

1

Dimensions of cooperative control

Jeff S. Shamma and Gurdal Arslan

1.1 WHY COOPERATIVE CONTROL?

1.1.1 Motivation

Cooperative control is concerned with engineered systems that can be characterized as a collection of decision-making components with limited processing capabilities, locally sensed information, and limited inter-component communications, all seeking to achieve a collective objective. Examples include mobile sensor networks, network congestion control and routing, transportation systems, autonomous vehicle systems, distributed computation, and power systems. Areas of research that are related to cooperative control include ‘multi-agent control’, ‘distributed systems’, ‘networked control’, as well as ‘team theory’ or ‘swarming’. Regardless of the nomenclature, the central challenge remains the same. That is, to derive desirable collective behaviors through the design of individual agent control algorithms.

The primary distinguishing feature of a cooperative control system is distribution of information. As opposed to ‘centralized’ solutions, no one decision-maker has access to the information gathered by all agents. Furthermore, there is typically a communication cost in distributing locally gathered information.

A secondary distinguishing feature is complexity. Even if information were centrally available, the inherent complexity of the decision problem makes a centralized solution computationally infeasible. Therefore, one looks to ‘decomposition’ approaches as in cooperative control.

Both these features result in the distributed decision architecture of cooperative control. The potential benefits of such architectures include the opportunity for real-time adaptation (or self-organization) and robustness to dynamic uncertainties such as individual component failures, non-stationary environments, and adversarial elements. These benefits come with significant challenges, such as the complexity associated with a potentially large number of interacting agents and the analytical difficulties of dealing with overlapping and partial information.

1.1.2 Illustrative example: command and control of networked vehicles

We will use the vehicle-target assignment problem, described below, as an illustrative scenario for the discussion of many of the concepts in this chapter.

Recent years have seen a boom of research motivated by the application of mobile unmanned systems that operate with either complete autonomy or semi-autonomy, i.e., given high level commands from a remote human operator. In fact, the broader topic of ‘network-centric operations’ was the subject of a very recent National Research Council committee’s study on Network Science for Future Army Applications.¹

A representative scenario is the vehicle-target assignment problem (Ahuja *et al.* 2003; Beard *et al.* 2002; Murphey 1999), illustrated in Figure 1.1. A collection of vehicles seeks to visit a collection of targets. Each vehicle has a limited range of reachable targets. There can be an advantage in having multiple vehicles visit a single target in that vehicle capabilities are complementary. However, there is also a diminishing return having redundancy in vehicle-target assignments. An additional complication is to allow vehicles to visit multiple targets in succession.

Vehicle-target assignment also encompasses the problem of multiple vehicle motion planning. In an obstacle-filled congested environment, vehicles need to get to their destination while avoiding obstacles and avoiding each other, yet having only limited communications with neighboring vehicles.

Again, both distinguishing features of cooperative control problems are apparent. Because of the spatial distribution of vehicles, any information gathered by a single vehicle does not reach other vehicles without explicit communication. This adds a dimension of what to communicate and to whom. Furthermore, the vehicle-target assignment problem is known to be NP-complete (Murphey 1999), and so even a centralized solution is computationally prohibitive for large numbers of vehicles.

An additional possible complication is the presence of an intelligent adversary. Most work on vehicle-target assignment assumes an asymmetry between vehicle and target capability – i.e., targets have simple scripted behavior. The problem becomes much more

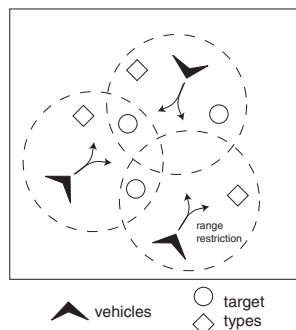


Figure 1.1 Vehicle target assignment.

¹ The NRC report, *Network Science*, is available online at <http://darwin.nap.edu/books/0309100267/html/>.

interesting if targets also have strategic capability, e.g., mobile targets with stealth modes of operation. In this case, the problem resembles a sort of networked team-vs-team encounter.

