Robotics and Autonomous Systems 57 (2009) 310–320

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Decentralised decision making in heterogeneous teams using anonymous optimisation

George M. Mathews [a,b,*], Hugh Durrant-Whyte [a], Mikhail Prokopenko [c]

[a] ARC Centre of Excellence for Autonomous Systems, The University of Sydney, Sydney NSW, Australia
[b] CSIRO Industrial Physics, Lindfield NSW, Australia
[c] CSIRO ICT Centre, North Ryde NSW, Australia

ARTICLE INFO

Article history:
Available online 19 November 2008

Keywords:
Asynchronous optimisation
Team decision making
Multi-agent systems

ABSTRACT

This paper considers the scenario where multiple autonomous agents must cooperate in making decisions to minimise a continuous and differentiable team cost function. A distributed and asynchronous optimisation algorithm is presented which allows each agent to incrementally refine their decisions while intermittently communicating with the rest of the team. A convergence analysis provides quantitative requirements on the frequency agents must communicate that is prescribed by the structure of the decision problem. In general the solution method will require every agent to communicate to and have a model of every other agent in the team. To overcome this, a specific subset of systems, called Partially Separable, is defined. These systems only require each agent to have a combined summary of the rest of the team and allows each agent to communicate locally over an acyclic communication network, greatly increasing the scalability of the system.

1. Introduction

This paper is part of an on going research program into decision making and control algorithms for large distributed active sensor networks. The vision of this work is a system of interconnected robotic agents, such as a team of unmanned air and ground vehicles. This type of system has applications in environmental monitoring, searching and rescue, surveillance, bush firefighting, target tracking, mapping and exploration, etc. Utilising an autonomous system to perform these tasks allows human operators to be removed from possibly dangerous (or simply mundane) situations.

The decision problems that are encountered in these types of multi-agent systems generally fall into two categories, discrete (e.g. task allocation) and continuous (e.g. motion control). This work is primarily focused on the latter. Current work in this area has mainly focused on the consensus problem [1–4], whereby multiple agents, each starting out with different local values of a common decision variable, must collaboratively reach agreement.

However, what is often overlooked in consensus theory is that some decisions are explicitly better than others (there exists a common cost function). To include this it is necessary to cast the decision problem as a distributed optimisation problem.

Existing approaches to this optimisation problem are either: (1) Fully centralised [5,6] where the system as a whole is modelled and the optimal decision found using conventional algorithms. (2) Distributed or hierarchical [7] which utilise the distributed computational capacity of the multiple platforms but require a single facility to fuse information or resolve global constrains, or (3) Fully decentralised which do not require any centralised facility.

The decentralised approach can be further broken down into systems that require each platform to have global information about the system and those that only require local information from a small subset [8,9]. It is the latter that is of interest here.

This paper presents a general asynchronous gradient-based distributed optimisation algorithm that can be used to solve the team decision problem. However, this algorithm requires every agent to communicate to every other agent, limiting the scalability of the system. To overcome this issue a specific class of systems, called *Partially Separable*, is defined that enable each agent to communicate in an anonymous fashion over an acyclic communication network.

The paper is organised as follows: Section 2 presents the general decentralised and asynchronous optimisation algorithm for smooth and differentiable cost functions. A convergence analysis quantitatively specifies the communication requirements

* Corresponding address: Advanced Technology Centre, BAE SYSTEMS, Sowerby Building, FPC 267, PO BOX 5, Filton, Bristol, United Kingdom. Tel.: +44 0 117 302 8149; fax: +44 0 117 302 8007.

E-mail addresses: george.mathews@baesystems.com (G.M. Mathews), h.durrant-whyte@cas.edu.au (H. Durrant-Whyte), mikhail.prokopenko@csiro.au (M. Prokopenko).

0921-8890/$ – see front matter
doi:10.1016/j.robot.2008.10.020

of each agent. The scalability implications of these requirements are discussed in Section 3 and the Partial Separability condition defined. Section 4 presents a decentralised system architecture for partially separable systems and gives numerical results for a simulated feature localisation task. Section 5 present conclusions and future work.

2. Decentralised optimisation

This section introduces the team decision problem and presents a general asynchronous decentralised optimisation algorithm based on the work of Bertsekas and Tsitsiklis [10–12].

2.1. Problem definition

Consider a team of p agents, where each agent i is in charge of a local decision or control variable $u_i \in \mathcal{U}_i$. For simplicity, this work assumes the decision spaces are scalar and thus $\mathcal{U}_i = \Re$. For an extension to multi-dimensional decisions see [13].

The agents are required to select their decisions such that a given team cost function $J : \mathcal{U}_1 \times \cdots \times \mathcal{U}_p \to \Re$ is minimised

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathcal{U}} J(\mathbf{u}) \quad (1)$$

where

$$\mathbf{u} = [u_1, u_2, \ldots, u_p]^{\mathrm{T}} \in \mathcal{U}_1 \times \cdots \times \mathcal{U}_p \triangleq \mathcal{U} \quad (2)$$

is the team decision vector and $\mathbf{u}^*$ is the desired optimal team decision.

Assumption 1a (*Objective Function*). The objective function J is continuous, twice differentiable and bounded from below.

Assumption 1b (*Convexity*). In addition to Assumption 1a, the objective function J is convex.

Under the strong convexity condition of Assumption 1b, the optimal decision vector obeys the sufficient condition

$$\frac{\partial J}{\partial u_i}(\mathbf{u}^*) = 0 \quad \forall i. \quad (3)$$

If the weaker condition of Assumption 1a holds, (3) represents the first order necessary condition for a local minimum.

2.2. Asynchronous optimisation

To find a decision obeying (3) an asynchronous gradient descent algorithm will be developed. During the optimisation process each agent maintains a local copy of the current team decision vector which, due to the asynchronous requirements, may contain outdated information about other agents proposed decisions. It is given at discrete time t for agent i as

$$\mathbf{u}^i(t) = [u_1^i(t), \ldots, u_p^i(t)]^{\mathrm{T}}. \quad (4)$$

In general superscripts refer to an agent, while subscripts represent a component of the decision vector (e.g. $u_j^i(t)$ represents agent i's local copy of agent j's component of the team decision vector).

