Asynchronous Gradient-Based Optimisation for Team Decision Making

George Mathews*†‡, Hugh Durrant-Whyte* and Mikhail Prokopenko§
*ARC Centre of Excellence for Autonomous Systems, The University of Sydney, Sydney, NSW
†CSIRO Industrial Physics, Lindfield, NSW
§CSIRO ICT Centre, North Ryde, NSW
‡Corresponding author, Email: g.mathews@cas.edu.au

***Abstract*— This paper describes a decentralised asynchronous algorithm for negotiation in team decision and control problems, allowing multiple decision makers to propose and refine future decisions to optimise a common non-linear objective or cost function. A convergence requirement provides an intuitive relationship between the communication frequency, transmission delays and the degree of inter-agent coupling inherent in the system. The coupling is defined by the cross derivative of the objective function. The algorithm is applied to the control of multiple vehicles performing a search task with simulation results given.**

I. Introduction

This paper examines the decentralised optimisation problem encounted in continuous team decision problems (e.g. distributed model predicitive control [1]). Classically this optimisation problem is solved using conventional algorithms by collecting the information at a single location [2], [3]. However, for large distributed systems containing many agents, this centralised approach is not practical. To address this, various distributed optimisation algorithms [4]–[6] have been proposed. Generally, these methods decompose the single iterative optimisation algorithm into several smaller subproblems, allowing the computation to be distributed among the agents.

However, the iterative structure of a distributed algorithm can greatly affect its performance. Distributed algorithms can be categorised into synchronous and asynchronous [6] methods. Synchronous algorithms require iterations to be executed in a predetermined order and can be broken down into Gauss-Seidel and Jacobi types (see Fig. 1). The iterations in a Gauss-Seidel algorithm must be computed sequentially, possibly causing the agents to be idle while waiting for their turn. The collision avoidance algorithm presented in [7] is an example of this type.

A Jacobi type algorithm allows the agents to work on subproblems in parallel, but still requires each agent to wait until information from all other agents has been received prior to starting a new iteration. The formation control algorithm presented in [8] falls in this category.

An asynchronous algorithm is similar to a Jacobi type, but does not require each agent to wait before starting the next iteration, an agent simply uses all the information it has available at the time. This type of algorithm allows each agent to compute and communicate at different rates without the overall progress being limited by the slowest agent.

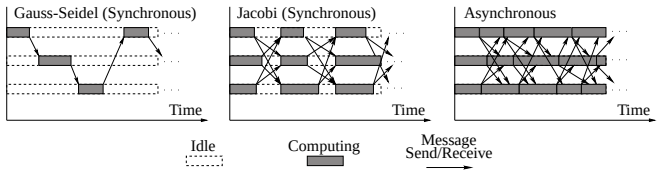

Fig. 1. Synchronous vs asynchronous iterations for a system of 3 agents.

Asynchronous algorithms can be further separated into totally asynchronous and partially asynchronous [9]. For totally asynchronous algorithms, the age of the information an agent has about another can become arbitrary large. While a partially asynchronous algorithm requires this to be bounded. The ADOPT algorithm [10] for distributed constraint optimization problems with discrete decisions is an example of a totally asynchronous algorithm.

The main contributions of this work are twofold: (i) the development of a practical decentralised partially asynchronous optimisation algorithm for a team of agents coupled via a common nonlinear objective function and (ii) a convergence condition for the algorithm is presented which intuitively relates the inherent inter-agent coupling to the communication frequency and transmission delays.

Section II defines the team decision problem and introduces the decentralised solution method and presents the main convergence result. Section III introduces a method to estimate the inter-gent coupling and the general asynchronous optimisation algorithm. Simulations results are presented for a multi-vehicles search scenario in Section IV. Conclusions are presented in Section V.

II. Decentralised Decision Making

Consider a team of p robotic agents, where each agent $i \in \{1, \ldots, p\}$ is in charge of a local decision variable $\mathbf{v}_i \in \mathcal{V}_i \subseteq \Re^{d_i}$. The global team decision vector[1] is given by $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p]$ and is defined on the product set $\mathcal{V} = \mathcal{V}_1 \times \cdots \times \mathcal{V}_p \subseteq \Re^d$, where $d = d_1 + \cdots + d_p$.

Assumption 1 (Decisions): The set of feasible decisions $\mathcal{V}_i$ is convex for all agents $i \in \{1, \ldots, p\}$.

The team is required to select decisions such that a given objective function $J \colon \mathcal{V} \to \Re$ is minimised

$$\mathbf{v}^* = \arg\min_{\mathbf{v} \in \mathcal{V}} J(\mathbf{v}), \tag{1}$$

[1]All vectors are assumed to be column vectors. For simplicity the 'transpose' has been omitted. Formally, $\mathbf{v}$ should be defined as $[\mathbf{v}_1^T, \ldots, \mathbf{v}_p^T]^T$.

where $\mathbf{v}^*$ is the desired optimal global decision. It is noted that this definition of the decision problem excludes hard constraints between the agents.

Assumption 2 (Objective Function): The objective function J is twice differentiable, convex and bounded from below.