1.2 DIMENSIONS OF COOPERATIVE CONTROL

It is helpful to impose some sort of structure on the broad scope of issues that can emerge in cooperative control. With this motivation, we will discuss cooperative control problems in terms of the following ‘dimensions’: (1) distributed control and computation; (2) adversarial interactions; (3) uncertain evolution; and (4) complexity management.

In the following subsections, we will describe each of these dimensions in general and in terms of the above illustrative example. The discussion will largely be in terms of chapters in this volume and selected immediately relevant results. Other reviews of cooperative control and related research may be found in Murray (2006) and Panai and Luke (2005).

1.2.1 Distributed control and computation

This is the primary defining characteristic of cooperative control. The distribution of information necessitates the distributed control and computation among interacting components. Not surprisingly, this dimension represents the highest number of contributions to the volume.

1.2.1.1 Multivehicle motion planning

Multivehicle motion planning arises in vehicle-target assignment in congested environments. The objective is to generate real-time trajectories to guide a collection of vehicles to their destination while avoiding obstacles and each other. Destinations may be specified in terms of absolute locations, relative locations, coverage patterns, or even general direction of flow. Example scenarios include rendezvous or formation keeping.

The distributed nature of multivehicle motion planning stems from each vehicle determining its own trajectory while having only partial measurements of only neighboring vehicles.

Chapter 2² focuses on a ‘swarm’ of vehicles, i.e., a large collection of autonomous vehicles that need not maintain a regular formation. An important underlying technique in this problem is that of ‘consensus algorithm’, which was introduced in Tsitsiklis *et al.* (1986) and has received considerable renewed interest in recent years (Blondel *et al.* 2005; Jadbabaie *et al.* 2003; Kashyap *et al.* 2006; Marden *et al.* 2007; Moreau 2004; Olfati-Saber and Murray 2003; Olfati-Saber *et al.* 2007; Xiao and Boyd 2004, 2005). The general form of the algorithm can be described as follows. Let $i \in \{1, 2, \dots, \mathcal{I}\}$

² R. Olfati-Saber, ‘Design of behavior of swarms: from flocking to data fusion using microfilter networks’.

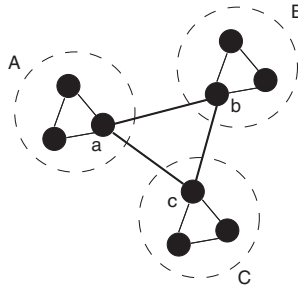


Figure 1.2 Hierarchical formation.

denote the index of the i^{th} agent in a collection $\{1, 2, \dots, \mathcal{I}\}$. Let $\theta_i(t)$, $t \in \{0, 1, 2, \dots\}$, denote a time-varying scalar quantity associated with the i^{th} agent. Now define

$$\theta_i(t+1) = \sum_{j \in N_i(t)} a_{ij}(t)\theta_j(t), \quad (1.1)$$

where $N_i(t) \subset \{1, 2, \dots, \mathcal{I}\}$ denotes the set of ‘neighbors’ of the i^{th} agent at time t and $a_{ij}(t)$ is the relative weight that agent i places on agent j . Under suitable assumptions, one can show that all agents asymptotically synchronize, i.e., for all $i \neq j$,

$$\lim_{t \rightarrow \infty} \theta_i(t) = \lim_{t \rightarrow \infty} \theta_j(t).$$

For motion planning problems, the consensus algorithm has been used as an approach to synchronize heading angles, velocities, or positions, using only ‘local’ information characterized by $N_i(t)$ and in the face of time-varying network topologies.

Chapter 3³ also addresses multivehicle motion planning, but the emphasis is shifted away from swarms in favor of formations. That is, each vehicle seeks to hold a specified distance from a subset of other vehicles. In particular, this chapter considers ‘hierarchical formations’. A very simple example (Figure 1.2) consists of three groups of vehicles, say, group A , B , and C . Group A seeks to maintain a relative formation, as do groups B and C . Furthermore, there is a single vehicle in each group, say, vehicles a , b , and c , that seek to maintain a relative formation with each other. In this way, the overall formation is induced by the intergroup formation and intragroup formation.