The time variable t is simply used to represent when discrete events take place (such as when an agent computes an update or communicates) and does not require each agent have access to a global clock or perform a periodic synchronisation.

To quantify the effects of communication delays, the variable $\tau_j^i(t)$ will denote when agent i's local copy $u_j^i(t)$ was generated by agent j and hence $u_j^i(t) = u_j^j\left(\tau_j^i(t)\right)$. It is assumed that $\tau_i^i(t) = t$ and thus agent i always has the latest copy of its decision variable. Using this notation (4) can be written as

$$\mathbf{u}^i(t) = [u_1^1\left(\tau_1^i(t)\right), \ldots, u_p^p(\tau_p^i(t))]^{\mathrm{T}}. \quad (5)$$

2.3. Local update

Each agent employs a local update rule $f_i : \mathcal{U} \to \mathcal{U}_i$ that modifies its proposed decision based on what it knows about the decisions of the rest of the team. To allow each agent to perform updates asynchronously, a set of times $T_U^i \subseteq \{1, 2, \ldots\}$ will be associated with each agent i that represent when the agent computes a local update. It is assumed the set T_U^i has an infinite number of elements and thus each agent never stops computing updates

$$u_i^i(t+1) = \begin{cases} f_i\left(\mathbf{u}^i(t)\right) & \text{if } t \in T_U^i \\ u_i^i(t) & \text{else.} \end{cases} \quad (6)$$

For a general gradient descent type algorithm, the update function has the form

$$f_i\left(\mathbf{u}^i(t)\right) = u_i^i(t) + \gamma_i s_i\left(\mathbf{u}^i(t)\right) \quad (7)$$

where γ_i is a constant step size to be defined and $s_i : \mathcal{U} \to \mathcal{U}_i$ is the update direction, e.g. for steepest descent

$$s_i\left(\mathbf{u}^i(t)\right) = -\frac{\partial J}{\partial u_i}\left(\mathbf{u}^i(t)\right). \quad (8)$$

2.4. Communication

Communication is initiated by an agent i sending a message, at some time $t \in T_C^{ij} \subseteq \{1, 2, \ldots\}$, to agent j containing its latest control $u_i^i(t)$, where the set T_C^{ij} is assumed to contain an infinite number of elements. After some communication delay $b_{ij}(t)$ agent j receives it and incorporates it into its local copy of the team decision vector, thus $u_i^j\left(t + b_{ij}(t)\right) = u_i^i(t)$ and $\tau_j^i\left(t + b_{ij}(t)\right) = t$.

Although this only explicitly models direct communication between two nodes, the actual implementation may employ information routing or forwarding via intermediate nodes.

2.5. Convergence results

This section presents a sufficient condition for the above algorithm to converge that quantitatively relates the communication frequency, update direction and the structure of the objective function to a maximum allowable step size for each agent.

Convergence is proved through ensuring cost reduction. This can be accomplished if a bound on the cost function's Hessian is known. This allows the effects of the outdated information one agent has about another to also be bounded.

Assumption 2 (*Local Descent*). For every i and $t \in T_U^i$

(i)

$$s_i\left(\mathbf{u}^i(t)\right) \frac{\partial J}{\partial u_i}\left(\mathbf{u}^i(t)\right) \le 0. \quad (9)$$

(ii) There exists positive constants C_i and D_i such that

$$C_i \left| \frac{\partial J}{\partial u_i}\left(\mathbf{u}^i(t)\right) \right| \le \left| s_i\left(\mathbf{u}^i(t)\right) \right| \le D_i \left| \frac{\partial J}{\partial u_i}\left(\mathbf{u}^i(t)\right) \right|. \quad (10)$$

This assumption ensures the updates made by individual agents are in a direction of decreasing cost (based on local information), are bounded and may only be zero if the gradient of the cost function is zero.

Assumption 3 (*Bounded Delays*). There exists positive constants B_{ij} such that

$$t - \tau_i^j(t) \le B_{ij} \quad \forall i, j, t. \quad (11)$$

This restricts the possible values of T_C^{ij} and $b_{ij}(t)$ by requiring the age of agent j's local copy of i's decision to be bounded. Informally, this can be relaxed such that B_{ij} represents the time difference, measured in the number of updates computed by agent i, between $u_i^i(t)$ and $u_i^j(t)$.

Assumption 4 (*Coupling*). There exist positive constants K_{ij} such that

$$\left|\frac{\partial^2 J}{\partial u_i \partial u_j}(\mathbf{u})\right| \leq K_{ij} \quad \forall i, j, \mathbf{u} \in \mathcal{U}. \tag{12}$$

This imposes a limit on the maximum coupling between the decisions made by different agents, represented by bounds on the second derivatives of the objective function.

Theorem 1 (*General Convergence*). *Assumptions 1a and 2–4 provide sufficient conditions for the distributed optimisation algorithm defined by (6) and (7) to converge with*

$$\lim_{t\to\infty} \frac{\partial J}{\partial u_i}\left(\mathbf{u}^G(t)\right) = 0 \quad \forall i \tag{13}$$

where $\mathbf{u}^G(t) = \left[u_1^1(t), \ldots, u_p^p(t)\right]$ *is the global decision vector formed with the most recent local decision from each agent and* $\gamma_i \in (0, \Gamma_i)$*, where*

$$\Gamma_i = \frac{2}{D_i\left(K_{ij} + \sum_{j\neq i} K_{ij}(1 + B_{ij} + B_{ji})\right)}. \tag{14}$$

See Appendix for proof.

This theorem guarantees that the algorithm will converge to the optimal decision if the cost function is convex. For nonconvex functions, convergence is still guaranteed but only to a local minimum. This general result provides a unified way to relate communication requirements and problem structure for a general cooperative multi-agent system.

2.6. Example 1: Quadratic cost

To demonstrate the convergence properties and communication requirements this section will consider a set of decision makers minimising the simple quadratic objective function

$$J(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{A}\mathbf{u}. \tag{15}$$

For some positive definite matrix $\mathbf{A} = \{A_{ij}\}$. Although this has a trivial solution it will highlight the issues involved.