Under assumptions 1 and 2, a necessary and sufficient condition [11] for $\mathbf{v}^*$ to be optimal is

$$(\boldsymbol{\nu} - \mathbf{v}^*)^T \nabla J(\mathbf{v}^*) \geq 0, \text{ for all } \boldsymbol{\nu} \in \mathcal{V}. \tag{2}$$

In terms of each agents local decision, this can be written as

$$(\boldsymbol{\nu}_i - \mathbf{v}_i^*)^T \nabla_i J(\mathbf{v}^*) \geq 0, \text{ for all } \boldsymbol{\nu}_i \in \mathcal{V}_i, \tag{3}$$

for all i, where $\mathbf{v}_i^*$ is the i^{th} agents component of the optimal decision $\mathbf{v}^*$ and $\nabla_i J(\mathbf{v}^*) \in \Re^{d_i}$ is the gradient vector with respect to the i^{th} agents decision, evaluated at $\mathbf{v}^*$.

It is noted that if J is not convex, (3) corresponds to the first order necessary condition for the point $\mathbf{v}^*$ to be a local minima.

To explicitly exclude the possibility of a centralised solution being considered, it is assumed that each agent can only compute the local gradient of the objective function $\nabla_i J(\mathbf{v}) \in \Re^{d_i}$ and possibly the local 2nd order derivatives $\nabla_{ii}^2 J(\mathbf{v})$ corresponding to the $d_i \times d_i$ submatrix of the full Hessian.

A. Distributed Asynchronous Optimisation

The proposed solution method allows each agent to propose an initial decision and then to incrementally refining it, while intermittently communicating these refinements to the team. The distributed nature of the problem requires each agent to execute and communicate asynchronously, thus the information it has about other agents may be outdated. It is assumed each agent maintains a local copy of the overall team decision. At discrete time l for agent i this is given by

$${}^i\mathbf{v}(l) = [{}^i\mathbf{v}_1(l), {}^i\mathbf{v}_2(l), \ldots, {}^i\mathbf{v}_p(l)]. \tag{4}$$

Where pre-superscripts represents a copy held by a specific agent, while subscripts represent a specific agents decision, e.g. ${}^i\mathbf{v}_j(l)$ represents agent i's local copy of agent j's decision.

The variable l 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.

To include the effects of communication delays, the variable $\tau_{ij}(l)$ will denote when agent j's local copy ${}^j\mathbf{v}_i(l)$ was generated by agent i and hence ${}^j\mathbf{v}_i(l) = {}^i\mathbf{v}_i\big(\tau_{ij}(l)\big)$. It is assumed that $\tau_{ii}(l) = l$ and thus agent i always has the latest copy of its own decision. Using this notation, (4) can be written as

$${}^i\mathbf{v}(l) = \big[{}^1\mathbf{v}_1\big(\tau_{1i}(l)\big), \ldots, {}^p\mathbf{v}_p\big(\tau_{pi}(l)\big)\big]. \tag{5}$$

B. Local Decision Refinement

To formalise the intuitive notion of *decision refinement* a local update rule $f_i : \mathcal{V} \to \mathcal{V}_i$ is defined for each agent that modifies its local decision ${}^i\mathbf{v}_i$, based on its copy of the team decision vector ${}^i\mathbf{v}$. To allow each agent to perform updates asynchronously a set of times $T_i^U \subseteq \{0, 1, 2, \ldots\}$ is associated with each agent i that represent when the agent computes a local update

$${}^i\mathbf{v}_i(l+1) = \begin{cases} f_i\big({}^i\mathbf{v}(l)\big) & \text{if } l \in T_i^U, \\ {}^i\mathbf{v}_i(l) & \text{else.} \end{cases} \tag{6}$$

To update the decision, each agent i determines the local gradient vector $\nabla_i J\big({}^i\mathbf{v}(l)\big)$ and applies a positive definite scaling matrix $\mathbf{A}_i(l)$ (e.g. an approximation to the local Hessian $\nabla_{ii}^2 J\big({}^i\mathbf{v}(l)\big)$ or simply the identity matrix $\mathbf{I}$). The decision is then updated by moving it in this direction and projecting it back onto a feasible decision

$$f_i\big({}^i\mathbf{v}(l)\big) = \mathrm{Proj}_{\mathcal{V}_i}^{\mathbf{A}(l)}\Big({}^i\mathbf{v}_i(l) - \gamma_i \left(\mathbf{A}_i(l)\right)^{-1} \nabla_i J\big({}^i\mathbf{v}(l)\big)\Big), \tag{7}$$

where γ_i is a step size to be defined and $\mathrm{Proj}_{\mathcal{V}_i}^{\mathbf{A}_i}(\cdot)$ denotes the projection onto the set $\mathcal{V}_i$ under the scaling of $\mathbf{A}_i$ and is defined for any vector $\boldsymbol{\mu}_i \in \Re^{d_i}$ as

$$\mathrm{Proj}_{\mathcal{V}_i}^{\mathbf{A}_i}(\boldsymbol{\mu}_i) = \underset{\boldsymbol{\nu}_i \in \mathcal{V}_i}{\arg\min} \, (\boldsymbol{\nu}_i - \boldsymbol{\mu}_i)^T \, \mathbf{A}_i \, (\boldsymbol{\nu}_i - \boldsymbol{\mu}_i). \tag{8}$$

The projection operation requires only local knowledge of the feasible set $\mathcal{V}_i$ and the scaling matrix $\mathbf{A}_i(l)$.

Assumption 3 (Continuous Computation): There exists a finite constant C, such that for all l and i

$$\exists\, q \in T_i^U, \text{ such that } q \in [l, l + C]. \tag{9}$$

This ensures that the interval between updates computed by each agent is bounded and thus, as $l \to \infty$ the number of updates computed by each agent also goes to ∞.