1.2.1.2 Distributed predictive control

Receding horizon control or model predictive control (e.g., Bemporad and Morari 1999) is a well-known approach to nonlinear control design through successive online optimizations. Distributed control and computation present an obstacle to implementing predictive control in a multiagent system.

³ S. Glavaški, A. Williams, and T. Samad, ‘Connectivity and convergence of formations’.

The ‘single agent’ version can be described briefly as follows. Assume state dynamics of the form

$$\dot{x} = f(x, u).$$

Let $u^*(\cdot)$ be the optimal control trajectory for the optimization problem

$$\min_{u(\cdot)} \int_t^{t+T} g(x(\tau), u(\tau), \tau) d\tau + G(x(t+T))$$

subject to the above dynamics with initial condition $x(t)$. In model predictive control, the initial portion of the optimal control is implemented, i.e.,

$$u(\tau) = u^*(\tau), \quad \tau \in [t, t + \delta)$$

for some $\delta > 0$. The optimization is then resolved at time $t + \delta$ using the state $x(t + \delta)$ as the initial condition, and the process is repeated.

Two complications arise in a multiagent framework. For simplicity, consider the case with two agents. One complication arises when the systems are dynamically coupled, as in

$$\dot{x}_1 = f_1(x_1, x_2, u_1)$$

$$\dot{x}_2 = f_2(x_1, x_2, u_2).$$

Another complication arises when the systems are not dynamically coupled, as in

$$\dot{x}_1 = f_1(x_1, u_1) \tag{1.2a}$$

$$\dot{x}_2 = f_2(x_2, u_2), \tag{1.2b}$$

but their objective functions are coupled, as in

$$\min_{u_1(\cdot)} \int_t^{t+T} g_1(x_1(\tau), x_2(\tau), u_1(\tau), \tau) d\tau + G_1(x_1(t+T)) \tag{1.3a}$$

$$\min_{u_2(\cdot)} \int_t^{t+T} g_2(x_1(\tau), x_2(\tau), u_2(\tau), \tau) d\tau + G_2(x_2(t+T)). \tag{1.3b}$$

In either case, the requirement of distributed control and computation prohibits an immediate application of receding horizon control, the main reason being that neither agent is in control of the other. Accordingly, neither agent is able to control or predict the trajectory of the other agent.

Chapters 4⁴ and 5⁵ present complementary approaches to multiagent versions of predictive control. These chapters address how to introduce additional constraints on the control trajectories, such as $u_i(\cdot) \in \mathcal{U}_i$, to guarantee stability and performance. As one

⁴ W.B. Dunbar, ‘Distributed receding horizon control: stability via move suppression’.

⁵ T. Keviczky, F. Borrelli, and G.J. Balas, ‘Distributed predictive control: synthesis, stability and feasibility’.

would expect, an important underlying concern of any single agent is the deviation of the trajectories of other agents from their predicted values. While the problem set-up is motivated by multivehicle motion planning, the methods and concepts carry over to more general settings.

1.2.1.3 Task assignment

The multivehicle motion planning component of vehicle-target assignment addresses issue of ‘how to get there?’. This was the main subject of the Chapters 2–5.

A higher level strategic decision is ‘where to go’? This is the problem of task assignment. Let $\mathcal{V} = \{1, 2, \dots, n_V\}$ denote a set of vehicles and $\mathcal{T} = \{1, 2, \dots, n_T\}$ denote a set of targets. Let a_i , $i \in \mathcal{V}$ denote the assignment of the i^{th} vehicle. For example, $a_1 = 2$ means that vehicle #1 is assigned to target #2. The collection of assignment profiles is denoted by the vector a . In general, these assignments are constrained (e.g., by vehicle range limitations), and so each vehicle’s assignment must belong to a specified set, denoted by \mathcal{A}_i for vehicle # i .

