Decentralised Data Fusion with Delayed States for Consistent Inference in Mobile Ad Hoc Networks

Tim Bailey and Hugh Durrant-Whyte

Australian Centre for Field Robotics, University of Sydney
{tbailey,hugh}@acfr.usyd.edu.au

Abstract. Delayed-state decentralised data fusion (DS-DDF) is proposed as a general methodology for consistent DDF, which does not impose constraints on network topology. The resulting estimates, although lagged in time, are optimal, equal to a centralised solution. The method is demonstrated in the context of dynamic node tracking and localisation, where a team of mobile robots track each other's position to obtain a joint estimate of the position of every team member.

1 Introduction

Consider a scenario where a team of mobile robots travel throughout an environment, communicating with each other over an ad hoc network. All are equipped with odometry and a ranging sensor capable of measuring the range to other robots. Some have GPS receivers, others do not. The task then is to enable each robot to estimate its own location and that of every other robot in the network.

A centralised solution is trivial. Each robot simply forwards all of its measurement data to a central server, which performs conventional data fusion to obtain a joint estimate over the poses of all robots. Individual robots can then request estimates from the server. However, the disadvantages of a centralised architecture include (i) high bandwidth requirements, especially for transmission of high-frequency motion data, (ii) limited range since each robot must be within communication range of the server (or, alternatively be able to reroute or forward data via intermediate nodes), and (iii) robustness issues because server failure means the whole system fails.

Decentralised data fusion (DDF) overcomes the weaknesses of centralised fusion and provides scalable, modular fusion capabilities [1,2]. Localised fusion distributes the computational load, peer-to-peer communications minimises bandwidth, and multiplicity of nodes provides redundancy and robustness to node failure.

This paper introduces to DDF the concept of delayed states, where past estimates are retained in the joint state vector to capture historical dependencies or correlations, and to permit fusion of data involving multiple nodes, such as range measurements. A motivating example of delayed-state DDF (DS-DDF) is provided by a multi-robot tracking and localisation scenario in which a team of robots track each others position to provide localisation information for the network as a whole. However, we stress that the concept of DS-DDF is general and not limited to a particular structure.

The format of the paper is as follows. The next section discusses previous work that influenced the development of DS-DDF. Section 3 presents the essential canonical-form operations for estimation and their properties. Section 4 applies DS-DDF to the problem of tracking a dynamic network of mobile robots, and the following section provides a simulation experiment of this scenario. Section 6 discusses improvements to the dynamic network implementation and further application domains for DS-DDF. The final section concludes with a summary of the key points.

2 Related Work

The primary influence on this current work is from recent research in simultaneous localisation and mapping (SLAM). This literature provides two key concepts. First is the idea of delayed (or deferred) states for managing historical dependencies. Delayed states are used for landmark initialisation [3–5] in situations when data must be accumulated over a period of time before the landmark becomes fully observable. The second contributing factor is the sparse canonical-form[1] representation of the SLAM problem [6–8], which demonstrates that efficient operations are possible when representing a Gaussian distribution in canonical form. In particular, augmenting the state and applying fusion updates can be very cheap (i.e., constant-time operations) if they operate over a small portion of the state vector.

Another significant influence is the more conventional form of estimation-theoretic DDF [1,2], which also uses the canonical-form Gaussian representation, primarily because it permits efficient additive fusion. Conventional DDF is optimal but restricted to networks that are either fully connected or possess non-cyclical topologies. In the latter case, channel filters are used to keep track of common information between nodes. When applied to decentralised target-tracking problems, this form of DDF also suffers from double counting of com-

[1] Canonical-form is also known as information-form.

mon prediction information, which previously has either been ignored or mitigated by covariance inflation heuristics.[2]

DS-DDF departs from conventional DDF by maintaining a history of past states rather than just the most recent momentary state. These past states are able to capture the contributions of individual nodes, providing an intrinsic record of common information, and so avoids the need for channel filters and restricted topology.