C. Communication

Communication is initiated by agent i sending a message, at some time $l \in T_{ij}^C \subseteq \{0, 1, 2, \ldots\}$ to agent j containing its latest decision ${}^i\mathbf{v}_i(l)$. After some communication delay $b_{ij}(l)$ agent j receives and incorporates it into its local copy of the team decision vector. Thus, when the message is received ${}^j\mathbf{v}_i\big(l + b_{ij}(l)\big) = {}^i\mathbf{v}_i(l)$ and hence $\tau_{ij}\big(l + b_{ij}(l)\big) = l$. To ensure each agent obtains the decision of each other agent, all agents must communicate to every other agent.

Assumption 4 (Bounded Delays): There exists finite positive constants B_{ij} for all i and j such that

$$l - \tau_{ij}(l) \leq B_{ij}, \quad \text{for all } l. \tag{10}$$

Informally, this can be relaxed so that B_{ij} represents the maximum difference, measured in numbers of updates computed by agent i, between ${}^i\mathbf{v}_i(l)$ and ${}^j\mathbf{v}_i(l)$.

It is noted that if the agents compute updates and communicate at a fixed frequency, the terms B_{ij} of assumption 4 can be approximated by knowing: (i) the rate R_i of iterations computed by agent i, (ii) the communicated rate C_{ij} from i to j, and (iii) the delivery delay D_{ij} between agent i sending and j receiving a message, using

$$\hat{B}_{ij} = R_i / C_{ij} + R_i \, D_{ij}. \tag{11}$$

D. Convergence Results

The condition presented here relates the degree of inter-agent coupling and communication properties to the maximum allowable step size. For each pair of agents, the magnitude of inter-agent coupling is captured by a single scalar that represents the maximum curvature of the objective function in the subspace of the two agents decisions.

Assumption 5 (Coupling): The second derivatives of the objective fucntion are finite. Thus, for every i and j, there exists a finite positive constant K_{ij}, such that

$$\boldsymbol{\nu}_i^T \, \nabla_{ij}^2 J(\mathbf{v}) \, \boldsymbol{\nu}_j \leq \|\boldsymbol{\nu}_i\| \, K_{ij} \, \|\boldsymbol{\nu}_j\|, \tag{12}$$

for all $\mathbf{v} \in \mathcal{V}$ and $\boldsymbol{\nu}_i \in \Re^{d_i}$ and $\boldsymbol{\nu}_j \in \Re^{d_j}$. The matrix $\nabla^2_{ij} J(\mathbf{v})$ corresponds to the $d_i \times d_j$ submatrix of the Hessian of the objective function.

The above assumption defines the inter-agent coupling K_{ij} for all agents i and $j \neq i$. For the case $j = i$, the term K_{ii} will be referred to as the internal coupling of agent i.

If the Hessian is available, the constants K_{ij} can be computed using

$$K_{ij} = \max_{\mathbf{v} \in \mathcal{V}} \overline{s}\big(\nabla^2_{ij} J(\mathbf{v})\big), \tag{13}$$

where $\overline{s}(\mathbf{M})$ represents the maximum singular value of matrix $\mathbf{M}$.

Theorem 1 (Convergence): Assumptions 1 – 5 provide sufficient conditions for the distributed optimisation algorithm defined by (6) and (7) to converge to the global optimum for all $\gamma_i \in (0, \Gamma_i)$ where

$$\Gamma_i = \frac{2\, a_i}{K_{ii} + \sum_{j \neq i} K_{ij}\big(1 + B_{ij} + B_{ji}\big)} \tag{14}$$

and a_i is a scaling normalisation factor given by

$$a_i = \min_{l \in \mathcal{I}_i^C} \underline{s}\big(\mathbf{A}_i(l)\big), \tag{15}$$

where $\underline{s}(\mathbf{M})$ represents the minimum singular value of matrix $\mathbf{M}$. See [12] for a proof. If the objective function is not convex, only convergence to a stationary point, obeying first order optimality conditions, is guaranteed.

This theorem is an extension to the partially asynchronous convergence theorem given in [5] and provides a unified way of relating the inherent inter-agent coupling and communication structure with the speed at which an agent can refine its local decision.

Based on Theorem 1 an algorithm can be developed by defining the step size as

$$\gamma_i = \frac{\beta\, a_i}{K_{ii} + \sum_{j \neq i} K_{ij}\big(1 + B_{ij} + B_{ji}\big)}, \tag{16}$$

for some $\beta \in (0, 2)$. With this formulation, the main issue becomes the evaluation of the coupling terms K_{ij}. This poses a significant problem, even for the internal coupling terms K_{ii}, since they require a solution to the maximisation problem (13), which maybe as hard as the initial optimisation problem. The inter-agent coupling terms pose an additional problem of requiring non-local information about the cross derivatives. These issues will be discussed further in Section III.

E. Communication and Convergence Rate

It has been previously shown [13] that the convergence rate of a similar algorithm based directly on the work of Bertsekas and Tsitsiklis [6] is linear, with an asynchronous convergence ratio of

$$\sqrt{1 - \gamma c}, \tag{17}$$

where γ is a step size fixed for all agents and c is a problem dependent constant. Although this expression was derived for a slightly different algorithm, it is expected that the general dependence will hold for the one defined by (6), (7) and (16), which uses a separate step size for each agent. Thus, this section will examine the convergence rate indirectly by examining the relationship between the properties of the inter-agent communications and the step sizes.