To completely specify a multi-agent algorithm to solve this, the local update equation (7) and the communication network/protocol must be specified.

- *Local update:* Each agent i computes an update at each time step $T_U^i = \{0, 1, 2, \ldots\}$, where the update direction is defined by the steepest decent direction. Thus

  $$u_i^i(t+1) = u_i^i(t) - \gamma_i \mathbf{A}_i \mathbf{u}^i(t), \quad \forall i \tag{16}$$

  where $\mathbf{A}_i$ is the ith row of $\mathbf{A}$. For this equation, the constant $D_i = 1$.
- *Communication:* Each agent i communicates after each local iteration to all other agents $j \neq i$ and thus $T_C^{ij} = \{1, 2, \ldots\}$. It is assumed that the message delivery delays are constant for each communication link and given by b_{ij}. This results in $u_i^j(t) = u_i^i(t - b_{ij})$. It is noted that for this particular setup, the delivery delay b_{ij} corresponds to the delay terms B_{ij}.

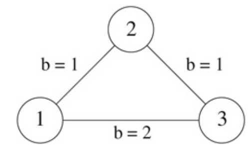

Fig. 1. Fully connected communication network between 3 agents with associated bi-directional communication delays.

The last term that must be specified is the step size γ_i. Based on Theorem 1 γ_i is given by

$$\gamma_i = \frac{\beta}{K_{ii} + \sum_{j\neq i} K_{ij}(1 + B_{ij} + B_{ji})} \tag{17}$$

where $\beta \in (0, 2)$ and for this example is set to a conservative 1.5. The delays terms B_{ij} are defined above, while the coupling terms are given by the matrix elements

$$K_{ij} = \left|\frac{\partial^2 J}{\partial u_i \partial u_j}(\mathbf{u})\right| = \left|A_{ij}\right| \tag{18}$$

for all i and j.

Consider the 3 dimensional case consisting of 3 agents with an $\mathbf{A}$ matrix defined as

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 1.5 \\ 2 & 3 & 2 \\ 1.5 & 2 & 3 \end{bmatrix}. \tag{19}$$

The delay terms are specified by a similar matrix $\mathbf{B}$

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} \tag{20}$$

and is shown in Fig. 1.

To demonstrate the full convergence properties, the local copy of each agents decision vector is initialised to the sum of the eigen vectors $\mathbf{v}_i$ of $\mathbf{A}$,

$$\mathbf{u}^i(0) = \sum_{i=1}^{3} \mathbf{v}_i \approx \begin{bmatrix} -0.89 \\ -1.25 \\ 0.53 \end{bmatrix}, \quad \text{for all } i \in \{1, 2, 3\}. \tag{21}$$

The resulting evolution of the local decision of each agent and the associated value of the objective function is shown in Fig. 2. It is noted from Fig. 2(b) that this distributed algorithm maintains the linear convergence rate typical of steepest descent algorithms approaching a non-singular stationary point. This property has been proved for a similar distributed algorithm in [14].

3. Heterogeneity and system structure

This section examines the general structure of heterogeneous team decision problems and explicitly considers what information each agent requires about the team.

To start, consider a general heterogeneous team of agents. For each agent to evaluate the objective function (or any higher order derivatives) it will not only require the decision of all the other agents but also require a model of how their decision will modify or impact the system.

For large systems, containing many agents, the computational costs of storing and operating with these models, along with the bandwidth required to communicate the local decisions will become inhibitive. To deal with this issue, a particular form of the objective function will be defined that will enable the effect or impact of a group of agents to be described in the same manner as the impact of a single agent. For systems with this type of cost function it will also be shown that each agent will only be required to communicate over an acyclic network.

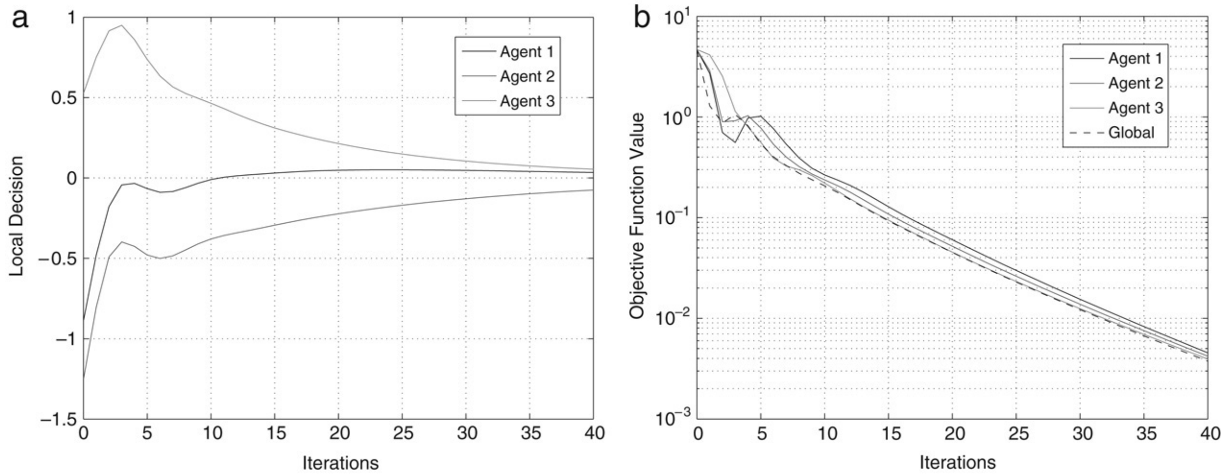

Fig. 2. Typical convergence results of the distributed optimisation algorithm for a quadratic objective function. (a) Local decision variables $u_i^i(t)$ versus iterations t. (b) Objective function value. The dashed line shows the value of the global team decision, $J\left(\mathbf{u}^G(t)\right)$ at each iteration. The solid lines represent the objective function value of the local decision vectors of each agent, $J\left(\mathbf{u}^i(t)\right)$ for each $i = 1, 2, 3$.