3 Essential Properties of the Canonical-Form Gaussian for DS-DDF

The properties of DS-DDF arise directly from the properties of the canonical-form Gaussian representation and, particularly, the form of the three essential operations for estimation: augmentation, marginalisation and fusion. These operations permit construction of a sparse joint information matrix. Furthermore, the augmentation and fusion operations are additive and, under mild conditions, their contributions may be separated into constituent parts.

Sparseness of the information matrix and the ability to separate and reconstruct components of the joint estimate are the fundamental properties of the DS-DDF paradigm.

3.1 Augmentation, Marginalisation and Fusion

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 canonical-form representation of a Gaussian probability distribution $\mathcal{N}(\mathbf{x}; \hat{\mathbf{y}}, \mathbf{Y})$ over these states is given by an information vector,

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \end{bmatrix}, \tag{2}$$

[2] This problem, although mentioned in [2, page 6], is different from the rather simpler "common process model" problem described in [2, Section 2.4.4]. It arises also in centralised track-to-track fusion [9]. As discussed later in this paper, the problem is due to premature marginalisation of past estimates, and can be resolved with delayed states, which keeps the process information in a separable form.

and an information matrix,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{12}^T & \mathbf{Y}_{22} \end{bmatrix}. \tag{3}$$

These canonical-form terms are related to the more conventional moment-form Gaussian representation according to $\mathbf{Y} \triangleq \mathbf{P}^{-1}$ and $\hat{\mathbf{y}} \triangleq \mathbf{Y}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is the mean vector and $\mathbf{P}$ is the covariance matrix.

To extend or *augment* the state vector as a function of sub-states $\mathbf{x}_2$ and some independent random variable $\mathbf{u}$,

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 = \mathbf{f}\left(\mathbf{x}_2, \mathbf{u}\right) \end{bmatrix}, \tag{4}$$

the canonical-form estimate is augmented as,

$$\hat{\mathbf{y}}_a = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 - \nabla\mathbf{f}_{\mathbf{x}_2}^T\mathbf{U}^{-1}[\mathbf{f}(\hat{\mathbf{x}}_2, \mathbf{u}) - \nabla\mathbf{f}_{\mathbf{x}_2}\hat{\mathbf{x}}_2] \\ \mathbf{U}^{-1}[\mathbf{f}(\hat{\mathbf{x}}_2, \mathbf{u}) - \nabla\mathbf{f}_{\mathbf{x}_2}\hat{\mathbf{x}}_2] \end{bmatrix}, \tag{5}$$

$$\mathbf{Y}_a = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} & \mathbf{0} \\ \mathbf{Y}_{12}^T & \mathbf{Y}_{22} + \nabla\mathbf{f}_{\mathbf{x}_2}^T\mathbf{U}^{-1}\nabla\mathbf{f}_{\mathbf{x}_2} & -\nabla\mathbf{f}_{\mathbf{x}_2}^T\mathbf{U}^{-1} \\ \mathbf{0} & -\mathbf{U}^{-1}\nabla\mathbf{f}_{\mathbf{x}_2} & \mathbf{U}^{-1} \end{bmatrix}, \tag{6}$$

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 (4) requires a trivial, though cluttered, extension replacing each instance of $\mathbf{U}^{-1}$ with $(\nabla\mathbf{f}_{\mathbf{u}}\mathbf{U}\nabla\mathbf{f}_{\mathbf{u}}^T)^{-1}$.

The sub-states $\mathbf{x}_2$ can be removed from (1) by *marginalisation* using the following canonical-form expressions,

$$\hat{\mathbf{y}}_{1_m} = \hat{\mathbf{y}}_1 - \mathbf{Y}_{12}\mathbf{Y}_{22}^{-1}\hat{\mathbf{y}}_2, \tag{7}$$

$$\mathbf{Y}_{11_m} = \mathbf{Y}_{11} - \mathbf{Y}_{12}\mathbf{Y}_{22}^{-1}\mathbf{Y}_{12}^T. \tag{8}$$