By assuming the communication and computation rates, and delivery delays are constant, the approximation for the delay terms B_{ij} given in (11) can be used. Combining this with (16), the step size obeys the relation

$$\frac{\beta\, a_i}{\gamma_i} = K_{ii} + \sum_{j \neq i} K_{ij}\Big(1 + R_i D_{ij} + R_j D_{ji} + \frac{R_i}{C_{ij}} + \frac{R_j}{C_{ji}}\Big). \tag{18}$$

Now consider the properties of each term:

- R_i: Computation rate of agent i, dependent on available computing power, will be considered a constant.
- C_{ij}: Communication rate from i to j, directly controllable be agent i, possibly subject to bandwidth constraints.
- D_{ij}: Transmission delay from i to j, dependent on physical communication infrastructure and possibly communication topology (if multi-hop routing or message forwarding is used).

For simplicity this work will assume the network topology is fully connected with fixed delays. A heuristic communication policy is proposed which sets the communication rate proportional to the square root of the inter-agent coupling

$$C_{ij} = \eta_i \sqrt{K_{ij}}. \tag{19}$$

The constant η_i can be chosen such that the strongest coupled agent is sent a message after every local iteration, or to satisfy some bandwidth limitations. This policy represents a compromise between fast convergence, requiring a high communication rate, and the practical requirements of limited bandwidth (see [12] for details).

III. Coupling Estimation

Since it cannot be assumed each agent can evaluate the full cross derivatives, it is not possible for any agent to solve (13) for the coupling terms. This section will present an approximation method for these terms that only relies on local information.

Consider the Taylor expansion of J about a decision vector $\mathbf{v}$ with a perturbation $\Delta\mathbf{v} = [\Delta\mathbf{v}_1, \ldots, \Delta\mathbf{v}_p]$

$$J(\mathbf{v} + \Delta\mathbf{v}) \approx J(\mathbf{v}) + \sum_{i=1}^{p} \Delta\mathbf{v}_i^T\, \nabla_i J(\mathbf{v}) + \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} \Delta\mathbf{v}_i^T\, \nabla^2_{ij} J(\mathbf{v})\, \Delta\mathbf{v}_j. \tag{20}$$

The use of the coupling K_{ij} gives a maximum bound on the value of the last term (see (12)). Thus, it is proposed to approximate the inter-agent coupling by simply estimating this term over successive iterations.

If only perturbations in the decisions of agents i and j are considered, then the cross derivative term can be estimated using

$$\Delta\mathbf{v}_i^T \nabla^2_{ij} J(\mathbf{v}) \Delta\mathbf{v}_j \approx \Delta\mathbf{v}_i^T\, \nabla_i J(\mathbf{v} + \Delta\mathbf{v}_j) - \Delta\mathbf{v}_i^T\, \nabla_i J(\mathbf{v}).$$

With some abuse of notation, the vector $\mathbf{v} + \Delta\mathbf{v}_j$ is assumed to represent $\mathbf{v} + [0, \ldots, 0, \Delta\mathbf{v}_j, 0, \ldots, 0]$. The inter-agent coupling K_{ij} can now be estimated using

$$K_{ij} \approx \frac{1}{\|\Delta\mathbf{v}_j\|} \left| \frac{\Delta\mathbf{v}_i^T}{\|\Delta\mathbf{v}_i\|}\, \nabla_i J(\mathbf{v} + \Delta\mathbf{v}_j) - \frac{\Delta\mathbf{v}_i^T}{\|\Delta\mathbf{v}_i\|}\, \nabla_i J(\mathbf{v}) \right|,$$

where the absolute value is used to maintain a positive estimate. By defining $\mathbf{e}_i = \frac{\Delta\mathbf{v}_i}{\|\Delta\mathbf{v}_i\|}$, the above equation can

be considered as the difference between the gradient of J in the direction $\mathbf{e}_i$ before and after a perturbation in agent j's decision.

By evaluating this at each iteration, and defining $\mathbf{e}_i$ and $\Delta\mathbf{v}_j$ to lie in the update directions of agents i and j respectively, the estimate will track the local curvature of the objective function in the vicinity of the actual team decision and in directions where the decisions are actively being refined. This is in contrast to using an absolute maximum over all positions and directions, as suggested in (12) and will result in a more appropriate value.

Thus, if agent i maintains the two most recent decisions communicated from agent j, ${}^i\mathbf{v}_j$ and ${}^i\mathbf{v}_j^{\text{old}}$, the coupling can be estimated by agent i at each iteration using

$$ {}^i\hat{K}_{ij} = \frac{1}{\|\Delta\mathbf{v}_j\|} \left| \mathbf{e}_i^T \, \nabla_i J({}^i\mathbf{v}) - \mathbf{e}_i^T \, \nabla_i J({}^i\mathbf{v}^{\text{old},j}) \right|, \qquad (21) $$

where $\Delta\mathbf{v}_j = {}^i\mathbf{v}_j - {}^i\mathbf{v}_j^{\text{old}}$, ${}^i\mathbf{v}^{\text{old},j} = \mathbf{v} - [0, \ldots, \Delta\mathbf{v}_j, \ldots, 0]$, and $\mathbf{e}_i$ is chosen to lie in the direction the local decision will be updated in, defined in (6) as $\mathbf{A}_i^{-1} \, \nabla_i J({}^i\mathbf{v})$.