3.1. *Partial separability*

This section defines a class of systems where each agent does not require specific information about every other agent. In general terms, it will be shown that if the effect or impact of a group of agents on the team objective or cost function, can be described in the same manner as the impact of a single agent. Then, each agent can consider the rest of the team as a single entity, described in the same manner as any other agent, irrespective of how large the team is.

The key concept behind this is the ability to describe the impact of a group of agents in the same manner (i.e. description length) as the impact of a single agent. Systems that exhibit this property will be called *Partially Separable*.

This enables each agent to independently evaluate its future impact for a proposed local decision. These impacts are then communicated around the team and combined such that each agent passes on an anonymous summary of what it knows about the impact of other agents. This concept will now be formalised.

Definition 1 (*Impact Function*). An impact function Υ_i, of a given agent *i*, abstracts the local sensor and actuator models and maps a decision u_i onto a common team impact space $\mathcal{S}$

$$\Upsilon_i : \mathcal{U}_i \to \mathcal{S}. \tag{22}$$

Given the impacts from two agents, $\alpha_i = \Upsilon_i(u_i)$ and $\alpha_j = \Upsilon_j(u_j)$, the combined impact will be generated using a binary composition operator $*$ defined on the impact space $\mathcal{S}$. This allows two impacts to be combined without losing any task related information.

Definition 2 (*Impact Properties*). For all non-empty and non-overlapping sets of agents $\mu, \nu \subset \{1, \dots, p\}$, such that $\mu \bigcap \nu = \emptyset$ and $\sigma = \mu \bigcup \nu$, there exists

(i) A commutative and associative binary operator $*$ defined for the impacts $\alpha_\mu, \alpha_\nu \in \mathcal{S}$ such that the combined impact α_σ is given by

$$\alpha_\sigma = \alpha_\mu * \alpha_\nu. \tag{23}$$

(ii) An inverse of all elements of $\mathcal{S}$ and a unity element $\alpha_\emptyset$, such that the impact α_μ can be recovered by

$$\alpha_\sigma * \alpha_\nu^{-1} = \alpha_\mu * \alpha_\nu * \alpha_\nu^{-1} = \alpha_\mu * \alpha_\emptyset = \alpha_\mu. \tag{24}$$

This formally defines an *Abelian group*.

A partially separable objective function of the team is defined as a mapping from the combined team impact to the real line. This is in contrast to the previous definition, which defines it as a mapping directly from the team controls u_i to $\Re$.

Definition 3 (*Partial Separability*). The partially separable objective function J is equivalent to the mapping $\psi : \mathcal{S} \to \Re$, given by

$$J(\mathbf{u}) = \psi(\alpha_T) \tag{25}$$

where

$$\alpha_T = \Upsilon_1(u_1) * \Upsilon_2(u_2) * \cdots * \Upsilon_p(u_p). \tag{26}$$

To demonstrate this concept, an reconnaissance scenario will be examined for the case when all uncertainties are Gaussian.

3.2. *Example 2: Reconnaissance with Gaussian Uncertainty*

A typical scenario of interest in this paper is the reconnaissance or information gathering task. This scenario requires the agents to decide on a joint decision that minimises the uncertainty about a particular random variable $\mathbf{x} \in \mathcal{X}$ given some prior information. This state variable may include positions of targets, terrain properties of a given region, or surface information of a remote planet, for example.

This section will show that the reconnaissance scenario, when all uncertainties are described by Gaussian distributions, has a partially separable objective function. For a detailed reference on Gaussian filtering and the extended Kalman filter see [15].

3.2.1. *Prior*

It is assumed that all agents have access to the prior information about probably values of $\mathbf{x}$, defined as a probability density function $P(\mathbf{x})$ and assumed Gaussian with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$

$$P(\mathbf{x}) \sim N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}). \tag{27}$$

3.2.2. *Observation model*

Agent *i*'s observation is modelled by

$$\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}, u_i) + \mathbf{w}_i \tag{28}$$

where $\mathbf{w}_i$ is Normally distributed with zero mean and covariance given by $\mathbf{R}_i(\mathbf{x}, u_i)$. In the implementation of the extended Kalman filter, this nonlinear model is linearised about the prior expected state $\hat{\mathbf{x}}$

$$\mathbf{h}_i(\mathbf{x}, u_i) \approx \mathbf{h}_i(\hat{\mathbf{x}}, u_i) + (\hat{\mathbf{x}} - \mathbf{x})\, \nabla_{\mathbf{x}} \mathbf{h}_i(\hat{\mathbf{x}}, u_i). \tag{29}$$

For simplicity the Jacobian $\nabla_{\mathbf{x}} \mathbf{h}_i(\hat{\mathbf{x}}, u_i)$ will be given as $\mathbf{H}_i(u_i)$ and the mean observation $\mathbf{h}_i(\hat{\mathbf{x}}, u_i)$ denoted by $\hat{\mathbf{z}}_i$. Thus, (29) becomes

$$\mathbf{h}_i(\mathbf{x}, u_i) \approx \hat{\mathbf{z}}_i + (\hat{\mathbf{x}} - \mathbf{x})\mathbf{H}_i(u_i). \tag{30}$$

The observation covariance is approximated by a constant function (in $\mathbf{x}$)

$$\mathbf{R}_i(\mathbf{x}, u_i) \approx \mathbf{R}_i(\hat{\mathbf{x}}, u_i) = \mathbf{R}_i(u_i). \tag{31}$$

