shown in Fig. 12. This robot, and also platform B, have no ability to localise from the map, and would exhibit unbounded error growth if travelling in isolation. However, the range-bearing observations made by platforms A and B transfer the pose information of platform A directly or indirectly to platform C, so that it exhibits bounded pose uncertainty.



(a) Estimate error in x-axis (m)



(b) Estimate error in heading (rad)

Fig. 12. The estimate error of platform C is bounded with time, as shown by the 2-σ uncertainty bounds. This demonstrates a transfer of pose information to platform C from platform A.

V. Conclusion

This paper presents an efficient algorithm for cooperative localisation, wherein fusion of egocentric data is decentralised and fusion of inter-robot measurements is performed on a central server. The server is easily decentralised by duplication if robustness is required. Since motion data tends to be very high bandwidth, this approach minimises communications costs. The resultant pose estimates are optimal, in the same sense as an EKF, and it is possible to splice server estimates back into the local platform Markov chains in a consistent manner. The ability of the algorithm to share pose information in a robot team, and in particular to transfer localisation capabilities to a platform that does not possess adequate sensing, is demonstrated by a simple experiment with indoor mobile robots.

In a forthcoming article we will detail the many subtle issues involved in implementing this algorithm on a real system. We will also provide a cost analysis for the server-side Cholesky factor operations—state reconstruction, inter-robot data fusion and moment recovery—for robot teams of various sizes and with various patterns of inter-robot observations.

In future work we plan to investigate efficient mechanisms for revising linearisations of process and observation models, especially estimates of past motion steps. These revisions can be significant if the state estimate includes static or slowly varying parameters, such as IMU bias estimates, that are sensitive to long-term linear approximations.

Acknowledgements

This research was undertaken through the Centre for Intelligent Mobile Systems (CIMS), and was funded by BAE Systems as part of an ongoing partnership with the University of Sydney. It was also supported by the ARC Centres of Excellence Programme funded by the Australian Research Council and the New South Wales State Government.

References

[1] A. Bahr, M.R. Walter, and J.J. Leonard. Consistent cooperative localisation. In *IEEE International Conference on Robotics and Automation*, 2009.

[2] T. Bailey and H. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine*, 13(3):108–117, 2006.

[3] T. Bailey and H. Durrant-Whyte. Decentralised data fusion with delayed states for consistent inference in mobile ad hoc networks. Technical report, University of Sydney, Australian Centre for Field Robotics, 2007.

[4] A. Bjorck. *Numerical Methods for Least Squares Problems*. Siam, 1996.

[5] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006.

[6] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters. In *IEEE International Conference on Robotics and Automation*, pages 2417–2424, 2005.

[7] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, 2000.

[8] A. Howard, M. Matari, and C. Sukhatme. Localization for mobile robot teams: A distributed MLE approach. In B. Siciliano and P. Dario, editors, *Experimental Robotics VIII*, pages 146–155. Springer-Verlag, 2003.

[9] R. Kurazame and S. Nagata. Cooperative positioning with multiple robots. In *IEEE International Conference on Robotics and Automation*, 1994.

[10] K.Y.K. Leung, T.D. Barfoot, and H.H.T. Liu. Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach. *IEEE Transactions on Robotics*, 26(1):62–77, 2010.

[11] R. Madhavan, K. Fregene, and L.E. Parker. Distributed heterogeneous outdoor multi-robot localization. In *IEEE International Conference on Robotics and Automation*, 2002.

[12] A. Makarenko, A. Brooks, T. Kaupp, H. Durrant-Whyte, and F. Dellaert. Decentralised data fusion: A graphical model approach. In *IEEE International Conference on Information Fusion*, 2009.

[13] H. Mu, T. Bailey, P. Thompson, and H. Durrant-Whyte. Decentralised solutions to the cooperative multi-platform navigation problem. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2), 2011.

[14] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *IEEE International Conference on Robotics and Automation*, 2009.

[15] M. Paskin, C. Guestrin, and J. McFadden. A robust architecture for distributed inference in sensor networks. In *International Symposium on Information Processing in Sensor Networks*, 2005.

[16] M. Paskin and G.D. Lawrence. Junction tree algorithms for solving sparse linear system. Technical report, University of California, Computer Science Division, 2003.

[17] S.I. Roumeliotis and G.A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.