5

Decentralized Decision Making for Multiagent Systems

George Mathews, Hugh Durrant-Whyte, and Mikhail Prokopenko

5.1 Introduction

Decision making in large distributed multiagent systems is a difficult problem. In general, for an agent to make a good decision, it must consider the decisions of all the other agents in the system. This coupling among decision makers has two main causes: (i) the agents share a common objective function (e.g., in a team), or (ii) the agents share constraints (e.g., they must cooperate in sharing a finite resource).

The classical approach to this type of problem is to collect all the information from the agents in a single center and solve the resulting optimization problem [see, e.g., Furukawa et al. (2003)]. However, this centralized approach has two main difficulties:

- The required communication bandwidth grows at least linearly with the number of agents.
- The resulting optimization complexity is generally exponential in the number of agents.

Thus, for a sufficiently large number of agents this problem becomes impractical to solve in a centralized fashion.

However, these difficulties can be overcome by allowing the agents to cooperate or self-organize in solving this distributed decision problem. The main issue in this decentralized approach is identifying which agents need to communicate, what information should be sent, and how frequently.

This chapter approaches the multiagent collaboration problem using analytical techniques and requires that several assumption be made about the form of the problem: (i) the decisions of the individual agents are represented by elements of a continuous and finite-dimensional vector space; (ii) the agents are coupled via a shared objective function that is continuous and twice differentiable, and (iii) there are no interagent constraints.

With these assumptions, this chapter presents fundamental results on the structure of the decentralized optimal decision problem. A simple and intuitive decentralized negotiation algorithm is presented which enables multiple decision makers to propose and refine decisions to optimize a given team objective function. A convergence analysis of this procedure provides an intuitive relationship between the communication frequency, transmission delays, and the inherent interagent coupling in the system.

The algorithm is applied to the control of multiple mobile robots undertaking an information-gathering task. The specific scenario considered requires that the robots actively localize a group of objects. For this scenario the interagent coupling loosely relates to the amount of overlap between the information that two agents receive when undertaking their respective plans. This requires communications only between coupled agents and results in a more scalable system.

Section 5.2 defines the multiagent decision problem and introduces the decentralized optimization algorithm to solve it. This section also defines the interagent coupling metric and its relationship to the communication structure. Section 5.3 examines the full dependency between the rate an agent can refine its decision with the communication frequency, transmission delays, and the interagent coupling of the system. A decomposition of the objective function is introduced in Section 5.4 that is used to explicitly specify what each agent must communicate. An approximation technique is proposed to enable each agent to calculate the interagent coupling on-line. An overview of the decentralized decision-making or negotiation algorithm is given in Section 5.5. This summarizes exactly what each agent must know about the other agents and details the local procedure executed by each agent. Section 5.6 describes the general multiagent information-gathering control problem and formulates it as a decentralized sequential decision problem. This is specialized for an object localization problem in Section 5.7, with results given in Section 5.8. Section 5.9 provides a summary and directions for future work.

5.2 Asynchronous Decision Making

Consider a system of p agents at some specific instant in time; each agent i is in charge of a local decision variable $\mathbf{v}_i \in \mathcal{V}_i$. As stated in the Introduction, it is assumed that the set of feasible decisions for agent i is a Euclidean space of dimension n_i, i.e., $\mathcal{V}_i \equiv \Re^{n_i}$. This may be relaxed such that $\mathcal{V}_i$ is a convex subset of $\Re^{n_i}$, but for simplicity this case is ignored. The global decision vector, $\mathbf{v} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_p^T]^T$, is defined on the product space $\mathcal{V} = \mathcal{V}_1 \times \cdots \times \mathcal{V}_p = \Re^n$, where $n = n_1 + \cdots + n_p$.

The system as a whole is required to select the decisions such that a given objective function $J : \mathcal{V}_1 \times \cdots \times \mathcal{V}_p \to \Re$ is minimized. The objective function captures the goals of the system and will generally incorporate a model of how the agents interact with the environment using their sensors and actuators. The optimal decision problem is given by

$$\mathbf{v}^* = \arg\min_{\mathbf{v} \in \mathcal{V}} J(\mathbf{v}), \tag{5.1}$$

where $\mathbf{v}^*$ is the desired optimal global decision.

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

Under the convexity assumption, Eq. (5.1) is equivalent to requiring the gradient vector to vanish:

$$\nabla J(\mathbf{v}^*) = \mathbf{0}. \tag{5.2}$$

In terms of each agent's local decision, this can be written as

$$\nabla_i J(\mathbf{v}^*) = \mathbf{0} \quad \forall\, i, \tag{5.3}$$

where $\nabla_i J(\mathbf{v}) \in \Re^{n_i}$ represents the components of the gradient vector in the subspace $\mathcal{V}_i$. It is this optimality condition that is considered throughout this chapter.

5.2.1 Local Decision Refinement

The proposed solution method for the optimization problem allows each agent to submit an initial decision and then to incrementally refine this, while intermittently communicating these refinements to the rest of the system. The distributed nature of the problem requires that each agent execute and communicate asynchronously; thus the information it has about other agents may be outdated. This requires that each agent maintain a local copy of the team decision vector, which is given at a discrete time t for each agent i as

$$\begin{aligned} {}^i\mathbf{v}(t) &= \left[{}^i\mathbf{v}_1(t), \ldots, {}^i\mathbf{v}_p(t)\right] & (5.4)\\ &= \left[{}^1\mathbf{v}_1\big(\tau_{1i}(t)\big), \ldots, {}^p\mathbf{v}_p\big(\tau_{pi}(t)\big)\right]. & (5.5) \end{aligned}$$

In general a presuperscript represents a copy held by a specific agent, while a subscript represents a specific agent's decision [e.g., ${}^i\mathbf{v}_j(t)$ represents agent i's local copy of agent j's decision]. The variable $\tau_{ji}(t)$ in Eq. (5.5) represents the time agent i's local copy ${}^i\mathbf{v}_j(t)$ was generated by agent j and hence ${}^i\mathbf{v}_j(t) = {}^j\mathbf{v}_j\big(\tau_{ji}(t)\big)$. It is assumed that $\tau_{ii}(t) = t$, and thus agent i always has the latest copy of its decision variable.

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

To formalize the intuitive notion of *decision refinement*, a local update rule $f_i\colon \mathcal{V} \to \mathcal{V}_i$ will be defined for each agent that modifies its local decision ${}^i\mathbf{v}_i$, based on its copy of the global decision vector ${}^i\mathbf{v}$. To allow each agent to perform updates asynchronously a set of times T_U^i is associated with each agent i that represents when the agent computes a local update:

$${}^i\mathbf{v}_i(t+1) = \begin{cases} f_i\big({}^i\mathbf{v}(t)\big) & \text{if } t \in T_U^i \\ {}^i\mathbf{v}_i(t) & \text{else} \end{cases}. \tag{5.6}$$

For the update to be beneficial, it should decrease the value of the objective function. Thus, a steepest descent update is used:

$$f_i\big({}^i\mathbf{v}(t)\big) = {}^i\mathbf{v}_i(t) - \gamma_i\, \nabla_i J\big({}^i\mathbf{v}(t)\big), \tag{5.7}$$

where γ_i is a step size.

However, since only local and possibly out-of-date information is available, it is not trivial to guarantee that this will decrease the value of the objective function corresponding to the latest decisions from all agents. The rest of this section develops conditions that will guarantee this property and the overall convergence of the algorithm by providing limits on the step size γ_i.

5.2.2 Communication

Communication is initiated by agent i sending a message, at some time $t \in T_C^{ij}$, to agent j containing its latest decision ${}^i\mathbf{v}_i(t)$. After some communication delay $b_{ij}(t)$, agent j receives it and incorporates it into its local copy of the team decision vector. Thus, when the message is received ${}^j\mathbf{v}_i\big(t + b_{ij}(t)\big) = {}^i\mathbf{v}_i(t)$, and hence $\tau_{ij}\big(t + b_{ij}(t)\big) = t$. For each agent to obtain the decisions of every other agent, each agent must communicate with every other agent.

Assumption 2 (Bounded Delays) *There exist finite positive constants B_{ij} such that*

$$t - \tau_{ij}(t) \leq B_{ij} \quad \forall\, i, j, t. \tag{5.8}$$

Thus, the age of agent j's local copy of agent i's decision is bounded.

Informally, Assumption 2 can be relaxed such that B_{ij} represents the time difference, measured in numbers of updates computed by agent i, between ${}^i\mathbf{v}_i(t)$ and ${}^j\mathbf{v}_i(t)$.

If the agents compute updates and communicate at a fixed frequency, these can be approximated by knowing: (i) the number of iterations R_i computed by agent i per second, (ii) the number of communicated messages $C_{i\to j}$ from i to j per second, and (iii) the delivery delay $D_{i\to j}$ between agent i sending and j receiving a message in seconds, using

$$\hat{B}_{ij} = \frac{R_i}{C_{i\to j}} + R_i D_{i\to j}. \tag{5.9}$$

5.2.3 Interagent Coupling

For each pair of agents, the magnitude of interagent coupling is captured by a single scalar that represents the maximum curvature of the objective function in the subspace containing the decision variables of the two agents.

Assumption 3 (Coupling) *For every i and j, there exists a finite positive constant K_{ij}, such that*

$$\mathbf{x}_i^T\, \nabla_{ij}^2 J(\mathbf{v})\, \mathbf{x}_j \leq \|\mathbf{x}_i\|\, K_{ij}\, \|\mathbf{x}_j\| \tag{5.10}$$

for all $\mathbf{v} \in \mathcal{V}$ and all column vectors $\mathbf{x}_i \in \mathcal{V}_i$ and $\mathbf{x}_j \in \mathcal{V}_j$. The matrix $\nabla_{ij}^2 J(\mathbf{v})$ corresponds to the $n_i \times n_j$ submatrix of the Hessian of the objective function. The vector norms are defined as the Euclidean l^2 norm.