3.2.3. Measurement update

For a given team decision $\mathbf{u} = [u_1, \ldots, u_p]^{\mathrm{T}}$ and observation $\mathbf{z} = [\mathbf{z}_1^T, \ldots, \mathbf{z}_p^T]^{\mathrm{T}}$, the posterior $P(\mathbf{x}|\mathbf{z}, \mathbf{u})$ can be computed using an extended Kalman filter update

$$P(\mathbf{x}|\mathbf{z}, \mathbf{u}) \sim N(\mathbf{x}; \hat{\mathbf{x}}_+, \mathbf{P}_+) \tag{32}$$

where the updated mean $\hat{\mathbf{x}}_+$ and covariance $\mathbf{P}_+$, are given in inverse covariance form [15] by

$$\mathbf{P}_+^{-1} = \mathbf{P}^{-1} + \sum_{i=1}^{p} \mathbf{H}_i^T(u_i)\, \mathbf{R}_i^{-1}(u_i)\, \mathbf{H}_i(u_i), \tag{33}$$

and

$$\mathbf{P}_+^{-1}\, \hat{\mathbf{x}}_+ = \mathbf{P}^{-1}\hat{\mathbf{x}} + \sum_{i=1}^{p} \mathbf{H}_i^T(u_i)\, \mathbf{R}_i^{-1}(u_i)(\mathbf{z}_i^k - \hat{\mathbf{z}}_i^k + \mathbf{H}_i^k\, \hat{\mathbf{x}}^{|k-1}). \tag{34}$$

3.2.4. Objective function

The objective of the task is to minimise the uncertainty after the observations have been taken. Thus, a suitable team cost function is the Shannon entropy of the posterior, which measures the uncertainty contained in the distribution (see [16])

$$H_{P(\mathbf{x}|\mathbf{z},\mathbf{u})} = -\mathrm{E}_{\mathbf{x}}[\log P(\mathbf{x}|\mathbf{z}, \mathbf{u})]\,. \tag{35}$$

For a Gaussian density this becomes

$$H_{P(\mathbf{x}|\mathbf{z},\mathbf{u})} = -\frac{1}{2} \log\left((2\pi e)^{d_x} \left|\mathbf{P}_+^{-1}\right|\right) \tag{36}$$

where d_x is the dimension of the state space $\mathcal{X}$. It is noted that this entropy is independent of the observation $\mathbf{z}$. This is one of the special properties of Gaussian filtering and does not occur in general. (In general some observations will produce a more informative posterior than others.)

Because of this independency the posterior entropy can be used directly as the objective or cost function

$$J(\mathbf{u}) = -\frac{1}{2} \log\left((2\pi e)^{d_x} \left|\mathbf{P}_+^{-1}\right|\right). \tag{37}$$

Using (33) and noting the objective function is only required as a preference ordering over actions, (37) becomes

$$J(\mathbf{u}) = -\left|\mathbf{P}^{-1} + \sum_{i=1}^{p} \mathbf{I}_i(u_i)\right| \tag{38}$$

where

$$\mathbf{I}_i(u_i) = \mathbf{H}_i^T(u_i)\, \mathbf{R}_i^{-1}(u_i)\, \mathbf{H}_i(u_i). \tag{39}$$

To evaluate this function (or gradient), only the matrix sum, $\sum_{i=1}^{p} \mathbf{I}_i(u_i)$, is required. Knowledge of the decisions u_i, models $\mathbf{H}_i$ and $\mathbf{R}_i$ or the individual matrices $\mathbf{I}_i$ from each agent is redundant.

This structure can be emphasised by defining an impact function for the ith agent as

$$\Upsilon_i(u_i) = \mathbf{I}_i(u_i) = \mathbf{H}_i^T(u_i)\, \mathbf{R}_i^{-1}(u_i)\, \mathbf{H}_i(u_i). \tag{40}$$

This abstracts the agents models by extracting only the task relevant features. It directly maps a decision onto the common impact space $\mathcal{S}$, of all $d_x \times d_x$ symmetric matrices.

With this formulation, the composition operator $*$ becomes matrix addition, an elements inverse is its negative, and the unity element becomes the zero matrix. It is also noted that the size of the impact space, $\mathcal{S}$, is only dependent on the size of the state space $\mathcal{X}$ and is independent of the number of agents.

The team cost function, in partially separable form is given by

$$J(\mathbf{u}) = \psi(\alpha_T) = -\left|\mathbf{P}^{-1} + \alpha_T\right|. \tag{41}$$

where

$$\begin{aligned}\alpha_T &= \Upsilon_1(u_1) * \cdots * \Upsilon_p(u_p) \\ &= \sum_{i=1}^{p} \mathbf{I}_i(u_i).\end{aligned} \tag{42}$$

This formulation only requires each agent to consider the combination of its own impact and that of the rest of the team, nothing else about the team is required.

4. Anonymous decentralised optimisation

The decentralised optimisation algorithm described in Section 2 will now be specialised for the structure of partially separable systems introduced in Section 3. The motivation behind this algorithm has come from the work of Grocholsky in [8,9] but can be applied to any Partially Separable cost function.

The original algorithm required each agent to maintain a local copy of the team decision vector, containing the (possibly outdated) decisions from all members of the team. For a partially separable system this requirement can be fulfilled by each agent maintaining a local copy of the combined team impact α_T^i, which due to the asynchronicity, may contain old information about other agents.

4.1. Decision update

Each agent incrementally updates or refines its local decision based on its current (possible outdated) knowledge of the team, through the update equation

$$f_i\left(\mathbf{u}^i(t)\right) = u_i^i(t) + \gamma_i s_i\left(\mathbf{u}^i(t)\right). \tag{43}$$

It will be assumed that the decision is moved in a direction of steepest descent

$$s_i\left(\mathbf{u}^i(t)\right) = -\frac{\partial J}{\partial u_i}\left(\mathbf{u}^i(t)\right) = -\frac{\partial \psi}{\partial u_i}\left(\alpha_T^i(t)\right). \tag{44}$$

Here $\alpha_T^i(t)$ represents the local copy of the combined team impact maintained by agent i at time t. Expanding the team impact into components from agent i and the rest of the team $\bar{\imath} = \{j : \forall j \neq i\}$

$$\begin{aligned}\alpha_T^i(t) &= \alpha_i^i(t) * \alpha_{\bar{\imath}}^i(t) \\ &= \Upsilon_i\left(u_i^i(t)\right) * \alpha_{\bar{\imath}}^i(t),\end{aligned} \tag{45}$$

the update direction, in (44), becomes

$$s_i\left(\mathbf{u}^i(t)\right) = -\frac{\partial \psi}{\partial u_i}\left(\Upsilon_i\left(u_i^i(t)\right) * \alpha_{\bar{\imath}}^i(t)\right). \tag{46}$$

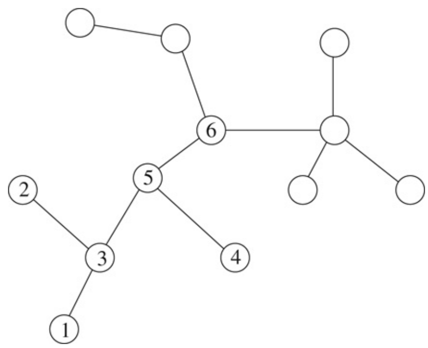

Fig. 3. Acyclic communication network. Agent 6 can only receive information about agents 1−4 via 5. This ensures the impacts from these agents are only counted once.