Data *fusion* occurs when an observation $\mathbf{z}$ is made according to some function of the state. If the observation model refers only to the sub-states $\mathbf{x}_2$ in (1),

$$\mathbf{z} = \mathbf{h}\left(\mathbf{x}_2\right) + \mathbf{r}, \tag{9}$$

where $\mathbf{r}$ is zero-mean Gaussian noise with covariance $\mathbf{R}$, then observations are fused according to,

$$\hat{\mathbf{y}}^+ = \begin{bmatrix} \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 + \nabla\mathbf{h}_{\mathbf{x}_2}^T\mathbf{R}^{-1}\mathbf{z} \end{bmatrix}, \tag{10}$$

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

3.2 Properties of Estimation Operations

Equations (4–11) make apparent several key properties of the canonical form. For this discussion, let the sub-states $\mathbf{x}_1$ comprise the bulk of the state-vector, such that $\dim(\mathbf{x}_2) \ll \dim(\mathbf{x}_1)$.

Augmentation (5, 6) is additive and sparse. The bulk of the extension to the information matrix is zeros. The change to $\mathbf{Y}_{22}$ is additive and the quantity of this additive term is recorded in the off-diagonal term $-\nabla\mathbf{f}_{\mathbf{x}_2}^T\mathbf{U}^{-1}$, permitting later separation of factors.

Marginalisation (7, 8) is not additive and removing any element will change the value of all other elements linked to it.[3] Marginalisation also introduces a clique of new links between any elements originally linked to the removed element. States that were not linked to the removed element are not affected.

Fusion (10, 11) is additive and affects only those states directly involved in the observation model.

Augmentation and fusion, being additive and localised in their effect, permit decomposition of the joint estimate into constituent parts. This can be achieved with minimal auxiliary bookkeeping, since much about the individual contributions is recorded implicitly in the off-diagonals of the information matrix.

Consider, for instance, a set of fusion operations involving three nodes A, B, and C. At time k, node A observes the range to B and C, and node B observes the range to C. The observation model for a measurement from A to B is given by,

$$z = \mathbf{h}\left(\mathbf{x}_a, \mathbf{x}_b\right) + r = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} + r, \tag{12}$$

with r being zero-mean noise with variance R. Its Jacobian is

$$\nabla\mathbf{h}_{\mathbf{x}_{ab}} = [\mathbf{H}_{ab}, -\mathbf{H}_{ab}], \tag{13}$$

where

$$\mathbf{H}_{ab} = \left.\frac{\partial\mathbf{h}}{\partial\mathbf{x}_a}\right|_{(\hat{\mathbf{x}}_a,\hat{\mathbf{x}}_b)} = \left[\frac{(\hat{x}_b - \hat{x}_a)}{\mathbf{h}\left(\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_b\right)}, \frac{(\hat{y}_b - \hat{y}_a)}{\mathbf{h}\left(\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_b\right)}, 0, 0\right], \tag{14}$$

[3] A very useful property of the canonical-form Gaussian representation is the exact analogy between the information matrix and an undirected graphical model. A diagonal (or block-diagonal) value in the matrix corresponds to a vertex in the graph and a non-zero off-diagonal value corresponds to a link between two vertices. Thus, we use the term “link” to refer to non-zero off-diagonal values.

assuming here that each node's momentary state is composed of position and velocity, $\mathbf{x}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T$. After the three observations, the joint information matrix for states $[\mathbf{x}_a^T, \mathbf{x}_b^T, \mathbf{x}_c^T]^T$ is

$$\begin{bmatrix} \mathbf{I}_{ab} + \mathbf{I}_{ac} & -\mathbf{I}_{ab} & -\mathbf{I}_{ac} \\ -\mathbf{I}_{ab} & \mathbf{I}_{ab} + \mathbf{I}_{bc} & -\mathbf{I}_{bc} \\ -\mathbf{I}_{ac} & -\mathbf{I}_{bc} & \mathbf{I}_{ac} + \mathbf{I}_{bc} \end{bmatrix}, \tag{15}$$

where $\mathbf{I}_{ab} = \mathbf{H}_{ab}^T R^{-1} \mathbf{H}_{ab}$. Notice that the diagonal elements are always positive and the off-diagonal elements record the individual contributions.