It is noted that the coupling constants are symmetric, and hence $K_{ij} = K_{ji}$.

If the actual Hessian is available, then the limit can be computed using

$$K_{ij} = \max_{\mathbf{v} \in \mathcal{V}} \left\| \nabla^2_{ij} J(\mathbf{v}) \right\|_M , \tag{5.11}$$

where $\|\cdot\|_M$ represents the induced matrix norm and is given by

$$\|A\|_M = \max_{\|x\|=1} \|Ax\| , \tag{5.12}$$

where x is a vector of appropriate dimension.

5.2.4 Asynchronous Convergence

The amount of interagent coupling and the magnitude of the communication delays play an important role in the amount each agent may refine its local decision. Intuitively, if the decisions of two agents are highly dependent, they should communicate more often while refining their decisions.

Equation (5.10) defined a metric K_{ij} measuring interagent coupling for a given multiagent decision problem. Similarly Eq. (5.8) encapsulated the effects of asynchronous execution and communication delays between two agents using B_{ij}. These are now used to provide an intuitive limit on the refinement step size γ_i introduced in Eq. (5.7).

Theorem 1 (Convergence). *Assumptions 1 to 3 provide sufficient conditions for the distributed asynchronous optimization algorithm defined by Eq. (5.7) to converge to the global optimum, defined by Eq. (5.3), for all $\gamma_i \in (0, \Gamma_i)$, where*

$$\Gamma_i = \frac{2}{K_{ii} + \sum_{j \neq i} K_{ij} \left(1 + B_{ij} + B_{ji}\right)} . \tag{5.13}$$

The convergence of this type of algorithm was first proved by Tsitsiklis et al. (1986).

This theorem provides a unified way of relating the inherent interagent 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

$$\gamma_i = \frac{\beta}{K_{ii} + \sum_{j \neq i} K_{ij} \left(1 + B_{ij} + B_{ji}\right)} \tag{5.14}$$

for some $\beta \in (0, 2)$.

5.3 Communication Structure

For the general problem under consideration, for each agent to receive the decisions of every other agent, there must exist a communication channel among all the agents in the system. Regardless of the implementation, the only relevant features of this communication network are the interagent communication frequency and transmission delays.

The communication frequency is directly controllable by the agents, whereas the transmission delays are determined by properties of the communication network and/or the nature of the physical communication medium.

For simplicity, this work assumes that the communication network is fully connected and that the transmission delays are fixed for any pair of agents. Although this may seem unrealistic it will be shown later that for most realistic scenarios each agent will only be coupled to agents in a local neighbourhood and hence will only be required to communicate to them. This assumption also allows the (higher-level) problem of network formation and routing to be ignored.

5.3.1 Communication Rate

The rate at which agents communicate has a significant impact on the convergence rate of the optimization process. Although a detailed analysis of the convergence rate is beyond the scope of this work, it is reasonable to assume that it is proportional to the step size γ_i (larger steps will generally allow the agents' decisions to be refined faster).

The step size for a given agent i is determined by Eq. (5.14), where the delay terms can be approximated by $\hat{B}_{ij}$, defined in Eq. (5.9). When these are combined, the step size obeys the relation

$$\frac{\beta}{\gamma_i} = K_{ii} + \sum_{j \neq i} K_{ij} \left(1 + \frac{R_i}{C_{i \to j}} + R_i D_{i \to j} + \frac{R_j}{C_{j \to i}} + R_j D_{j \to i} \right). \tag{5.15}$$

If it is assumed that the computation rates (R_i and R_j) and interagent communication delays ($D_{j \to i}$ and $D_{i \to j}$) are fixed, this becomes

$$\frac{\beta}{\gamma_i} = K_{ii} + \sum_{j \neq i} K_{ij} \left(W_{ij} + \frac{R_i}{C_{i \to j}} + \frac{R_j}{C_{j \to i}} \right), \tag{5.16}$$

where $W_{ij} = 1 + R_i D_{i \to j} + R_j D_{j \to i}$ and is a constant. Thus, from Eq. (5.16), the step size γ_i is maximized when all the communication rates are also maximized; i.e. every agent communicates to every other agent after every local iteration.

However, this policy is impractical for large systems containing many agents. Potentially this can be overcome by allowing each pair of agents to communicate at a rate proportional to the coupling, i.e.,

$$C_{i \to j} = \eta_i K_{ij} \tag{5.17}$$

for some constant η_i. However, this will also be impractical for large systems since the step size will be directly related to the number of agents. This can be demonstrated by considering all agents to have a fixed computation rate (i.e., $R_i = R$ and $\eta_i = \eta, \forall\, i$) and substituting Eq. (5.17) into Eq. (5.16):

$$\frac{\beta}{\gamma_i} = K_{ii} + \sum_{j \neq i} K_{ij} W_{ij} + 2(p-1)\frac{R}{\eta}. \tag{5.18}$$

Thus, for large p, the last term will dominate causing the step size γ_i to approach 0 regardless of how small the actual interagent coupling may be. Hence, a communication rate proportional to the coupling is generally too low.

To overcome this it is proposed to set the communication rate proportional to the *square root* of the coupling:

$$C_{i \to j} = \eta_i \sqrt{K_{ij}}, \tag{5.19}$$

which represents a compromise between fast convergence, requiring a high communication rate, and the practical requirements of a slow communication rate. The step size γ_i, and hence the possible convergence rate, is now only dependent on the coupling, which is in turn determined by the inherent complexity of the problem. This can be demonstrated by substituting Eq. (5.19) into Eq. (5.16):

$$\frac{\beta}{\gamma_i} = K_{ii} + \sum_{j \neq i} \Big(K_{ij} W_{ij} + \sqrt{K_{ij}} \big(\eta_i R_i + \eta_j R_j \big) \Big). \tag{5.20}$$

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.

5.4 Modularization

Until now it has been implicitly assumed that each agent has a full representation of the system's objective function. In general this requires the state and sensor/actuator models of all agents. This may not be a problem for a small number of agents, but poses a significant issue for a large distributed system.

Ideally, each agent should only require a model of itself and relatively little knowledge of the other agents. This issue has been examined by Mathews et al. (2006), who identified a composable or partially separable form of the objective function that enables this type of modularization.

5.4.1 Partial Separability

The partially separable objective function allows the effects or impacts of each agent to be separated and evaluated independently. The objective function then becomes a function of the composition of these individual impacts.

Definition 1 (Partial Separability). *A partially separable*[1] *system has an objective function of the form*

$$J(\mathbf{v}) = \psi\big(\Upsilon_1(\mathbf{v}_1) * \Upsilon_2(\mathbf{v}_2) * \cdots * \Upsilon_p(\mathbf{v}_p)\big), \tag{5.21}$$

where $\Upsilon_i : \mathcal{V}_i \to \mathfrak{I}$ *is the* i^{th} *agent's* impact function *and maps a decision to an element of an* impact space $\mathfrak{I}$. *An element* $\alpha \in \mathfrak{I}$ *of this space will be referred to as an* impact. *The* composition operator $* : \mathfrak{I} \times \mathfrak{I} \to \mathfrak{I}$ *allows impacts to be summarized without losing any task-relevant information. The* generalized objective function $\psi : \mathfrak{I} \to \Re$ *maps a combined team impact to a scalar cost.*

[1] The function becomes fully separable if $\mathfrak{I} \equiv \Re$, $* \equiv +$ and ψ is linear.

An agent's impact function is an abstraction of its state and sensor/actuator models and maps a decision onto a task-specific impact space. It is assumed that each agent i only has knowledge of its own impact function Υ_i and thus requires the impacts $\alpha_j = \Upsilon_j(\mathbf{v}_j)$ from every other agent $j \neq i$ for the objective function to be evaluated. Thus, instead of each agent maintaining a local copy of every other agent's decision vector $\mathbf{v}_j$, it simply maintains their impact α_j.

This definition allows one to abstract away state and sensor/actuator models of the other agents and defines a common language of impacts that the agents use to communicate. For simplicity, the cumulative composition operator $\bigodot$ will be used, such that Eq. (5.21) can be written as

$$J(\mathbf{v}) = \psi\Big(\bigodot_{i=1}^{p} \Upsilon_i(\mathbf{v}_i)\Big). \tag{5.22}$$

Example: Collision Avoidance

To provide an example of an impact, consider a multiagent path-planning scenario with a separation requirement. For this task the objective function will be dependent on the separation between the agents, which in turn requires the individual paths (possibly represented by a fixed number of points) from all the agents. In this case each agent abstracts away its motion model and corresponding control inputs. Thus, an agent's path can be considered as its impact, and the composition operator simply collects the paths. The generalized objective function is used to calculate the associated cost of these specific paths.

It is noted that for this example the size of the impact space is proportional to the number of agents (the number of paths is the same as the number of agents). This differs from the example presented in Section 5.7, which has a composition operator given by addition. For this case the size of the impact space is independent of the number of agents (the sum of many numbers is still a number).

5.4.2 Modular Decision Refinement

With the use of the partially separable form of the objective function, the local decision refinement process, presented in Section 5.2.1, can be modified such that each agent i is only required to maintain a local copy of the impact of every other agent $\{{}^i\alpha_j(t) : \forall\, j \neq i\}$. This information is updated via messages from the other agents in the system and may contain delayed information (according to Assumption 2).

From this, the local decision refinement equations for agent i [corresponding to Eq. (5.7)] becomes

$$f_i\big({}^i\mathbf{v}(t)\big) = {}^i\mathbf{v}_i(t) - \gamma_i\, \mathbf{s}_i(t), \tag{5.23}$$

where $\mathbf{s}_i(t)$ is the local steepest descent direction $\nabla_i J\big({}^i\mathbf{v}(t)\big)$ and is given by

$$\mathbf{s}_i(t) = \nabla_i \psi\Big(\Upsilon_i\big({}^i\mathbf{v}_i(t)\big) * \bigodot_{j\neq i} {}^i\alpha_j(t)\Big). \tag{5.24}$$

Calculation of the step size γ_i requires the coupling terms K_{ij}. If the system is static, e.g, for a monitoring and control system for a distributed processing plant, these can be calculated in advance during the design phase by solving Eq. (5.11). However, for the majority of multiagent systems this will not be the case and the coupling terms must be calculated on-line.