This allows the inter-agent coupling to be evaluated locally be each agent using only gradient evaluations. Further more, only the inner product between the gradient and a unit vector is required, which can be significantly cheaper than a direct gradient evaluation.

A. Dynamic Communication Rates

Through the communication policy (19), the communication rate is determined by the inter-agent coupling, which will be expected to change throughout the optimisation procedure. As a consequence, the communication rates can be adjusted to automatically adapt to the changing requirements of the system as it traverses through the joint decision space $\mathcal{V}$.

B. Dynamic Step Size

In turn, the approximate coupling terms and current communication rates are used to calculate the step size γ_i of the current iteration

$$ \gamma_i(l) = \frac{\beta \, \hat{a}_i(l)}{{}^i\hat{K}_{ii}(l) + \sum_{j \neq i} {}^i\hat{K}_{ij}(l) \big(1 + \hat{B}_{ij} + \hat{B}_{ji}\big)}. \qquad (22) $$

The internal coupling ${}^i\hat{K}_{ii}$, representing the maximum curvature in the subspace of agent i's decision, can be approximated in a similar manner to ${}^i\hat{K}_{ij}$ or, if the Hessian submatrix $\nabla^2_{ii} J(\mathbf{v})$ is available, can be calculated directly using

$$ {}^i\hat{K}_{ii}(l) = \mathbf{e}_i^T \, \nabla^2_{ii} J({}^i\mathbf{v}) \, \mathbf{e}_i. \qquad (23) $$

Similarly, the scaling normalisation constant a_i is approximated with

$$ \hat{a}_i(l) = \mathbf{e}_i^T \, \mathbf{A}_i(l) \, \mathbf{e}_i. \qquad (24) $$

To accommodate for possible underestimation of the coupling and the inaccuracies in using $\hat{B}_{ij}$ with varying communication rates, the value of β should be selected appropriately. For problems with slowly varying Hessians and small communication delays, a large β can be used. However, for large delays or rapidly varying Hessians, a value closer to zero should be employed. For the results presented in this paper a value of $\beta = 1.5$ was used. The final distributed optimisation algorithm is given by Fig. 2.

Fig. 2. Local optimisation algorithm for agent i

Initialise decision vector ${}^i\mathbf{v} = [{}^i\mathbf{v}_1, \ldots, {}^i\mathbf{v}_n]$
for all $j \neq i$ **do**
  Initialise communication link to j
  Exchange computation rates R_i and R_j
  Determine communication delays D_{ij} and D_{ji}
end for
repeat
  $\mathbf{g}_i \Leftarrow \nabla_i J({}^i\mathbf{v})$ *(Local gradient)*
  $\mathbf{A}_i \Leftarrow \nabla^2_{ii} J({}^i\mathbf{v})$ or $\mathbf{I}$ *(Generate scaling matrix)*
  $\mathbf{d}_i \Leftarrow -\mathbf{A}_i^{-1} \, \mathbf{g}_i$ *(Update direction)*
  $\mathbf{e}_i \Leftarrow \mathbf{d}_i / \|\mathbf{d}_i\|$ *(Unit vector in update direction)*
  $\hat{a}_i \Leftarrow \mathbf{e}_i^T \mathbf{A}_i \, \mathbf{e}_i$ *(scaling normalisation)*
  $g_{\mathbf{e}_i} \Leftarrow \mathbf{e}_i^T \, \mathbf{g}_i$ *(Gradient in update direction)*
  ${}^i\hat{K}_{ii} \Leftarrow \mathbf{e}_i^T \, \nabla^2_{ii} J({}^i\mathbf{v}) \, \mathbf{e}_i$ *(Or via finite diff. approx.)*
  for all $j \neq i$ **do** *(Calculate coupling and delay terms†)*
    ${}^i\mathbf{v}^{\text{old},j} \Leftarrow {}^i\mathbf{v} - [0, \ldots, \Delta\mathbf{v}_j, \ldots, 0]$ *(Perturbed decision)*
    $g^{\text{old}}_{\mathbf{e}_i} \Leftarrow \mathbf{e}_i^T \, \nabla_i J({}^i\mathbf{v}^{\text{old},j})$ *(Perturbed gradient)*
    ${}^i\hat{K}_{ij} \Leftarrow \left| g_{\mathbf{e}_i} - g^{\text{old}}_{\mathbf{e}_i} \right| / \|\Delta\mathbf{v}_j\|$ *(Inter-agent coupling)*
    $\hat{B}_{ij} \Leftarrow R_i / C_{ij} + R_i \, D_{ij}$ *(Inter-agent delay terms)*
    $\hat{B}_{ji} \Leftarrow R_j / C_{ji} + R_j \, D_{ji}$ *(Requires comm. rates of j)*
  end for
  $\gamma_i \Leftarrow \dfrac{\beta \, \hat{a}_i}{{}^i\hat{K}_{ii} + \sum_{j \neq i} {}^i\hat{K}_{ij} \big(1 + \hat{B}_{ij} + \hat{B}_{ji}\big)}$ *(Step size)*
  ${}^i\mathbf{v}_i \Leftarrow \mathrm{Proj}^{\mathbf{A}_i}_{\mathcal{V}_i} \big({}^i\mathbf{v}_i + \gamma_i \mathbf{d}_i\big)$ *(Update local decision)*
  for all $j \neq i$ **do** *(Manage communications)*
    $C_{ij} \Leftarrow \eta \sqrt{{}^i\hat{K}_{ij}}$ *($\eta = R_i / \max_{j \neq i} \sqrt{{}^i\hat{K}_{ij}}$ or is determined by some bandwidth constraint)*
    if Req. to send msg to j **then** *(Determined from C_{ij})*
      Send $m_{ij} = {}^i\mathbf{v}_i$ to j
    end if
    if Msg $m_{ji} = {}^j\mathbf{v}_j$ received from j **then**
      $\Delta\mathbf{v}_j \Leftarrow {}^i\mathbf{v}_j - {}^j\mathbf{v}_j$ *(Determine change in external decision)*
      ${}^i\mathbf{v}_j \Leftarrow {}^j\mathbf{v}_j$ *(Update local copy)*
      Use time between messages to estimate C_{ji}
    end if
  end for
until Converged