Thus, the update equation can be written as

$$u_i^i \Leftarrow u_i^i - \gamma_i \frac{\partial \psi}{\partial u_i} \left(\Upsilon_i \left(u_i^i(t) \right) * \alpha_{\bar{i}}^i(t) \right). \tag{47}$$

This enables each agent to continually refine its decision while only requiring very little information about the rest of the team. The impact $\alpha_{\bar{i}}^i$, abstracts the decisions, models and even the number of agents in rest of the team. This enables each agent to calculate the update directions in a scalable fashion.

4.2. Communication topology

As each agent locally modifies its decision, it changes its contribution to the team impact. Hence, for each agent to reconstruct the full team impact, it must receive information from every other agent. The simplest method to accomplish this is a fully connected communication network where every agent sends its impact to every other agent, as used for Example 1. However, this technique is not scalable, since each agent must send and receive a number of messages proportional to the size of the team.

To overcome this issue, each agent i is restricted to only communicate to agents in a local neighbourhood $\mathcal{N}_i$ of some communication graph. Each agent combines the impacts it receives and passes it on, such that eventually every agent has a team impact that contains information from every other agent.

However, due to the anonymous nature of the impacts, if the network contains loops, it is impossible for an agent to know which other agents it has information about and double counting information can happen. This is similar to *rumour propagation* in data fusion networks [17].

To prevent this, the communication network will be restricted to an acyclic tree network. This ensures each agent can only receive information about any other particular agent through a single, third agent. This property is illustrated in Fig. 3.

4.3. Message structure

The changes in each agent's local impact (and changes received from other agents) are communicated in the form of update messages. These messages represent the difference between the previous impact and the updated one.

To ensure information is not lost or double counted, each agent i maintains a buffer $C_{ij} \in \mathcal{S}$ for each local communication link to a neighbouring agent $j \in \mathcal{N}_i$. This contains the component of the team impact that is known by both agents i and j. The buffer is initialised with the identity element $\alpha_\emptyset$, representing that nothing is common.

Each time agent i is required to communicate to agent $j \in \mathcal{N}_i$, it sends an update message $m_{ij} \in \mathcal{S}$ by first removing this common component (i.e. by combining its inverse)

$$m_{ij} = \alpha_T^i * C_{ij}^{-1}. \tag{48}$$

This message contains the differences between what it has communicated before and what it now knows (by changing its own impact or through communications from other agents). Once the message is sent (and it is know to be received) the agent must update its buffer by setting

$$C_{ij} \Leftarrow C_{ij} * m_{ij} = \alpha_T^i. \tag{49}$$

This represents that agent j will know everything that it knew previously about the team and what was just sent.

On reception, agent j combines the message with its current team impact

$$\alpha_T^j \Leftarrow \alpha_T^j * m_{ij} \tag{50}$$

and similarly updates its buffer

$$C_{ji} \Leftarrow C_{ji} * m_{ij}. \tag{51}$$

By noting that the messages an agent receives is based on changes to other agents impacts (not its own), (50) can be written as

$$\alpha_j^j * \alpha_{\bar{j}}^j \Leftarrow \alpha_j^j * \alpha_{\bar{j}}^j * m_{ij} \tag{52}$$

and thus

$$\alpha_{\bar{j}}^j \Leftarrow \alpha_{\bar{j}}^j * m_{ij}. \tag{53}$$

This allows the impact from the rest of the team $\alpha_{\bar{j}}^j$ to be separated and updated independently from that of its own impact α_j^j.

The generic internal structure of each agent is shown in Fig. 4. The left hand side of the diagram displays the optimisation loop, where local decisions are continually refined. On the right, the external information regarding the rest of the teams impact $\alpha_{\bar{i}}^i$ (which the refinements are based on) is continually updated via the communication links.

4.4. Update step size

The convergence results of Theorem 1, provides a quantitative relationship between the allowable step size γ_i, the age of the information contained in the local team impact α_T^i, and the inter-agent coupling.

However, these parameters will generally require global knowledge of the system, i.e. all agent models, communication protocols and network topology, and must be specified *a priori*. One approach to overcome this requirement, is to supply each agent with some maximum bound which allow it to produce a conservative estimate. This will be demonstrated in the following example. Other methods to guarantee an appropriate step size is employed are given in [18].

4.5. Example 3: Feature localisation

This is a continuation of the previous example of reconnaissance with Gaussian uncertainties. It considers the multi-agent decision problem pictured in Fig. 5. The agents represent range only sensors and they are required to select the observation angles such the posterior uncertainty of a 2D point feature is minimised.

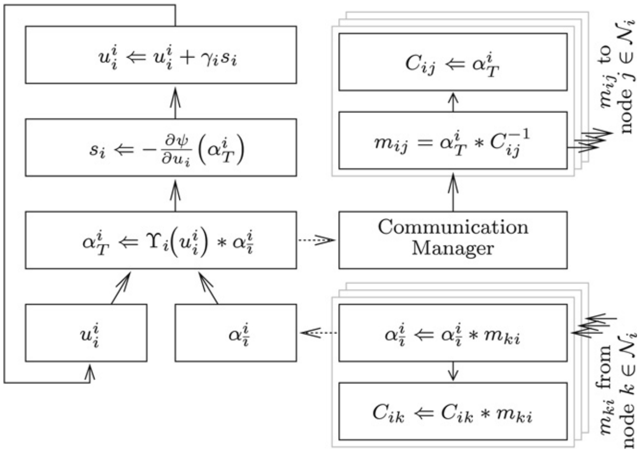

Fig. 4. Internal agent structure. Dashed arrows indicate asynchronous information flows. The left half of the algorithm incrementally refines the agents decision based on the local team impact. While the right side continually updates the local team impact that the decisions are based on.