5.4.3 On-line Coupling Estimation

Since no single agent has an explicit representation of the full objective function, it is not possible for any agent to solve Eq. (5.11) for the coupling terms. Rather, these terms are approximated using a finite difference evaluated locally by each agent from two successive iterations and communications.

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

$$J(\mathbf{v} + \Delta\mathbf{v}) \approx J(\mathbf{v}) + \sum_{i=1}^{p} \Delta\mathbf{v}_i^T \, \nabla_i J(\mathbf{v}) + \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} \Delta\mathbf{v}_i^T \, \nabla_{ij}^2 J(\mathbf{v}) \, \Delta\mathbf{v}_j. \quad (5.25)$$

The use of the coupling K_{ij} gives a maximum bound on the value of the last term [see Eq. (5.10)]. It is proposed to approximate the 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 a backwards difference approximation,

$$\Delta\mathbf{v}_i^T \nabla_{ij}^2 J(\mathbf{v}) \Delta\mathbf{v}_j \approx J(\mathbf{v}^{aa}) + J(\mathbf{v}^{bb}) - J(\mathbf{v}^{ba}) - J(\mathbf{v}^{ab}),$$

where $\mathbf{v}^{aa} = \mathbf{v}$, $\mathbf{v}^{ba} = \mathbf{v} - \Delta\mathbf{v}_i$, $\mathbf{v}^{ab} = \mathbf{v} - \Delta\mathbf{v}_j$, and $\mathbf{v}^{bb} = \mathbf{v} - \Delta\mathbf{v}_i - \Delta\mathbf{v}_j$; note that the increments are assumed to be added on to the correct components in the team decision vector.

For a system with a partially separable objective function, the i^{th} agent, at iteration t, can estimate its coupling to any other agent j using the local decisions ${}^i\mathbf{v}_i^b$ and ${}^i\mathbf{v}_i^a$ from the two previous iterations, with corresponding impacts ${}^i\alpha_i^b = \Upsilon_i\big({}^i\mathbf{v}_i^b\big)$ and ${}^i\alpha_i^a = \Upsilon_i\big({}^i\mathbf{v}_i^a\big)$, and a decision increment with norm $d_i = \big\|{}^i\mathbf{v}_i^a - {}^i\mathbf{v}_i^b\big\|$. Also required are the previous two impacts ${}^i\alpha_j^b$ and ${}^i\alpha_j^a$ communicated from agent j with a corresponding decision increment with norm d_j using

$${}^i\hat{K}_{ij}(t) = \left| \frac{\psi({}^i\alpha_T^{aa}) + \psi({}^i\alpha_T^{bb}) - \psi({}^i\alpha_T^{ba}) - \psi({}^i\alpha_T^{ab})}{d_i d_j} \right|, \quad (5.26)$$

where

$${}^i\alpha_T^{aa} = {}^i\alpha_{\bar{i}\bar{j}} * {}^i\alpha_i^a * {}^i\alpha_j^a, \qquad {}^i\alpha_T^{bb} = {}^i\alpha_{\bar{i}\bar{j}} * {}^i\alpha_i^b * {}^i\alpha_j^b,$$
$${}^i\alpha_T^{ba} = {}^i\alpha_{\bar{i}\bar{j}} * {}^i\alpha_i^b * {}^i\alpha_j^a, \qquad {}^i\alpha_T^{ab} = {}^i\alpha_{\bar{i}\bar{j}} * {}^i\alpha_i^a * {}^i\alpha_j^b$$

and ${}^{i}\alpha_{\bar{\imath}\bar{\jmath}}$ is the combined impact of the rest of the team:

$$ {}^{i}\alpha_{\bar{\imath}\bar{\jmath}} = \bigodot_{q \neq i,j} {}^{i}\alpha_q^a. $$

This approximation requires that each agent communicate its local impact ${}^{i}\alpha_i \in \mathfrak{I}$ and the amount its decision has changed since the last communication $d_i \in \Re^+$. It is also noted that the coupling can only be approximated after two messages have been received.

This method allows the coupling estimates ${}^{i}\hat{K}_{ij}(t)$ to track the curvature of the objective function in the vicinity of the actual team decision vector and in the directions in which the decisions are actively being refined. This is in contrast to using an absolute maximum over all positions and directions, as suggested in Eq. (5.10).

This approximation is used to calculate the step size γ_i of the subsequent iteration:

$$ \gamma_i(t) = \frac{\beta}{{}^{i}\hat{K}_{ii}(t) + \sum_{j \neq i} {}^{i}\hat{K}_{ij}(t)\big(1 + \hat{B}_{ij} + \hat{B}_{ji}\big)}. \tag{5.27} $$

The term ${}^{i}\hat{K}_{ii}(t)$, representing the maximum curvature in the subspace of agent i's decision, is approximated directly by agent i with

$$ {}^{i}\hat{K}_{ii}(t) = \left\| \nabla_{ii}^2 \psi\big(\Upsilon_i({}^{i}\mathbf{v}_i^a) * \bigodot_{j \neq i} {}^{i}\alpha_j^a\big) \right\|_M. $$

Or, if the Hessian submatrix cannot be calculated directly, ${}^{i}\hat{K}_{ii}$ can be calculated using a finite difference approach in a manner similar that for ${}^{i}\hat{K}_{ij}$.

The value of β can be large for problems with slowly varying Hessians, small communication delays, and high communication rates. However, for large delays or rapidly varying Hessians, a value closer to zero should be employed. For the results presented in this chapter a value of $\beta = 1$ was used.

5.4.4 Dynamic Communication

Equation (5.26) allows each agent to approximate the interagent coupling at each iteration t of the algorithm. This, in turn, can be used to calculate an appropriate communication rate using Eq. (5.19). Although this violates the assumption of a fixed communication rate, which is made when approximating B_{ij} in Eq. (5.9), it will be approximately true for the iterations around t.

This allows the interagent communication rate to dynamically vary throughout the operation of the algorithm in response to changing interagent coupling.

5.5 Algorithmic Details

A brief illustration of the structure of the final decentralized algorithm is given in Fig. 5.1. Table 5.1 lists all the functions and variables an agent must maintain, while the full details of the local algorithm executed by each agent is given by Algorithm 5.1.

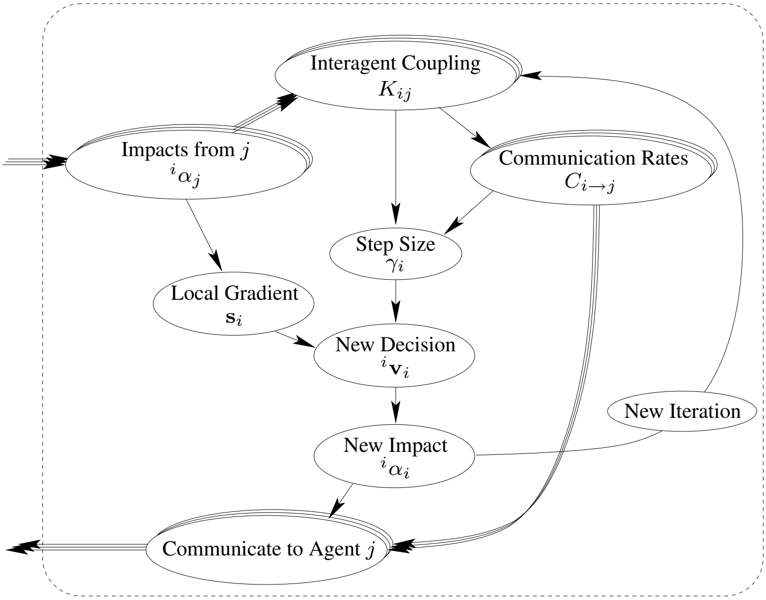


Fig. 5.1. General structure of the local algorithm executed by the i^{th} agent. For full details see Algorithm 5.1.

Table 5.1. Functions and variables maintained by the i^{th} agent.

Functions	Description
$\Upsilon_i : \mathcal{V}_i \to \mathfrak{I}$	Local impact function (abstraction of sensor and actuator models)
$* : \mathfrak{I} \times \mathfrak{I} \to \mathfrak{I}$	Impact composition operator
$\psi : \mathfrak{I} \to \Re$	Generalized objective function
Variables	**Description**
${}^{i}\mathbf{v}_i^a \in \mathcal{V}_i$	Local decision
$d_i \in \Re^+$	Distance local decision moved during last iteration
$\{{}^{i}\mathbf{v}_i^{\to j} \in \mathcal{V}_i : \forall\, j \neq i\}$	Set of previous local decisions, corresponding to communication events to agent j
${}^{i}\alpha_i^a \in \mathfrak{I}$	Local impact
${}^{i}\alpha_i^b \in \mathfrak{I}$	Local impact from previous iteration
$\{{}^{i}\alpha_j^a \in \mathfrak{I} : \forall\, j \neq i\}$	Set of impacts from other agents
$\{{}^{i}\alpha_j^b \in \mathfrak{I} : \forall\, j \neq i\}$	Set of previous impacts from other agents
$\{d_j \in \Re^+ : \forall\, j \neq i\}$	Set of distances other agents decisions moved between last two communications
$\{R_j \in \Re^+ : \forall\, j\}$	Set of computation rates
$\{C_{i\to j} \in \Re^+ : \forall\, j \neq i\}$, $\{C_{j\to i} \in \Re^+ : \forall\, j \neq i\}$	Sets of communication rates
$\{D_{i\to j} \in \Re^+ : \forall\, j \neq i\}$, $\{D_{j\to i} \in \Re^+ : \forall\, j \neq i\}$	Sets of transmission delays

Algorithm 5.1: Local algorithm run by the i^{th} agent.