The separability of augmentation and fusion operations permits considerable flexibility in manipulating the joint state estimate; the information matrix may be sliced into sub-matrices, operated on in parts, and reconstituted into a joint representation. Marginalisation, on the other hand, is not additive or separable and must be applied with care. Recent SLAM research [6–8] shows that marginalisation tends to increase the density of the information matrix and should be avoided for computational reasons. For DDF, separability is a greater issue; the values for all states that were linked to an element removed by marginalisation become correlated and mixed together. Fortunately, for DDF, the augmented joint estimate captures all available information and marginalisation is never necessary, except for computational tractability, and can be applied judiciously to states whose information is truly obsolete.

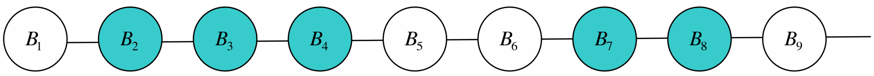
4 Tracking and Localisation in a Dynamic Network

The scenario described at the beginning of this paper is an instance of a dynamic network, where moving nodes track the state of other nodes in the network. The objective then is for all robots to localise the entire team in a decentralised manner. This looks like a SLAM problem in which the other robots are both landmarks, which are in motion, and localisers in their own right. The complexity of this scenario arises because of the range measurements. Without range measurements each robot's state would be independent of the rest; but, of course, this also means those robots without GPS would have no notion of their location. The introduction of relative measurements, each involving a pair of robots, correlates the entire system.

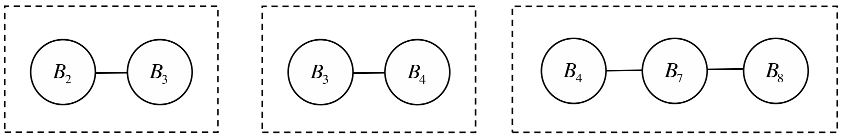
This example is a very simple illustration of DS-DDF insofar as it involves only basic decomposition of the joint state estimate, and does not exploit fully the separation possibilities available. However, implementing even this simple example involves significant bookkeeping to keep track of node identities and timestamps. There are also various subtleties, particularly regarding marginalisation and its effect on adjacent states. Due to limited space, these important but incidental details are omitted, so as to focus on the essential concepts.

4.1 Own-State Estimation

Each node has three sources of information: odometry, GPS and range measurements to other nodes. The range data is not used directly but is instead communicated to all nodes in the network as it is obtained. Odometry and GPS, on the other hand, are fused locally. Odometric data $\mathbf{u}_k$ is used to augment the node's estimate of itself according to its motion model, $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$. The augmented "own-state" estimate forms a Markov chain, as shown in Fig. 1(a), where each momentary state is linked only to the states immediately before and after it. A GPS measurement, if received at time k, updates the estimate of the momentary state for time k.



(a) Own-state estimate



(b) Own-state packets

Fig. 1. Node B estimates its own state, augmenting the state vector with each motion measurement and fusing GPS, if available. It marks those states with associated range measurements (a), removes irrelevant states, and generates state packets (b).

As a node receives packets of range data from all nodes, including itself, it notes the subset of ranges involving itself. These are its own range measurements observing other nodes, and measurements from other nodes observing it. The timestamps of these measurements define which of its own-states are relevant for joint fusion. Since motion data (odometry) is typically much higher frequency than range measurements, the great majority of own-states are irrelevant, and these are marginalised away. The remaining states are formed into own-state packets, as shown in Fig. 1(b).[4] Once packetised, an old own-state has no influence on future own-states and the node can simply "forget" past states by removing them from its Markov chain.

Each packet must include at least two momentary states. The first momentary state in each packet has the same timestamp as the last momentary state in the preceding packet.[5] Thus, a sequence of own-state packets from a particular node permits recovery of the complete own-state Markov chain by connecting the common "stub-states" in each packet; the beginning stub-state of the later packet overwrites the last stub-state of the previous packet.