† The inter-agent coupling ${}^i\hat{K}_{ij}$ can only be estimated after two messages have been received from j.

As noted, the inter-agent coupling terms ${}^i\hat{K}_{ij}$ are estimated using information from the two previous messages communicated by j. Thus, before this information has been received no estimate can be made and the term can be set to zero. For simplicity, the extra logic required to deal with this situation has been ignored.

IV. Multi-Vehicle Search

A typical scenario of interest is the search for a lost object or target by multiple vehicles, (e.g. maritime search for a capsized yacht at sea using multiple search aircraft). The search problem was first examined by Stone [14] and more recently by Bourgault *et al*. [15]–[17].

This paper follows the general formulation of Bourgault, which uses a receding horizon control method, whereby a open loop plan is generated by optimising the controls over a finite planning horizon. Only the initial portion of the plan is implemented, afterwhich a new plan is generated.

A. System Description

1) Lost Object: The object state is given by its 2D position $\mathbf{x}_o = [x_o, y_o]^T$ and is assumed to lie within the domain $\mathcal{X} \subseteq$

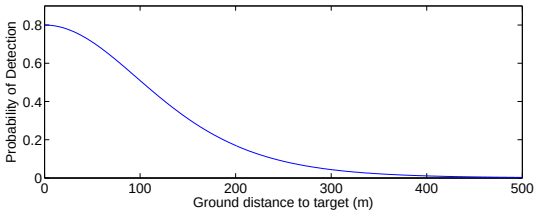

Fig. 3. Probability of detection as a function of ground distance to target from an aircraft travelling at a fixed altitude of 250m.

$\Re^2$. It is assumed that the object has slow dynamics compared to the motion of the vehicles and thus, can be approximated as stationary.

2) Search Vehicles: The state of the i^{th} vehicle is given by its 2D position and heading, $\mathbf{r}_i = [x_i, y_i, \theta_i] \in \mathcal{R}_i$ and travel at constant altitude h_i and velocity V_i. They are controlled from time t^k to $t^{k+1} = t^k + \Delta t$ by applying a fixed turn rate $\mathbf{u}_i^{k+1} \in \mathcal{U}_i \subseteq \Re$, where $\mathcal{U}_i$ models the control limits (and is thus a convex set). The resulting motion is assumed deterministic and denoted by $\mathbf{r}_i^{k+1} = f_i^k(\mathbf{r}_i^k, \mathbf{u}_i^k)$.

3) Observations: Each aircraft is equipped with a downward looking scanning radar. It is assumed at each time step the sensor either detects the position of the object accurately $\mathbf{z}_i^k = D$ and the mission is over, or no detection is made $\mathbf{z}_i^k = \overline{D}$.

The observations are modelled by the conditional probability distribution $P(\mathbf{z}_i^k|\mathbf{x}; \mathbf{r}_i^k)$. This determines the probability of receiving an observation $\mathbf{z}_i^k$ given the object position $\mathbf{x}$ and vehicles state $\mathbf{r}_i^k$. The modelled probability of detection adopted in this work is based on [15] and has the form

$$P(\mathbf{z}_i^k = D|\mathbf{x}; \mathbf{r}_i^k) = a_1 \frac{1}{d^4} e^{-a_1 d}, \tag{25}$$

where a_1 and a_2 are constants and d is the distance between the sensor and the object. The function used in the example is plotted in Fig. 3. The probability of not detecting the target is defined as $P(\mathbf{z}_i^k = \overline{D}|\mathbf{x}; \mathbf{r}_i^k) = 1 - P(\mathbf{z}_i^k = D|\mathbf{x}; \mathbf{r}_i^k)$. The combined or joint observation is denoted by $\mathbf{z}^k = [\mathbf{z}_1^k, \ldots, \mathbf{z}_p^k]$.

B. Recursive Bayes Filtering

Baysian filtering is used to maintain the probability density function (PDF) of the position of the object. This requires the search team be given some prior belief of the target location based on specific domain knowledge, such as the estimate of a distress call. Once this has been established, the belief at any later stage can be constructed recursively.

Consider the system at a given time step k. The team's state is given by $\mathbf{r}^k = [\mathbf{r}_1^k, \ldots, \mathbf{r}_p^k]$ and the PDF of $\mathbf{x}$, conditioned on all past observations and agent configurations, is $P(\mathbf{x}|\mathbf{Z}^k; \mathbf{R}^k)$, where $\mathbf{Z}^k = [\mathbf{z}^1, \ldots, \mathbf{z}^k]$ and $\mathbf{R}^k = [\mathbf{r}^1, \ldots, \mathbf{r}^k]$ and $\mathbf{Z}^0 = \mathbf{R}^0 = \emptyset$.