for all $j \neq i$ **do**
  Initialise communication link to j
  Determine communication delays $D_{i\to j}$ and $D_{j\to i}$
  Exchange computation rates R_i and R_j
  $\text{NumMsg}_j \Leftarrow 0$
end for
Initialise decision ${}^i\mathbf{v}_i^a$ *(e.g. using* ${}^i\mathbf{v}_i^a \Leftarrow \arg\min_{\mathbf{v}_i \in \mathcal{V}_i} \psi\big(\Upsilon_i(\mathbf{v}_i)\big)$*)*
repeat
  for all $j \neq i$ **do** *(Estimate interagent coupling)*
    if $\text{NumMsg}_j < 2$ **then**
      ${}^i\hat{K}_{ij} \Leftarrow 0$
      $\hat{B}_{ij} \Leftarrow 0$
      $\hat{B}_{ji} \Leftarrow 0$
    else
      Evaluate ${}^i\hat{K}_{ij}$ using Eq. (5.26)
      $\hat{B}_{ij} \Leftarrow R_i/C_{i\to j} + R_i D_{i\to j}$
      $\hat{B}_{ji} \Leftarrow R_j/C_{j\to i} + R_j D_{j\to i}$
    end if
  end for
  ${}^i\hat{K}_{ii} \Leftarrow \left\| \nabla_{ii}^2 \psi\big(\Upsilon_i({}^i\mathbf{v}_i^a) * \bigodot_{j\neq i} {}^i\alpha_j^a\big) \right\|_M$
  $\gamma_i \Leftarrow \dfrac{\beta}{{}^i\hat{K}_{ii} + \sum_{j\neq i} {}^i\hat{K}_{ij}(1 + \hat{B}_{ij} + \hat{B}_{ji})}$ *(Evaluate step size)*
  $\mathbf{s}_i \Leftarrow -\nabla_i \psi\big(\Upsilon_i({}^i\mathbf{v}_i^a) * \bigodot_{j\neq i} {}^i\alpha_j^a\big)$ *(Evaluate update direction)*
  ${}^i\mathbf{v}_i^a \Leftarrow {}^i\mathbf{v}_i^a + \gamma_i \mathbf{s}_i$ *(Update local decision)*
  $d_i \Leftarrow \|\gamma_i \mathbf{s}_i\|$ *(Save distance decision moved)*
  ${}^i\alpha_i^b \Leftarrow {}^i\alpha_i^a$ *(Save previous local impact)*
  ${}^i\alpha_i^a = \Upsilon_i({}^i\mathbf{v}_i^a)$ *(Evaluate new local impact)*
  for all $j \neq i$ **do** *(Manage communications)*
    $C_{i\to j} \Leftarrow \eta\sqrt{{}^i\hat{K}_{ij}}$ *(*$\eta = R_i / \max_{j\neq i} \sqrt{{}^i\hat{K}_{ij}}$ *or is determined by some bandwidth constraint)*
    if Required to send message to j **then** *(Determined from* $C_{i\to j}$*)*
      Send $m_{ij} = \{{}^i\alpha_i^a, \left\|{}^i\mathbf{v}_i^a - {}^i\mathbf{v}_i^{\to j}\right\|, C_{i\to j}\}$ to j
      ${}^i\mathbf{v}_i^{\to j} \Leftarrow {}^i\mathbf{v}_i^a$ *(Save decision for use in next message)*
    end if
    if Message $m_{ji} = \{{}^j\alpha_j^m, d_j^m, C_{j\to i}^m\}$ received from j **then**
      ${}^i\alpha_j^b \Leftarrow {}^i\alpha_j^a$ *(Save previous impact)*
      ${}^i\alpha_j^a \Leftarrow {}^j\alpha_j^m$ *(Update current impact)*
      $d_j \Leftarrow d_j^m$ *(Update the distance the decision moved)*
      $C_{j\to i} \Leftarrow C_{j\to i}^m$ *(Update the communication rate)*
      $\text{NumMsg}_j \Leftarrow \text{NumMsg}_j + 1$
    end if
  end for
until Converged

5.5.1 Convergence

Since this algorithm is based on an approximation of Theorem 1, it is no longer theoretically guaranteed to converge, but experience has shown that it still converges for a wide range of problems. This is because the theorem only presents a sufficient condition for convergence and is in general overly pessimistic.

5.5.2 Termination

If the algorithm converges, it will reach the optimum set of decision only in the limit of infinite iterations. However, due to the gradient descent property of the algorithm, each iteration will, on average, reduce the system cost. This ensures that the quality of the decisions will continue to improve over time. Thus, for time-critical applications it is reasonable for each agent to simply terminate the algorithm and use the final decision generated.

5.6 Active Information Gathering

The application considered in this work consists of multiple mobile robots undertaking a reconnaissance or information-gathering task. This type of scenario requires the agents to actively gather information on a particular external random variable $\mathbf{x} \in \mathcal{X}$. In general this may include the positions and identities of stationary or mobile objects, terrain properties of a given region, or even surface information of a remote planet. In Section 5.7 this will be specialized for the active localization of a group of objects.

5.6.1 Agents

The mobile robotic agents are modelled as discrete time dynamical systems, with the i^{th} agent's state given by $\mathbf{s}_i \in \mathcal{S}_i$. The agent is controlled from discrete time $k-1$ to k by applying a particular control input $\mathbf{u}_i^k \in \mathcal{U}_i$. In general this causes the agent's state to change according to the probabilistic discrete time Markov motion model $P(\mathbf{s}_i^k|\mathbf{s}_i^{k-1}, \mathbf{u}_i^k)$. However, for simplicity it is assumed that the agent's motion is known with precision, i.e., $\mathbf{s}_i^k = \mathbf{f}_i(\mathbf{s}_i^{k-1}, \mathbf{u}_i^k)$. The joint system state and transition model is given by $\mathbf{s}^k = \mathbf{f}(\mathbf{s}^{k-1}, \mathbf{u}^k) = \{\mathbf{f}_1(\mathbf{s}_1^{k-1}, \mathbf{u}_1^k), \ldots, \mathbf{f}_p(\mathbf{s}_p^{k-1}, \mathbf{u}_p^k)\} = \{\mathbf{s}_1^k, \ldots, \mathbf{s}_p^k\}$.

The observations made by the i^{th} agent are modelled by the conditional density $P(\mathbf{z}_i^k|\mathbf{x}^k; \mathbf{s}_i^k)$, which describes the probability of obtaining a particular observation $\mathbf{z}_i^k$ given the external state $\mathbf{x}^k$ and agents state $\mathbf{s}_i^k$. The notation $\mathbf{z}^k$ denotes the set of observations from all the agents at time step k, i.e., $\mathbf{z}^k = \{\mathbf{z}_1^k, \ldots, \mathbf{z}_p^k\} \in \mathcal{Z} = \prod_{i=1}^p \mathcal{Z}_i$. With the assumption that the observations are conditionally independent given the states $\mathbf{x}^k$ and $\mathbf{s}_i^k$, the combined sensor model can be written as $P(\mathbf{z}^k|\mathbf{x}^k; \mathbf{s}^k) = \prod_{i=1}^p P(\mathbf{z}_i^k|\mathbf{x}^k; \mathbf{s}_i^k)$.

The agents are to be controlled such that the combined observations they receive produce the most informative (or least uncertain) belief about $\mathbf{x}$. To accomplish this the distribution's entropy will be used as a measure of its associated uncertainty. The entropy of a distribution $P(\mathbf{x})$ is the negative of the expectation of its logarithm:

$$H_{P(\mathbf{x})} = -\mathrm{E}_{\mathbf{x}}[\log P(\mathbf{x})]. \tag{5.28}$$

This can be used for continuous variables, where $P(\mathbf{x})$ is the probability density function, or for discrete variables, where $P(\mathbf{x})$ is the probability distribution function. For a detailed description of the properties of this metric see Cover and Thomas (1991).

5.6.2 Bayesian Filtering

This section details the process of maintaining an accurate belief (as a probability) about the state $\mathbf{x}$. The Bayesian approach requires that the system be given some prior belief; if nothing is known this may simply be a noninformative uniform prior. Once this has been established, the belief at any later stage can be constructed recursively. To avoid potential confusion, instantiated variables (variables assuming a specific value) will be denoted using a tilde, e.g., $P(\tilde{\mathbf{x}}) \equiv P(\mathbf{x} = \tilde{\mathbf{x}})$.

Consider the system at a given time step k. The system's state is given by $\tilde{\mathbf{s}}^k$ and the belief about $\mathbf{x}^k$, conditioned on all past observations and agent configurations, is $P(\mathbf{x}^k|\tilde{\mathbf{Z}}^k;\tilde{\mathbf{S}}^k)$, where $\tilde{\mathbf{Z}}^k = \{\tilde{\mathbf{z}}^1,\ldots,\tilde{\mathbf{z}}^k\}$ and $\tilde{\mathbf{S}}^k = \{\tilde{\mathbf{s}}^1,\ldots,\tilde{\mathbf{s}}^k\}$ and $\tilde{\mathbf{Z}}^0 = \tilde{\mathbf{S}}^0 = \{\emptyset\}$.