4.2 Joint Estimation

A node accumulates own-state packets from all nodes, including itself, and forms them into a joint estimate by chaining stub-states as shown in Fig. 2(a). Range data is then fused, which links the two momentary states from the associated nodes and timestamps, as between A_3 and B_3 in Fig. 2(b). The result is a reconstructed joint state as if from conventional centralised data fusion. (Note, range data fusion cannot involve stub-states as this would prevent connection to the next own-state packet. Range fusion is deferred until both states are not stubs.) Older joint states that are not stub-states and not directly linked to a stub-state are obsolete and can be removed by marginalisation as shown in Fig. 2(c).

4.3 Communication Strategies

The essential property for communications is that each packet of range data and own-state data reaches every node in the network. Nodes need not communicate directly with every other node, since

[4] Marginalisation and packetisation can only occur up to a time-horizon—the timestamp for when the node knows it has been notified of *all* range measurements referring to itself.

[5] The *value* of the beginning stub-state estimate is typically not the same as the last stub-state of the preceding packet. During joint estimation, a moment-form estimate (i.e., mean and covariance) might be required for linearisation or visualisation purposes. To recover this from the canonical-form, the last stub-state must be made a "terminal" stub-state—a state that is not tied to a later state—but, when a sequence of own-states is extracted from the middle of the Markov chain, the last state includes prediction information linking it to the next state. The last stub-state is therefore modified to remove this information.

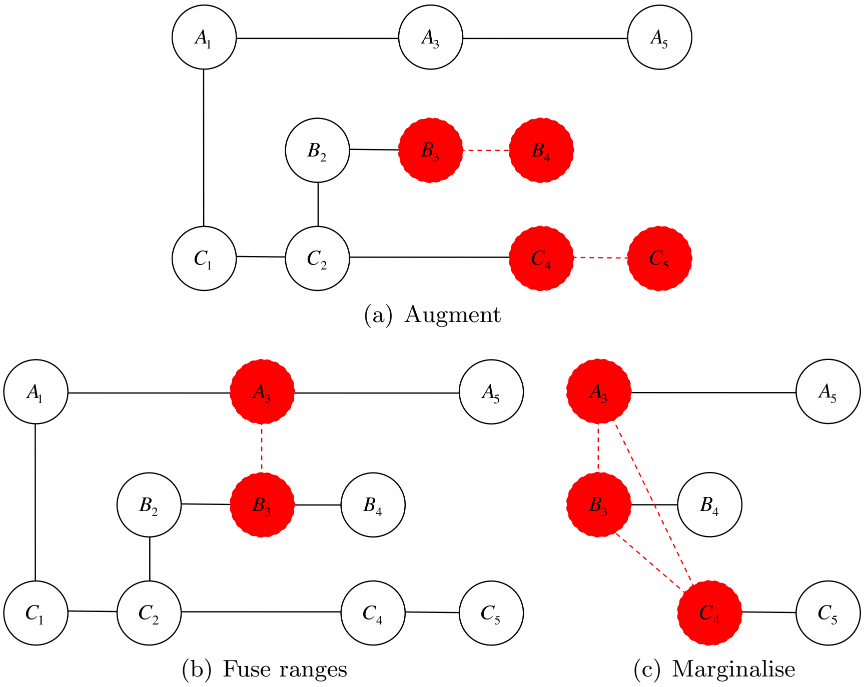


Fig. 2. Joint fusion augments the state with new state packets (a), connecting stub-states. It then fuses any range data not involving stub-states (b), and removes older states (c). The parts of the state affected by each operation are shown in red.

packets may be forwarded via intermediaries. Various communications strategies are possible, including packet tagging and request-reply handshaking between peers. The method implemented for this paper was a simple bookkeeping strategy, which tracks the last sender for each packet and records the latest direct communications for each node. The result is redundant packet forwarding—each node receives every packet, but some packets are received multiple times—which exhibits minimal redundancy if each node keeps to a fixed neighbourhood of direct links.

5 Experimental Results

DS-DDF for dynamic node tracking and localisation is demonstrated by a simulation experiment. Ten mobile robots, each moving according to a random walk behaviour, communicate with each other and make range observations. Only node 1 has GPS. For simplicity, we assume the system is synchronous,[6] operating in 0.1 second timesteps. The implementation does not attempt to address data association, data validation or large non-linearities. To mitigate problems with the non-linear range measurement model, all nodes were initialised with a reasonable position estimate.