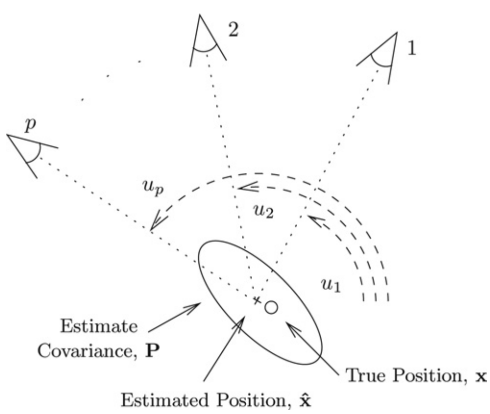

Fig. 5. Feature localisation Task. Each agent is equipped with a range sensor and must decide on which angle to view the feature.

4.5.1. Models

The state $\mathbf{x}$ described the position of the feature

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}. \tag{54}$$

The agents have access to a common Gaussian prior distribution about the location of the feature

$$P(\mathbf{x}) \sim N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}), \tag{55}$$

and are equipped with range sensors that produce observations according to

$$\begin{aligned} \mathbf{z}_i &= \hat{r}_i + w_i \\ &= \sqrt{(x - x_i)^2 + (y - y_i)^2} + w_i \end{aligned} \tag{56}$$

where the disturbance w_i is zero mean with variance σ_i^2 and x_i and y_i are the positions of the sensors. It is assumed that the variances are independent of the range and thus it is expected that only the bearing to the estimate $\hat{\mathbf{x}}$ will be of interest

$$u_i = \tan^{-1} \frac{\hat{y} - y_i}{\hat{x} - x_i}. \tag{57}$$

The team decision problem involves selecting the observation angle u_i such that the expected posterior uncertainty is minimised. Using the formulation given in (41) and (42), the objective function becomes

$$J(\mathbf{u}) = -|\mathbf{P}^{-1} + \alpha_T| \tag{58}$$

where

$$\alpha_T = \Upsilon_1(u_1) + \cdots + \Upsilon_p(u_p) \tag{59}$$

and

$$\Upsilon_i(u_i) = \mathbf{H}_i^T(u_i)\, \mathbf{R}_i^{-1}(u_i)\, \mathbf{H}_i(u_i). \tag{60}$$

The Jacobian, linearised about the prior mean, is given by

$$\begin{aligned} \mathbf{H}_i(u_i) &= \nabla_{\mathbf{x}} \mathbf{h}_i(\hat{\mathbf{x}}, u_i) \\ &= \left[\frac{\hat{x} - x_i}{\sqrt{(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2}} \quad \frac{\hat{y} - y_i}{\sqrt{(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2}} \right] \\ &= \left[\cos(u_i) \quad \sin(u_i) \right], \end{aligned} \tag{61}$$

and the observation covariance is simply the scalar variance

$$\mathbf{R}_i(u_i) = \sigma_i^2. \tag{62}$$

Thus, the impact function becomes

$$\Upsilon_i(u_i) = \frac{1}{\sigma_i^2} \begin{bmatrix} \cos^2(u_i) & \cos(u_i)\sin(u_i) \\ \cos(u_i)\sin(u_i) & \sin^2(u_i) \end{bmatrix}. \tag{63}$$

The model of each agent is defined by their Jacobian $\mathbf{H}_i(u_i)$ and covariance $\mathbf{R}_i(u_i)$. For this scenario the agents only differ through their observation variances, σ_i^2, which is assumed only known by the corresponding agent (this is analogous to each agent only knowing its own model).

Each agent employs the internal algorithm depicted in Fig. 4. The decisions are updated by calculating a direction, using (46) and an appropriate step size. The gradient calculation only requires the impact of the rest of the team, and thus nothing about the *individual* variances of the other agents sensors is required.

4.5.2. Step size and inter-agent coupling

To determine if a step size γ_i is appropriate, knowledge of the Hessian and network delays are required. This can be observed by examining the results of Theorem 1.

The coupling coefficients are related to the maximum of the Hessian elements

$$K_{ij} \geq \left| \frac{\partial^2 J}{\partial u_i \partial u_j}(\mathbf{u}) \right| \quad \forall \mathbf{u} \in \mathcal{U} \tag{64}$$

and for this system are given by

$$\max_{\mathbf{u} \in \mathcal{U}} \left| \frac{\partial^2 J}{\partial u_i \partial u_j}(\mathbf{u}) \right| = \frac{2}{\sigma_i^2 \sigma_j^2} \quad \text{if } i \neq j \tag{65a}$$

or

$$\max_{\mathbf{u} \in \mathcal{U}} \left| \frac{\partial^2 J}{\partial {u_i}^2}(\mathbf{u}) \right| = \sqrt{\frac{4(Y_x - Y_y)^2}{\sigma_i^4} + \frac{16 Y_{xy}^2}{\sigma_i^4}} + \frac{2}{\sigma_i^2} \sum_{j \neq i} \frac{1}{\sigma_j^2} \tag{65b}$$

where the terms Y_x, Y_y and Y_{xy} are elements of the prior information or inverse covariance matrix,

$$\mathbf{P}^{-1} = \begin{bmatrix} Y_x & Y_{xy} \\ Y_{xy} & Y_y \end{bmatrix}. \tag{66}$$

The maximum step size, using the steepest descent direction, is

$$\Gamma_i = \frac{2}{K_{ii} + \sum_{j \neq i} K_{ij} \left(1 + B_{ij} + B_{ji}\right)}. \tag{67}$$

Thus it can be seen explicitly, for each agent to calculate the maximum allowable step size, the observation variances (σ_j^2) from each other agents and the communication delays caused by the network topology (B_{ij} and B_{ji}) are required.