When a joint control action, $\tilde{\mathbf{u}}^{k+1}$ is taken, the new state of the agents becomes $\tilde{\mathbf{s}}^{k+1} = \mathbf{f}(\tilde{\mathbf{s}}^k, \tilde{\mathbf{u}}^{k+1})$, and an observation $\tilde{\mathbf{z}}^{k+1}$ is taken. To update the belief about $\mathbf{x}^{k+1}$, it must first be predicted forwards in time using the Chapman-Kolmogorov equation:

$$P(\mathbf{x}^{k+1}|\tilde{\mathbf{Z}}^k;\tilde{\mathbf{S}}^k) = \int_{\mathcal{X}} P(\mathbf{x}^{k+1}|\mathbf{x}^k)P(\mathbf{x}^k|\tilde{\mathbf{Z}}^k;\tilde{\mathbf{S}}^k)d\mathbf{x}^k. \tag{5.29}$$

The belief can now be updated using Bayes rule:

$$P(\mathbf{x}^{k+1}|\tilde{\mathbf{Z}}^{k+1};\tilde{\mathbf{S}}^{k+1}) = \frac{1}{\mu}P(\mathbf{x}^{k+1}|\tilde{\mathbf{Z}}^k;\tilde{\mathbf{S}}^k)\prod_{i=1}^{p} P(\tilde{\mathbf{z}}_i^{k+1}|\mathbf{x}^{k+1};\tilde{\mathbf{s}}_i^{k+1}), \tag{5.30}$$

where $\tilde{\mathbf{z}}^{k+1} = \{\tilde{\mathbf{z}}_1^{k+1},\ldots,\tilde{\mathbf{z}}_p^{k+1}\}$ are the actual observations taken by the agents. The term $P(\tilde{\mathbf{z}}_i^{k+1}|\mathbf{x}^{k+1},\tilde{\mathbf{s}}_i^{k+1})$ is the i^{th} agent's observation model evaluated at the actual observation and agent configuration, resulting in a likelihood over $\mathbf{x}^{k+1}$. The normalization constant μ is given by

$$\begin{aligned}\mu &= P(\tilde{\mathbf{z}}^{k+1}|\tilde{\mathbf{Z}}^k;\tilde{\mathbf{S}}^{k+1}) \\ &= \int_{\mathcal{X}} P(\mathbf{x}^{k+1}|\tilde{\mathbf{Z}}^k;\tilde{\mathbf{S}}^k)\prod_{i=1}^{p} P(\tilde{\mathbf{z}}_i^{k+1}|\mathbf{x}^{k+1};\tilde{\mathbf{s}}_i^{k+1})d\mathbf{x}^{k+1}. \end{aligned} \tag{5.31}$$

For all agents to maintain this belief, each agent must communicate the observation likelihood function $\lambda(\mathbf{x}^{k+1}) = P(\tilde{\mathbf{z}}_i^{k+1}|\mathbf{x}^{k+1},\tilde{\mathbf{s}}_i^{k+1})$ after each observation is made. The field of decentralized data fusion examines efficient ways for this to be communicated around a sensor network (Manyika and Durrant-Whyte 1994; Liggins et al.

1997); however for this work it is assumed that each agent simply communicates it to every other agent.

The key problem in this process is deciding on the system's control inputs $\mathbf{u}^k$ such that the uncertainty in the state $\mathbf{x}$ is minimized.

5.6.3 Control Parametrization

The objective of the system is to minimize its uncertainty in its joint belief about $\mathbf{x}$. There are many ways to formally define this control problem, the best being a discounted infinite horizon dynamic programming problem (Bertsekas 2005). However, for any relatively complex scenario this becomes intractable and approximate techniques must be used.

Thus, a finite look ahead will be considered and an open loop control policy for each agent developed. To accommodate feedback, a rolling time horizon will be employed. This requires that the control policies be recomputed at short intervals to keep the look ahead approximately constant and allows the system to adapt to changes.

The control policy will be parametrized by a piecewise constant function, defined over N equal time partitions of M time steps each (Goh and Lee 1988). This results in a look ahead of NM time steps. Thus, the open loop control policy for a time interval $[k+1, k+NM]$ can be specified with the parameters $\mathbf{v}_i^k = \{\mathbf{v}_i^k(1), \ldots, \mathbf{v}_i^k(N)\} \in \mathcal{V}_i = (\mathcal{U}_i)^N$ with actual controls given at each time step $k+q$ by $\mathbf{u}_i^{k+q} = \mathbf{v}_i^k\left(\lceil \frac{q}{M} \rceil\right)$, where $q \in \{1, \ldots, NM\}$ and $\lceil\cdot\rceil$ represents the roundup operator.

5.6.4 Objective Function

For a given time k, the utility of a future control policy $\mathbf{v}^k = \{\mathbf{v}_1^k, \ldots, \mathbf{v}_p^k\} \in \mathcal{V} = \mathcal{V}_1 \times \cdots \times \mathcal{V}_p$ and observation series $\mathbf{z}^{k+1:k+NM} = \{\mathbf{z}^{k+1}, \ldots, \mathbf{z}^{k+NM}\}$ is proportional to the amount of uncertainty in the resulting posterior belief at time $k+NM$. Actions and observations that produce a smaller uncertainty in the posterior belief are favoured over others. Thus, a suitable cost function may be the posterior entropy:

$$C^k(\mathbf{z}^{k+1:k+NM}, \mathbf{v}^k) = H_{P(\mathbf{x}^{k+NM}|\mathbf{z}^{k+1:k+NM}, \mathbf{s}^{k+1:k+NM}(\mathbf{v}^k), \tilde{\mathbf{Z}}^k, \tilde{\mathbf{S}}^k)}$$
$$= -\mathrm{E}_{\mathbf{x}^{k+NM}}\left[\log P(\mathbf{x}^{k+NM}|\mathbf{z}^{k+1:k+NM}, \mathbf{s}^{k+1:k+NM}(\mathbf{v}^k), \tilde{\mathbf{Z}}^k, \tilde{\mathbf{S}}^k)\right]. \quad (5.32)$$

However, the actual observations made will not be known in advance. Thus, an expectation over all possible future observations must be performed, resulting in the expected team cost or objective function:

$$J^k(\mathbf{v}^k) = \mathrm{E}_{\mathbf{z}^{k+1:k+NM}}\left[C^k(\mathbf{z}^{k+1:k+NM}, \mathbf{v}^k)\right]. \quad (5.33)$$

The finite horizon optimal control problem, at time k, becomes the parameter optimization problem:

$$\mathbf{v}^{k^*} = \arg \min_{\mathbf{v}^k \in \mathcal{V}} J^k(\mathbf{v}^k). \quad (5.34)$$

For a rolling time horizon, this must be resolved every $N_r \leq NM$ time step.

This parameter optimization problem translates directly to the distributed decision problem discussed earlier and, provided that the objective function is partially separable, can be solved using the decentralized optimization algorithm given in Algorithm 5.1.

5.7 Example: Object Localization

The proposed decentralized control strategy was implemented in a simulated object localization task. For this scenario robots equipped with bearing-only sensors (see Fig. 5.2) and moving in a 2D plane are required to cooperatively localize a collection of stationary point objects.

The multiagent control problem for this type of scenario was previously examined by Grocholsky et al. (2003). In this work each agent shared past observations but developed a plan independently. This section extends the work of Grocholsky by applying the decentralized optimization procedure to find the optimal joint plans.

5.7.1 Modelling

Objects

The state $\mathbf{x}$ is separated into m independent objects; thus

$$\mathbf{x} = \{\mathbf{x}_{o_1}, \ldots, \mathbf{x}_{o_m}\}. \tag{5.35}$$

Since the objects are stationary the time index has been dropped.

Each object o_j is specified by a 2D position $\mathbf{x}_{o_j} = [x_{o_j}, y_{o_j}]^T$ that is independent of all other objects.

Fig. 5.2. Typical mobile robot equipped with a bearing-only sensor (panoramic camera).

Agent Motion

The agents are based on an aircraft model and are described by their position and orientation, $\mathbf{s}_i^k = [x_i^k, y_i^k, \theta_i^k]^T$, travel at a constant velocity $V_i = 50$m/s, and are controlled via a single scalar defining the robots rate of turn $\mathbf{u}_i^k = \dot{\theta}_i^k$. Thus, the deterministic motion model $\mathbf{s}_i^{k+1} = \mathbf{f}_i(\mathbf{s}_i^k, \mathbf{u}_i^{k+1})$ is given by

$$x_i^{k+1} = x_i^k + \frac{2V_i}{\mathbf{u}_i^{k+1}} \sin(\frac{1}{2}\mathbf{u}_i^{k+1}\Delta t) \cos(\theta_i^{k+1} + \frac{1}{2}\mathbf{u}_i^{k+1}\Delta t) \quad (5.36a)$$

$$y_i^{k+1} = x_i^k + \frac{2V_i}{\mathbf{u}_i^{k+1}} \sin(\frac{1}{2}\mathbf{u}_i^{k+1}\Delta t) \sin(\theta_i^{k+1} + \frac{1}{2}\mathbf{u}_i^{k+1}\Delta t) \quad (5.36b)$$

$$\theta_i^{k+1} = \theta_i^k + \mathbf{u}_i^{k+1}\Delta t, \quad (5.36c)$$

where Δt is the time between k and $k+1$.

Observations

It is assumed that each agent i receives an independent bearing observation $\mathbf{z}_{i,o_j}^k$ from each object o_j at each time k; thus $\mathbf{z}_i^k = \{\mathbf{z}_{i,o_j}^k : \forall\, j\}$. The independency assumption allows the observations of the objects to be modelled separately.

The i^{th} agent's observation model for object o_j, defined as the conditional probability density, is assumed to be Gaussian and is given by

$$P(\mathbf{z}_{i,o_j}^k | \mathbf{x}_{o_j}; \mathbf{s}_i^k) = N(\mathbf{z}_{i,o_j}^k; \mathbf{h}_{i,o_j}(\mathbf{x}_{o_j}, \mathbf{s}^k), \mathbf{R}_{i,o_j}^k). \quad (5.37)$$

Here, the notation $N(\xi; \mathbf{m}_\xi, \mathbf{C}_\xi)$ represents a Gaussian (or normal) density defined on the state ξ with a mean of $\mathbf{m}_\xi$ and variance $\mathbf{C}_\xi$.

The mean observation for a given object o_j with state $\mathbf{x}_{o_j}$ when the agent is in state $\mathbf{s}_i^k$ is given by the nonlinear function

$$\begin{aligned} \bar{\mathbf{z}}_{i,o_j}^k &= \mathbf{h}_{i,o_j}(\mathbf{x}_{o_j}, \mathbf{s}_i^k) \\ &= \tan^{-1}\Big(\frac{y_{o_j} - y_i^k}{x_{o_j} - x_i^k}\Big). \end{aligned} \quad (5.38)$$

The variance of the bearing observations is set to $\mathbf{R}_{i,o_j}^k = 25°$ for all agents and objects.