The experimental parameters are as follows. The standard deviations for measurements are: initial node positions, 5 metres; velocity measurements (from odometry), 1 metre/sec; GPS measurements, 5 metres; range measurements, 2 metres. At each timestep, the probability of a node obtaining a measurement are: GPS, 10% (node 1 only); range measurement to another node, 5%; communication with another node, 10%. Note, the probabilities for range measurements and communication are for *each* node interaction; so each timestep, node X has 10% probability of measuring node 1, and 10% probability of measuring node 2, etc.

The above parameters are non-critical since the lagged DS-DDF estimate is identical to an optimal centralised estimate. (There are minor variations due to differences in linearisation points for range measurements.) Only the communication behaviour is important as it affects the propagation characteristics of data packets and so determines the system lag.

Each node exhibits different lags in its estimate of itself and other nodes. Fig. 3 shows the estimate that node 1 has of node 3 in terms of its x-axis position. At each moment t, node 1 holds an optimal estimate of node 3 referring to a past moment in time, $t - t_{lag}$. The time lag t_{lag} is shown in Fig. 4. To obtain a current-time estimate, node 1 applies a predictive model (e.g., a constant-velocity inertial model) to the most recent optimal estimate. This forward-predicted estimate is conservative and is generated by an auxiliary estimator, playing no part in the lagged optimal estimator.

6 Discussion and Future Directions

This dynamic network tracking and localisation example shows a rather primitive and inefficient form of DS-DDF. Its main advantage is local fusion of high frequency motion information, avoiding the bandwidth congestion of communicating this data in raw form.

[6] It is straightforward to implement an asynchronous system in practice by predictive time-alignment, but tedious to simulate. There is no loss of generality in terms of DS-DDF.

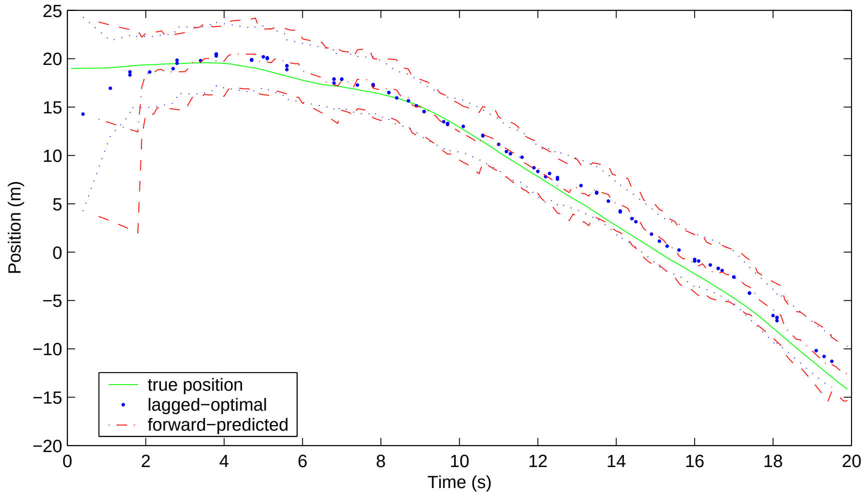


Fig. 3. The estimate, and 2-σ bounds, that node 1 has for the x-axis position of node 3. The optimal estimates are shown here at their correct timestamps, but are not known by node 1 until after the lag shown in Fig. 4. A conservative current-time estimate can be predicted from the latest optimal estimate.

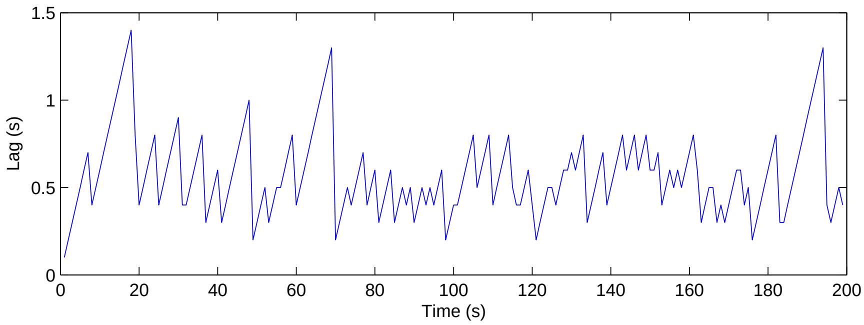


Fig. 4. Time lag of the optimal estimate. At each moment t, the optimal estimate refers to the node's position from $t - t_{lag}$ seconds in the past.