When a joint control action, $\mathbf{u}^k = [\mathbf{u}_1^k, \ldots, \mathbf{u}_p^k]$ is taken, the new state of the vehicles becomes $\mathbf{r}^{k+1}$, and a joint observation $\mathbf{z}^{k+1}$ is taken. Since the target is assumed stationary and assuming the observations are conditionally independent, the prior PDF can be updated directly using Bayes rule

$$P(\mathbf{x}|\mathbf{Z}^{k+1}; \mathbf{R}^k) = \frac{1}{\rho^{k+1}} P(\mathbf{x}|\mathbf{Z}^k; \mathbf{R}^k) \prod_{i=1}^{p} \Lambda_i^{k+1}(\mathbf{x}), \tag{26}$$

where $\Lambda_i^{k+1}(\mathbf{x}) \triangleq P(\mathbf{z}_i^{k+1}|\mathbf{x}; \mathbf{r}_i^{k+1})$ is the i^{th} vehicles likelihood function defined once the observation $\mathbf{z}_i^{k+1}$ and vehicle configuration $\mathbf{r}_i^{k+1}$ are known, and ρ^{k+1} is a normalisation constant. For the i^{th} vehicle to update its PDF, it must receive the likelihood function $\Lambda_j^{k+1}(\mathbf{x})$ from all other vehicles $j \neq i$. For simplicity, it is assumed each vehicle communicates the likelihood directly to every other vehicle.

C. Trajectory Control

The vehicles will be controlled using receding horizon control. Thus, at each time k an optimal planning problem is constructed over a fixed time horizon of N steps. The objective function is defined by considering the probability of the target not being detected over this horizon. For the system at time k, with vehicle states $\mathbf{r}_1^k, \ldots, \mathbf{r}_p^k$ and object PDF $P(\mathbf{x}|\mathbf{Z}^k; \mathbf{R}^k)$, the objective function for a given open loop control plan $\mathbf{u}^{k:k+N-1} = [\mathbf{u}^k, \ldots, \mathbf{u}^{k+N-1}]$ is

$$J^k(\mathbf{u}^{k:k+N-1}) = \int_{\mathcal{X}} P(\mathbf{x}|\mathbf{Z}^k; \mathbf{R}^k) \prod_{\ell=1}^{N} \prod_{i=1}^{p} P(\mathbf{z}_i^{k+\ell} = \bar{D}|\mathbf{x}; \mathbf{r}_i^{k+\ell}) d\mathbf{x}, \tag{27}$$

where $\mathbf{r}_i^{k+\ell}$ is calculated recursively from $\mathbf{r}_i^k$ and $\mathbf{u}_i^{k:k+N-1}$ using $\mathbf{r}_i^{k+\ell+1} = f_i^{k+\ell}(\mathbf{r}_i^{k+\ell}, \mathbf{u}_i^{k+\ell})$.

Assuming the time required to perform the minimisation of (27) is small compared to the dynamics of the system (and the time horizon $\Delta t N$), the optimal control problem can be solved using the algorithm in Fig. 2.

The objective function has a minimum value of zero and, assuming the prior PDF is smooth, the objective function is twice differeciable (since the motion and observation models are also smooth). However, it cannot be expected to be convex and thus only a local optimum can be expected.

D. Numerical Results

Due to space requirements only a single scenario is considered here. This consists of 5 vehicles, starting in two groups as shown in Fig. 4(a). Each vehicles planned trajectory is parameterised such that a single turn rate is held constant for 5 time steps of 1s duration. For a time horizon of 25s, this results in a decentralised optimisation problem (that is resolved every time step) involving 25 parameters distributed over the 5 vehicles.

Snapshots of the scenario are shown in Fig. 4(a) – (d), also displayed are the planned future trajectories (dashed). The inter-agent coupling structure, across the whole team, for each optimal planning problem is displayed in Fig. 4(e) – (h) for each optimisation problem represented in Fig. 4(a) – (d) respectively. Each matrix element represents the average of the coupling estimates generated by each agent, for the corresponding instance of the optimal planning problem, normalised against the maximum. At the start of the scenario, only the adjacent vehicles are coupled. While as the mission evolves, the coupling adapts to reflect the degree of overlap of the vehicles future observations. For example consider agent 3 in Fig. 4(b), from Fig. 4(f) agent 3 is strongly coupled to agent 2, which has an adjacent future trajectory, while slightly less coupled to agents 4 and 5 which have less of their future