5.7.2 Filtering

Consider some time $k-1$, where the teams belief about $\mathbf{x}_{o_j}$ is Gaussian with mean $\bar{\mathbf{x}}_{o_j}^{|k-1}$ and covariance $\mathbf{P}_{o_j}^{|k-1}$:

$$P(\mathbf{x}_{o_j} | \tilde{\mathbf{Z}}^{k-1}; \tilde{\mathbf{S}}^{k-1}) = N(\mathbf{x}_{o_j}; \bar{\mathbf{x}}_{o_j}^{|k-1}, \mathbf{P}_{o_j}^{|k-1}). \quad (5.39)$$

Here the notation $(\cdot)^{|k-1}$ represents "given information up to $k-1$."

Since all the objects are independent, the belief is simply given by the product of the individual probability densities for each object:

$$P(\mathbf{x}^{k-1}|\tilde{\mathbf{Z}}^{k-1};\tilde{\mathbf{S}}^{k-1}) = \prod_{j=1}^{m} P(\mathbf{x}_{o_j}^{k-1}|\tilde{\mathbf{Z}}^{k-1};\tilde{\mathbf{S}}^{k-1}). \tag{5.40}$$

Thus, only the belief $P(\mathbf{x}_{o_j}^{k-1}|\mathbf{Z}^{k-1};\mathbf{S}^{k-1})$ will be considered.

Owing to the static nature of the objects, the prediction step (corresponding to Eq. (5.29)) can be skipped. If the update step (corresponding to Bayes rule Eq. (5.30)) is implemented directly, due to the nonlinearity in the observation model, the posterior will not be Gaussian (even if the prior is Gaussian). The extended Kalman filter (Maybeck 1982) overcomes this by linearizing the observation model about some nominal state ${}_n\mathbf{x}_{o_j}^k$, given by the prior mean $\bar{\mathbf{x}}_{o_j}^{|k-1}$.

Using a first-order Taylor expansion on $\mathbf{h}_{i,o_j}(\mathbf{x}_{o_j}, \mathbf{s}_i^k)$ about ${}_n\mathbf{x}_{o_j}^k$ yields

$$\mathbf{h}_{i,o_j}(\mathbf{x}_{o_j}, \mathbf{s}_i^k) \approx {}_n\mathbf{z}_{i,o_j}^k + \mathbf{H}_{i,o_j}^k\,[\mathbf{x}_{o_j} - {}_n\mathbf{x}_{o_j}^k], \tag{5.41}$$

where ${}_n\mathbf{z}_{i,o_j}^k = \mathbf{h}_{i,o_j}({}_n\mathbf{x}_{o_j}^k, \mathbf{s}_i^k)$ is the nominal observation and the matrix $\mathbf{H}_{i,o_j}^k = \nabla_{\mathbf{x}}\mathbf{h}_{i,o_j}({}_n\mathbf{x}_{o_j}^k, \mathbf{s}_i^k)$ is the Jacobian of the observation function evaluated at the nominal object state and agent state.

For a bearing observation these become

$${}_n\mathbf{z}_{i,o_j}^k = \tan^{-1}\left(\frac{{}_ny_{o_j}^k - y_i^k}{{}_nx_{o_j}^k - x_i^k}\right) \tag{5.42}$$

and

$$\mathbf{H}_{i,o_j}^k = \frac{1}{{}_nr_{i,o_j}^k}\left[-\sin({}_n\mathbf{z}_{i,o_j}^k),\ \cos({}_n\mathbf{z}_{i,o_j}^k)\right], \tag{5.43}$$

where ${}_nr_{i,o_j}^k = \sqrt{({}_ny_t^k - y_i^k)^2 + ({}_nx_{o_j}^k - x_i^k)^2}$ is the range from the agent to the nominal object state.

Now, with this linearized model the posterior will remain Gaussian for a Gaussian prior. The following update equations take the prior mean and covariance and an observation and produce the associated posterior mean and covariance. It is given in information or inverse covariance form by defining the Fisher information matrix $\mathbf{Y}$ as the inverse covariance, i.e., $\mathbf{Y} \equiv \mathbf{P}^{-1}$,

$$\mathbf{Y}_{o_j}^{|k} = \mathbf{Y}_{o_j}^{|k-1} + \sum_{i=1}^{p}(\mathbf{H}_{i,o_j}^k)^T\,(\mathbf{R}_{i,o_j}^k)^{-1}\,\mathbf{H}_{i,o_j}^k, \tag{5.44}$$

and

$$\mathbf{Y}_{o_j}^{|k}\,\bar{\mathbf{x}}_{o_j}^{|k} = \mathbf{Y}_{o_j}^{|k-1}\,\bar{\mathbf{x}}_{o_j}^{|k-1} + \sum_{i=1}^{p}(\mathbf{H}_{i,o_j}^k)^T\,(\mathbf{R}_{i,o_j}^k)^{-1}\,\big(\tilde{\mathbf{z}}_{i,o_j}^k - {}_n\mathbf{z}_{i,o_j}^k + \mathbf{H}_{i,o_j}^k\,\bar{\mathbf{x}}_{o_j}^{|k-1}\big). \tag{5.45}$$

An interesting property of this representation is that the updated or posterior information matrix $\mathbf{Y}_{o_j}^{|k}$ (and hence covariance $\mathbf{P}_{o_j}^{|k}$) is independent of the actual observations, $\mathbf{z}_i^k$, taken [see Eq. (5.44)]. This is an important property and will allow the expected entropy required for the objective function to be calculated very efficiently.

5.7.3 Objective Function

The objective function, defined in Section 5.6.4, represents the expected posterior entropy of the team belief at the end of an NM step time horizon. Since the objects are independent (the density can be decomposed into the product of the densities of each object alone), the entropy becomes a sum of the entropies of each individual object; thus

$$H_{P(\mathbf{x}|\tilde{\mathbf{Z}}^{k+NM};\tilde{\mathbf{S}}^{k+NM})} = \sum_{j=1}^{m} H_{P(\mathbf{x}_{o_j}|\tilde{\mathbf{Z}}^{k+NM};\tilde{\mathbf{S}}^{k+NM})}, \tag{5.46}$$

where the entropy of the Gaussian density for the object state is given by

$$H_{P(\mathbf{x}_{o_j}|\tilde{\mathbf{Z}}^{k+NM};\tilde{\mathbf{S}}^{k+NM})} = -\frac{1}{2}\log\Big((2\pi e)^{d_x}\left|\mathbf{Y}_{o_j}^{|k+NM}\right|\Big), \tag{5.47}$$

with $d_x = 2$ the dimension of the state $\mathbf{x}_{o_j}$.

It is noted that the entropy is only dependent on the information matrix $\mathbf{Y}_{o_j}^{|k+NM}$. By examining the modelling and filtering equations of Section 5.7.2, it can be seen that the observations only influence the covariance or information matrix (hence the posterior entropy) by changing the point at which the observation model is linearized (through changing the prior densities mean).

Thus, to remove this dependency and the requirement to perform the expectation in Eq. (5.33), the nominal state, about which the observation model is linearized, will be given by the mean object state at time k:

$${}_n\mathbf{x}_{o_j}^{k+l} = \bar{\mathbf{x}}_{o_j}^{k}, \quad \forall\, l \in \{1,\ldots,NM\}. \tag{5.48}$$

Hence, the posterior information matrix will be independent of the observation sequence $\tilde{\mathbf{z}}_{o_j}^{k+1:k+NM}$ and may be evaluated directly using

$$\mathbf{Y}_{o_j}^{|k+NM} = \mathbf{Y}_{o_j}^{|k} + \sum_{i=1}^{p}\sum_{l=1}^{NM} \mathbf{I}_i^{k+l}(\mathbf{v}_i^k), \tag{5.49}$$

where $\mathbf{I}_{i,o_j}^{k+l}(\mathbf{v}_i^k)$ is the observation information matrix and is given by

$$\mathbf{I}_{i,o_j}^{k+l}(\mathbf{v}_i^k) = (\mathbf{H}_{i,o_j}^{k+l})^T\,(\mathbf{R}_{i,o_j}^{k+l})^{-1}\,\mathbf{H}_{i,o_j}^{k+l}. \tag{5.50}$$

The posterior entropy of object o_j is now given by

$$\begin{aligned} J_{o_j}^k(\mathbf{v}^k) &= H_{P(\mathbf{x}_{o_j}|\tilde{\mathbf{Z}}^{k+NM};\tilde{\mathbf{S}}^{k+NM})} \\ &= -\frac{1}{2}\log\Big((2\pi e)^{d_x}\left|\mathbf{Y}_{o_j}^{|k+NM}\right|\Big) \end{aligned} \tag{5.51}$$

and the final team objective function, corresponding to the joint entropy of all objects, is

$$J^k(\mathbf{v}^k) = \sum_{j=1}^{m} J^k_{o_j}(\mathbf{v}^k). \tag{5.52}$$

Partial Separability

For each agent to evaluate the objective function, it requires the sum of the observation information matrices from all other agents and over all steps in the planning horizon. The actual observation models, $\mathbf{H}^k_{i,o_j}$ and $\mathbf{R}^k_{i,o_j}$, and the position of the agents $\mathbf{s}^k_i$ are irrelevant once this information is obtained.