It also enables consistent fusion of range data, which is dependent on node pairs. However there is increased communications lag, as each node must wait for range timestamps from neighbouring nodes before performing marginalisation and transmission of "own-states". There also remains significant bandwidth and computational costs, since each range measurement, and its associated pair of own-state estimates, is transmitted to every node in the network, and each node duplicates the joint fusion operation.

This rather excessive redundancy may be greatly reduced by assuming a client-server structure. A small subset of nodes are appointed servers and the rest are clients. All nodes send their range measurements to the nearest server, and this server sends back the relevant range timestamps so that a node may perform own-state marginalisation and send own-state packets to the server. Servers forward their information to all other servers, and so perform DS-DDF in the fashion described above. Clients can request marginal estimates from any server when desired, but do not participate in joint fusion. Thus, the redundancy is reduced to the number of servers rather than the total number of nodes.

We expect that still more efficient strategies are possible. The separability properties of the canonical-form estimate would permit portions of the joint state to be fused on one node and transmitted in parts to other nodes. The amount of redundancy would then be fully configurable. A more general strategy for manipulating the joint state is the subject of current research.

DS-DDF is not limited to dynamic node tracking and appears to be a general paradigm for decentralised estimation. Further applications include (i) static nodes tracking moving targets,[7] (ii) moving nodes tracking static targets (such as DDF-SLAM [10]), and (iii) combinations of moving and static nodes and targets.

7 Conclusion

This paper introduces delayed states to decentralised data fusion. Delayed states capture statistical dependence with past estimates and between multiple nodes. The ability to retain historical correlations and the separability properties of the canonical-form Gaussian representation open up exciting new possibilities for Bayesian DDF.

Acknowledgments

This work is funded by L-3 Communications. The authors would like to thank Ben Upcroft, Paul Thompson and Lee-Ling Ong from the ACFR, Chris Lloyd from BAE Systems, and Steve Rounds and Joe Nicosia from L-3 Communications for fruitful discussions related to delayed states and DDF.

[7] The difference between a node and a target is that the former obtains measurements (i.e., motion, position, etc.) about itself, while the latter is passive.

References

1. S.H. Grime, *Communication in Decentralised Sensing Architectures*, Ph.D. thesis, University of Oxford, Department of Engineering Science, 1992.
2. E. Nettleton, *Decentralised Architectures for Tracking and Navigation with Multiple Flight Vehicles*, Ph.D. thesis, University of Sydney, Australian Centre for Field Robotics, 2003.
3. M. Deans and M. Hebert, "Experimental comparison of techniques for localization and mapping using a bearing-only sensor," in *International Symposium on Experimental Robotics*, 2000.
4. J.J. Leonard and R.J. Rikoski, "Incorporation of delayed decision making into stochastic mapping," in *International Symposium On Experimental Robotics*, 2000.
5. J.J. Leonard, R.J. Rikoski, P.M. Newman, and M.C. Bosse, "Mapping partially observable features from multiple uncertain vantage points," *International Journal of Robotics Research*, vol. 21, no. 10–11, pp. 943–975, 2002.
6. R.M. Eustice, H. Singh, and J.J. Leonard, "Exactly sparse delayed-state filters," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 2417–2424.
7. F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.
8. T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (SLAM): Part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
9. K. Chang, R. Saha, and Y. Bar-Shalom, "On optimal track-to-track fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271–1276, 1997.
10. P. Thompson, *Trajectory State Estimation, Graphical Formulation and Local and Decentralised Estimation Algorithms*, Ph.D. thesis, University of Sydney, Australian Centre for Field Robotics, (forthcoming) 2008.