Although this seems to suggest global knowledge is required by each agent, only an upper bound on these variables (K_{ij} and B_{ij}) are required. Hence, a smaller upper bound $\bar{\Gamma}_i \leq \Gamma_i$ for the step size can be obtained through more general knowledge of the team, such as the minimum observational variance in the team $\sigma_{\min}$, total number of agents p, and the diameter of the communication network $B_{\max}$.

Estimates of the coupling coefficients can be calculated locally by each agent using

$$\bar{K}_{ij} = \frac{2}{\sigma_i^2 \sigma_{\min}^2} \quad \text{if } i \neq j \tag{68}$$

and

$$\bar{K}_{ii} = \sqrt{\frac{4(Y_x - Y_y)^2}{\sigma_i^4} + \frac{16Y_{xy}^2}{\sigma_i^4}} + \frac{2p}{\sigma_i^2 \sigma_{\min}^2}. \tag{69}$$

Thus, a bound of the upper limit, which is guaranteed to be less than or equal to the true limit, is given by

$$\bar{\Gamma}_i = \frac{2}{\bar{K}_{ii} + \sum_{j \neq i} \bar{K}_{ij}\,(1 + 2B_{\max})} \leq \Gamma_i. \tag{70}$$

This will usually be a very conservative estimate and hence, will be used as the actual step size

$$\gamma_i = \bar{\Gamma}_i. \tag{71}$$

4.6. Numerical results

The convergence properties of this decision problem will now be demonstrated using two simulations. Each agent employs the internal algorithm shown in Fig. 4. The agents perform simultaneous updates and communicates to its neighbours after every local iteration (i.e. $T_U^i = \{0, 1, 2, \ldots\}$ and $T_C^{ij} = \{1, 2, \ldots\}$ for all i and $j \neq i$). Thus, it will be assumed information about an agents impact propagates a single hop in the acyclic network each iteration.

Each agent has a random observation variance in the interval $[1, 2]$, starts with a randomly initialised decision between $-\pi$ and π and knows nothing about the impact of the rest of the team.

For both examples, the same prior density is used for the feature, namely a Gaussian with zero mean and an inverse covariance of

$$\mathbf{P}^{-1} = \begin{bmatrix} 0.75 & 0.2 \\ 0.2 & 0.5 \end{bmatrix}. \tag{72}$$

4.6.1. Two agents

For this simulation the true coupling coefficients were used and the step size set as $\gamma_i = 0.75\Gamma_i$, corresponding to a $\beta = 1.5$. The evolution of the global team decision is shown in Fig. 6, along with contours of the team objective function. It is noted that this information is not available to any of the agents as each agent only knows about the other through the communicated impacts.

The convergence of the value of the objective function is shown in Fig. 7, while Fig. 8 shows the final optimal team decisions.

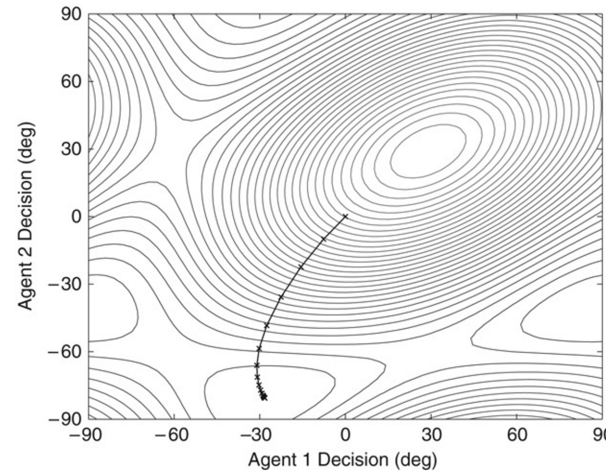

Fig. 6. Trajectory of the team decision through the decision space $\mathcal{U}$ displayed with contours of the team objective function. Each agent updates it decision based on old information from the other. Global information about network topology and the coupling coefficients are used to determine the allowable step size.

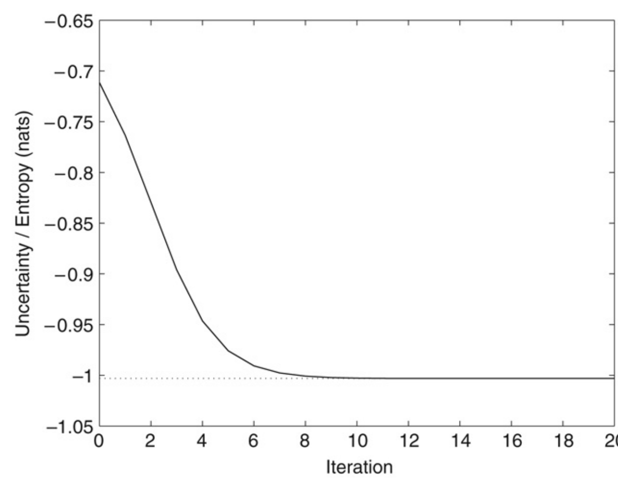

Fig. 7. Value of the objective function for the trajectory shown in Fig. 6. The dotted line represents the minimum cost found using the centralised optimisation algorithm, MATLAB's fminsearch function.

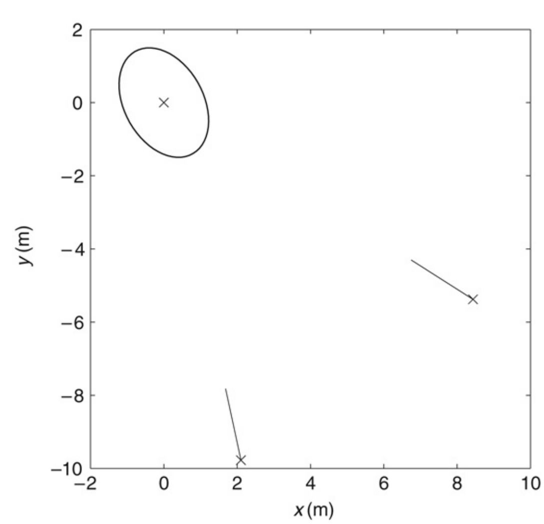

Fig. 8. Prior feature covariance ellipse displayed with the optimal observation angles of the two agents.