- *Impact Space:* Due to this structure, an impact space can be defined that contains the required matrices for each of the m objects.

$$\mathcal{I} = \prod_{j=1}^{m} \mathcal{M}^S_{2\times 2}, \tag{5.53}$$

  where $\mathcal{M}^S_{2\times 2}$ is the vector space containing all symmetric 2×2 matrices.
- *Impact Function:* The impact function for an agent i maps a decision onto an element of this space by summing the individual information matrices $\mathbf{I}^{k+l}_{i,o_j}(\mathbf{v}^k_i)$ for all $l \in \{1, \ldots, NM\}$ for each object $j \in \{1, \ldots, m\}$ and, i.e.,

$$\Upsilon_i(\mathbf{v}^k_i) = \Big\{ \sum_{l=1}^{NM} \mathbf{I}^{k+l}_{i,o_j}(\mathbf{v}^k_i) : \forall\, j \in \{1, \ldots, m\} \Big\}. \tag{5.54}$$

- *Composition Operator:* This operator combines impacts from different agents. It is given by matrix addition and simply adds corresponding observation information matrices. Thus, if $\alpha^k_a = \Upsilon_a(\mathbf{v}^k_a)$ and $\alpha^k_b = \Upsilon_b(\mathbf{v}^k_b)$, the composition operator is given by

$$\alpha^k_a * \alpha^k_b = \Big\{ \sum_{l=1}^{NM} \mathbf{I}^{k+l}_{a,o_j} + \sum_{l=1}^{NM} \mathbf{I}^{k+l}_{b,o_j} : \forall\, j \in \{1, \ldots, m\} \Big\}. \tag{5.55}$$

- *Generalized Objective Function:* This function evaluates the cost (expected posterior entropy) of the agents decisions directly from the combined system impact. Consider the combined impact $\alpha^k_T = \bigodot_{i=1}^{p} \Upsilon_i(\mathbf{v}^k_i)$, given as

$$\alpha^k_T = \Big\{ \alpha^k_{T,o_j} : \forall\, j \in \{1, \ldots, m\} \Big\}, \tag{5.56}$$

  where $\alpha^k_{T,o_j} = \sum_{i=1}^{p} \sum_{l=1}^{NM} \mathbf{I}^{k+l}_{i,o_j}$. Now, the generalized objective function $\psi^k(\alpha^k_T)$ is given by duplicate word

$$\psi^k(\alpha^k_T) = -\sum_{j=1}^{m} \frac{1}{2} \log\Big((2\pi e)^{d_x} \left| \mathbf{Y}^{|k+NM}_{o_j} \right| \Big), \tag{5.57}$$

where $\mathbf{Y}_{o_j}^{|k+NM}$ is given

$$\mathbf{Y}_{o_j}^{|k+NM} = \mathbf{Y}_{o_j}^{|k} + \alpha_{T,o_j}^{k}. \tag{5.58}$$

5.7.4 Collaborative Control

With the above definition, the collaborative multiagent decision problem becomes

$$\mathbf{v}^{k^*} = \arg\min_{\mathbf{v}^k \in \mathcal{V}} \psi^k(\alpha_T^k), \tag{5.59}$$

where $\alpha_T^k = \bigodot_{i=1}^{p} \Upsilon_i(\mathbf{v}_i^k)$, and is solved using Algorithm 5.1.

5.8 Results

5.8.1 Two Agents—Single Object

To demonstrate the workings of the decentralized optimization algorithm, a system comprising two agents observing a single stationary object is considered. The configuration is shown in Fig. 5.3. For this scenario each agent has to decide on a control policy consisting of a single control parameter ($N = 1$) that defines its rate of turn over a planning horizon of 12 s.

The optimal joint plans are found by each agent executing Algorithm 5.1. Although this procedure is designed to allow asynchronous execution, it was executed synchronously. Agents communicated after every local iteration with an imposed communication delay corresponding to three iterations.

As each agent has only a bearing sensor, individually they have a very poor ability to localize the object. However, if they cooperate and observe the object from perpendicular directions they can greatly minimize its position uncertainty. However, there is

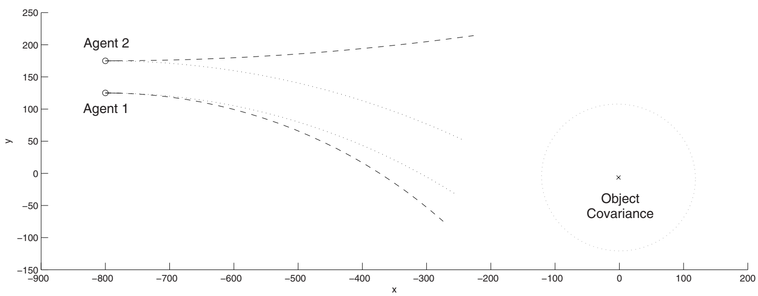


Fig. 5.3. The dashed trajectories corresponding to the jointly optimal plans. The dotted trajectories represent the optimal solution to the corresponding single-agent problem and were used to initialize the negotiated solution. The prior probability density of the position of the object is given by a Gaussian with the mean shown by the cross (×) and the variance by the dotted circle.

also a dependency on the range at which they observe the object, such that a shorter range will give a smaller uncertainty in the measured position.

These seemingly opposing objectives are captured by the single objective function defined in Eq. (5.52). As shown in Fig. 5.3, the optimal decisions cause the agents to move towards the object and separate, such that a better triangulation angle is obtained.

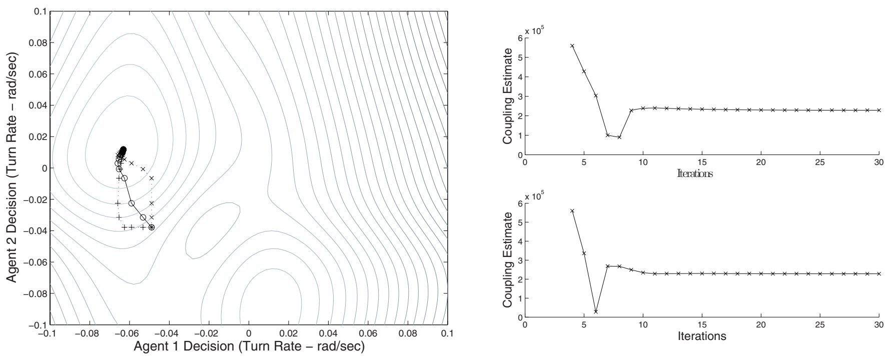
Although this resulting behaviour is intuitive to the observer, an agent cannot reason about this sort of global behaviour. Each agent only knows about the other through the communication of abstract *impacts*; the position of the other agent and its planned trajectory are completely unknown.

Figure 5.4a displays the evolution of the agents' decisions throughout the optimization procedure. Although the communication delay causes a significant difference in the perceived trajectories through the decision space, the system still converges to the optimum. It is noted that one agent never knows about the actual decision of the other; it only knows its impact. Figure 5.4a simply plots the decision corresponding to this impact.

Figure 5.4b plots the interagent coupling as approximated by both agents. The curves have similar shapes, but are not identical because they are measuring the curvature of the objective function at different points in the decision space.

5.8.2 Nine Agents—Eighteen Objects

This scenario consists of nine agents cooperatively localizing 18 objects. The agents start from the left side of Fig. 5.5a, in an arrow formation. They take observations at a rate of 2 Hz and plan a trajectory 16 s into the future (corresponding to an 800-m path). The trajectory is parametrized by four variables defining the required turn rates



(a) (b)

Fig. 5.4. (a) Evolution of agents' decisions through the global decision space $\mathcal{V}$ during the optimization procedure, overlaid with contours of the objective function. The true path $\mathbf{v} = [{}^1\mathbf{v}_1, {}^2\mathbf{v}_2]^T$ is shown with circles ($\circ$), while the pluses ($+$) and crosses ($\times$) represent the perceived path ${}^i\mathbf{v} = [{}^i\mathbf{v}_1, {}^i\mathbf{v}_2]^T$ for agents $i = 1, 2$, respectively. The difference is caused by the communication delays. (b) Coupling estimates ${}^1\hat{K}_{12}$ (top) and ${}^2\hat{K}_{21}$ (bottom) calculated by agents 1 and 2, respectively.

for each quarter. This corresponds to $N = 4$, $M = 8$, and $\Delta t = 0.5\,\mathrm{s}$ and results in an optimal planning problem consisting of 36 parameters distributed over the nine agents.

A rolling planning horizon is employed, requiring that a new plan be developed every second (i.e., $N_r = 2$). When generating the very first plan, the agents initialize their decisions using the locally optimal decision (as discussed in Sect. 5.8.1 for the two-agent case); however at all later stages the decisions are initialized using the previous decision.

The snapshots of the system are shown in Fig. 5.5a–d. Figure 5.5d shows the final state of the system after all the objects have been sufficiently localized, but the current optimal plans are also shown for completeness.

Figure 5.6 shows some data of a typical run of the optimization algorithm. It plots the estimated coupling constants, communication events, and the evolution of the decision parameters from the perspective of a single agent. These data are for agent 6 for the very first decision problem (the results of which are shown in Fig. 5.5a) and demonstrate the relation between coupling (top graph) and communication frequency (middle graph). The agent communicates at a high frequency to agent 5, to which it is coupled

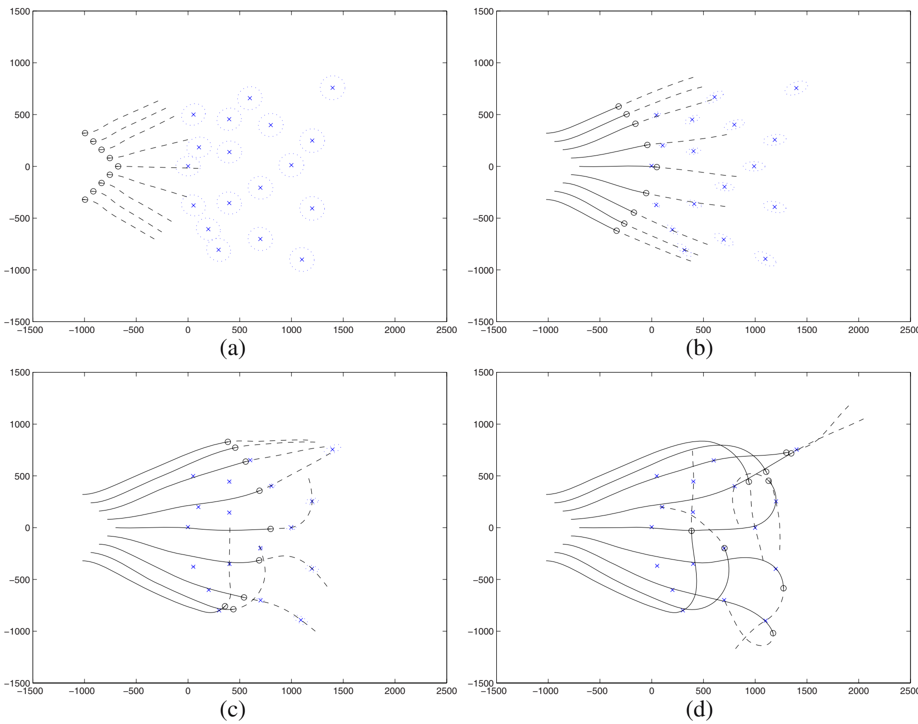


Fig. 5.5. Snapshots throughout the scenario at (a) $k = 0$, (b) $k = 30$, (c) $k = 60$, and (d) $k = 90$. Current agent positions are shown with a circle (○), and the optimal future planned trajectory with a dotted line. The current probability density of the location of each object is represented by its mean (×) and covariance (dotted circle).