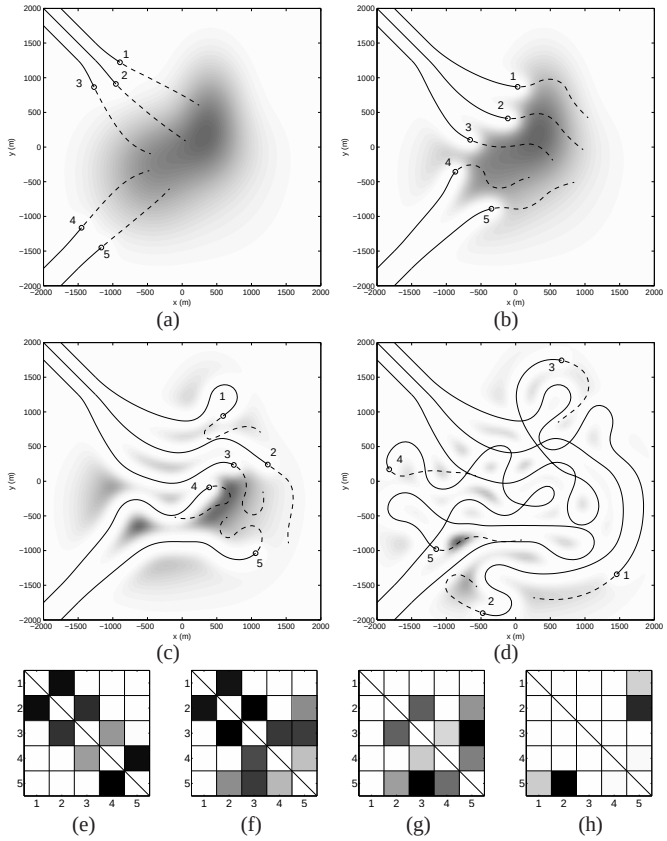

Fig. 4. (a) – (d) Snapshots at $k = 10$, 30, 60 and 150 respectively. The dashed lines represent the optimal future trajectory of each agent. The PDF of the target is represented by the shading, with dark (light) representing high (low) probability density. (e) – (h) The corresponding inter-agent coupling for each snapshot in (a) – (d). The i^{th} row of each matrix represents the average of ${}^{i}\hat{K}_{ij}$ throughout the optimisation procedure, scaled to the maximum throughout the team. Dark (light) represents high (low) coupling.

trajectories adjacent to agent 3's. Further more, agent 3 is not coupled to agent 1.

The coupling is used to determine how often to communicate to the rest of the agents in the team through the policy (19). It is surprising that this purely mathematical construction agrees with the intuitive notion that only vehicles that will observe similar regions are coupled and should communicate while generating joint plans.

V. Conclusion

This paper has presented a distributed asynchronous optimisation algorithm for multi-agent decision problems involving continuously varying decisions and twice differentiable objective functions. The algorithm allows each agent to incrementally refine their decision while intermittently receiving updates from the rest of the team. To guarantee convergence, a sufficient condition was presented which intuitively relates communication frequency, transmission delays and the amount of inter-agent coupling inherent within the system, to the rate at which each agent's local decision may be refined.

The inter-agent coupling is related to the cross derivatives of the objective function and requires considerable knowledge of the structure of the team objective function. To overcome this issue, an approximation method was introduced that allows each agent to calculate its coupling to each other agent using only local information. The estimate is based on a finite difference approximation of the cross derivative terms and is evaluated using communicated information from two subsequent iterations.

The algorithm was applied to the on-line control of a simulated multi-vehicle search problem. For this scenario, it was observed the inter-agent coupling is related to the potential overlap between the future observations and gives an intuitive requirement on which vehicles must communicate when generating future plans, and often results in a very sparse communication topology.

Acknowledgments

This work is partly supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government; and by CSIRO Industrial Physics.

References

[1] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[2] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. of the American Control Conf.*, 2002.

[3] W. Li and C. G. Cassandras, "A cooperative receding horizon controller for multivehicle uncertain environments," *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 242–257, 2006.

[4] G. M. Baudet, "Asynchronous iterative methods for multiprocessors," *Journal of the ACM*, vol. 25, no. 2, pp. 226–244, April 1978.

[5] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athens, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.

[7] G. Inalhan, D. Stipanovic, and C. Tomlin, "Decentralized optimization, with application to multiple aircraft coordination," in *Proc. IEEE Conf. on Decision and Control*, 2002.

[8] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: Application to formation flight," in *Proc. IEEE Conf. on Decision and Control*, 2004.

[9] D. P. Bertsekas and J. N. Tsitsiklis, "Some aspects of parallel and distributed iterative algorithms - A survey," *Automatica*, vol. 27, no. 1, pp. 3–21, 1991.

[10] P. Modi, W. Shen, M. Tambe, and M. Yokoo, "ADOPT: Asynchronous distributed constraint optimization with quality guarantees," *Artificial Intelligence*, vol. 161, no. 1-2, pp. 149–180, 2005.

[11] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[12] G. M. Mathews, H. Durrant-Whyte, and M. Prokopenko, "Asynchronous gradient-based optimisation for team decision making and control," *IEEE Transactions on Robotics*, Submitted, 2007.

[13] P. Tseng, "On the rate of convergence of a partially asynchronous gradient projection algorithm," *SIAM Journal on Optimization*, vol. 1, no. 4, pp. 603–619, 1991.

[14] L. D. Stone, *Theory of Optimal Search*. Academic Press, New York, 1975.

[15] F. Bourgault, A. Goktogan, T. Furukawa, and H. F. Durrant-Whyte, "Coordinated search for a lost target in a bayesian world," *Journal of Advanced Robotics*, vol. 18, no. 10, pp. 979–1000, 2004.

[16] F. Bourgault, G. Mathews, A. Brooks, and H. Durrant-Whyte, "An indoor experiment in decentralized coordinated search," in *The 9th International Symposium on Experimental Robotics*, 2004, pp. 407–416.

[17] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Decentralized bayesian negotiation for cooperative search," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004.