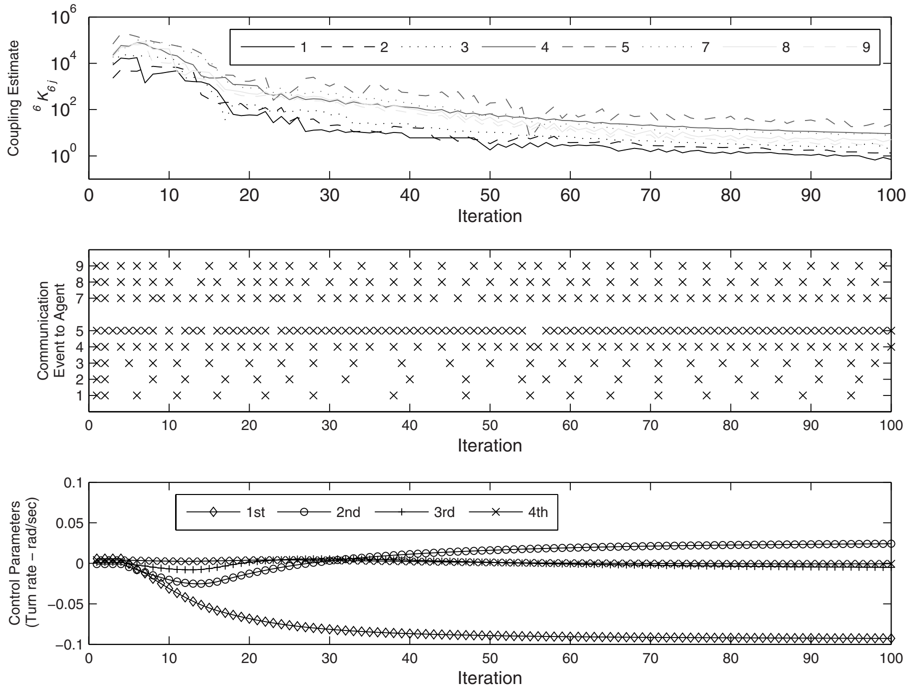


Fig. 5.6. Typical data during a single run of the distributed optimization algorithm. These data correspond to agent 6 for $k = 0$, as seen in to Fig. 5.5a. Top: Estimated coupling terms ${}^6\hat{K}_{6j}$ for $j \in \{1, 2, 3, 4, 5, 7, 8, 9\}$. Middle: Each cross indicates the time a message is sent to each specific agent in the system (the frequency of these events is determined by the coupling metric, according to Eq. (5.19). Bottom: Evolution of the agent's decision parameters (corresponding to the agent's rate of turn during the planning horizon).

the most, and at a much lower frequency to other agents (especially agent 1), where the interagent coupling is smaller.

Figure 5.7 displays the interagent coupling for the whole system for each snap shot in Fig. 5.5. The i^{th} row of each matrix represents the average of ${}^i\hat{K}_{ij}$ over all the iterations of Algorithm 5.1 for every other agent j. As expected, the matrix is reasonably symmetric (the coupling terms correspond to cross derivatives, which by definition are symmetric) and shows a large amount of structure. The matrix in Fig. 5.7a displays the intuitive result that agents close to each other are highly coupled (due to the diagonal structure).

However, agent separation is not directly important; the coupling essentially measures the sensitivity of the effect of the information that one agent receives from its observations on the decisions of another. This is demonstrated in the last matrix corresponding to Fig. 5.5d. At this time in the mission all the objects are well localized and for the agents to gather more information (and reduce the uncertainty) about the positions of the objects, they must travel very close to them. Thus, only agents with

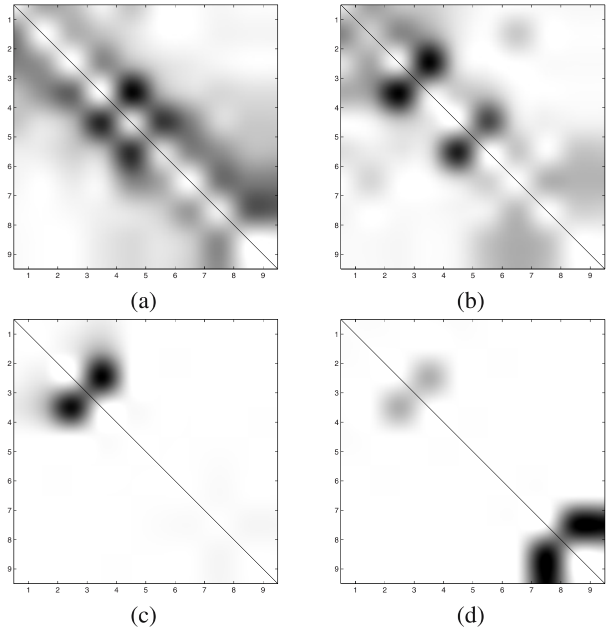


Fig. 5.7. Coupling matrix for the initial decision problem. The checker board type appearance (especially the two minor diagonals) represent that generally agents closer to each other are more strongly coupled (see Fig. 5.5(a)).

planned trajectories passing by a common object are coupled, e.g., agents 8 and 9 and agents 3 and 4.

This coupling metric captures how the underlying agent models interact through the system's objective function, which is precisely what is important for a multiagent system undertaking a specific task.

5.9 Discussion and Future Work

This chapter has approached the problem of multiagent decision making and planning using the tools of asynchronous distributed optimization. This analytical approach led to the definition of a coupling metric that intuitively links the rate at which an agent may refine its local decision to its interagent communication frequency and transmission delays. The coupling is determined by the cross derivatives of the objective function and captures how the underlying agent models interact with the definition of the system's task.

This decentralized negotiation algorithm was used to control a simulated multi-agent system involving multiple mobile robots undertaking an object localization task. This example demonstrated that agents are required to communicate often only with the agents to which they are coupled.

It is envisaged for much larger distributed systems that sparseness of inter-agent coupling (e.g., as shown in Fig. 5.7) will be more prevalent, causing each agent to be coupled only to a small subset of the system. This will overcome the main requirement that each agent must negotiate with every other agent, as only agents that are coupled are required to communicate.

Another extension under investigation is the use of hierarchical clustering techniques, such as described in Balch (2000), to build and maintain dynamic hierarchies within the system. This can be illustrated by considering the coupling structure shown in Fig. 5.7b, which can be easily partitioned into three clusters of $\{1, 2, 3, 4\}$, $\{5, 6\}$, and $\{7, 8, 9\}$. Agents within a cluster are highly coupled and communicate directly to each other, while intercluster communication is directed via a cluster head, which summarizes the impacts of all the agents in the cluster. This abstraction of agents to clusters can be taken to higher levels with clusters of clusters and clusters of clusters of clusters, and so forth, each participating in higher-level negotiations.

Acknowledgements

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

Balch, T. (2000). Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, Belmont, Massachusetts 8(3):209–138.

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*. Athena Scientific.

Cover, T. M., and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley, New York.

Furukawa, T., Dissanayake, G., and Durrant-Whyte, H. F. (2003). Time-optimal cooperative control of multiple robot vehicles. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA 2003, September 14–19, 2003, Taipei, Taiwan*, pages 944–950. IEEE.

Goh, C., and Teo, K. (1988). Control parametrization: a unified approach to optimal control problems with general constraints. *Automatica*, 24(1):3–18.

Grocholsky, B., Makarenko, A., Kaupp, T., and Durrant-Whyte, H. F. (2003). Scalable control of decentralised sensor platforms. In Zhao, F., and Guibas, L. J., editors, *Information Processing in Sensor Networks, Second International Workshop, IPSN 2003, Palo Alto, CA, April 22–23, 2003, Proceedings*, volume 2634 of *Lecture Notes in Computer Science*, pages 96–112. Springer, New York.

Liggins, M.E., I., Chong, C.-Y., Kadar, I., Alford, M., Vannicola, V., and Thomopoulos, S. (1997). Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1): 95–107. New York.

Manyika, J., and Durrant-Whyte, H. F. (1994). *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Ellis Horwood, New York.

Mathews, G. M., Durrant-Whyte, H. F., and Prokopenko, M. (2006). Scalable decentralised decision making and optimisation in heterogeneous teams. In *Proccedings of IEEE International Conference on Multi-Sensor Fusion and Integration for Intelligent Systems (MFI-2006), Heidelberg, Germany*, pages 383–388. IEEE.

Maybeck, P. S. (1979–1982). *Stochastic Models, Estimation and Control*. Academic, New York.

Tsitsiklis, J. N., Bertsekas, D. P., and Athens, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, AC-31:803–812.