In static assignment problems, there is a scoring function that indicates the utility of an assignment profile, e.g.,

$$U_g(a) = U_g(a_1, a_2, \dots, a_{n_V}).$$

The task assignment problem is then the combinatorial optimization

$$\max_{a_i \in \mathcal{A}_i, i=1,2,\dots,n_V} U_g(a_1, a_2, \dots, a_{n_V}).$$

The situation is more complicated in dynamic assignment problems. In these problems, vehicle locations, target locations, and target constraint sets are all time-varying, which induces time-variations in the overall utility function. The problem is then to design dynamic vehicle-target assignment policies whose performance is measured by time-aggregated behavior such as the rate of successful target assignments.

This problem becomes one of distributed computation and control because individual vehicles need not be aware of the entirety of target locations. Furthermore, there may be an incentive for vehicles to be assigned to the same target, and so vehicles must coordinate to complement the actions of other vehicles. Typically, one assumes some sort of communication graph over which vehicles message each other in an effort to compute an assignment profile. This brings in the additional dimension of what to communicate as part of the decision problem. Another reason for a distributed approach is complexity. Even if a single decision-maker had access to all target locations to determine an assignment profile, the resulting optimization is known to be NP-complete (cf., Murphey 1999). Imposing a distributed architecture does not alleviate the inherent complexity, per se, but represents an attempt to compute effective solutions with low computational cost.

Chapters 6–7 address different aspects of the task assignment problem. Chapter 6⁶ investigates the one-to-one assignment problem with static targets. Vehicles can communicate through a specified communication topology, but these communications introduce delays in determining an assignment profile. Furthermore, the assignment utility function changes as vehicles move in the environment en route to their interim assignments.

⁶ B.J. Moore and K.M. Passino, ‘Task assignment for mobile agents’.

Vehicles use a modified distributed auction algorithm to determine the action profile, and the performance is measured through Monte Carlo simulation.

Chapter 7⁷ also considers mobile vehicles and static targets, but targets are regenerated by a stochastic process. The performance measure is to minimize the average time to visit a target once it has appeared. This performance objective is an example of the aforementioned time-aggregated behavior (as opposed to optimal assignment for a specific fixed target scenario.) The emphasis in Chapter 7 is to investigate the role of communication. Namely, to what degree do explicit vehicle communications affect performance? Surprisingly, the result in this formulation is that asymptotic performance is not affected. The intuition is that vehicles implicitly communicate when one vehicle reaches a target before another vehicle.

Another approach to task assignment is reported by Arslan *et al.* (2006). This paper takes a learning in games (Fudenberg and Levine 1998; Young 2006) approach to the many-to-one static assignment problem. Since many vehicles can visit the same target, there is an assumed communication topology where two vehicles can communicate if and only if they share a common target, i.e., $\mathcal{A}_i \cap \mathcal{A}_j \neq \emptyset$.

The method proposed by Arslan *et al.* (2006) is twofold. The first element is design of local utility functions. Individual vehicles are endowed with a ‘local’ utility function that reflects a vehicle’s local influence. A particularly appealing local utility construction is the marginal contribution utility (or wonderful life utility (Wolpert and Tumer 1999)) defined as follows. We will write an assignment profile as $a = (a_i, a_{-i})$ where a_i is the assignment of the i^{th} vehicle, and a_{-i} is the assignment of all *other* vehicles. Let T_\emptyset denote the ‘null’ target, i.e., no assignment, and assume that $T_\emptyset \in \mathcal{A}_i$ for all agents. Now define the local utility function

$$U_i(a_i, a_{-i}) = U_g(a_i, a_{-i}) - U_g(T_\emptyset, a_{-i}).$$

In words, $U_i(\cdot)$ reflects the change in utility that the i^{th} vehicle can affect. Any sort of geographical distribution of targets and vehicles reflected in the global utility is automatically reflected in this definition.

The second element is the design of negotiation protocols. Vehicles that share common targets communicate repeatedly and propose self-assignments to each other. The negotiation protocol determines how the new proposals are generated as a (stochastic) function of the history of prior proposals. In particular, Arslan *et al.* (2006) present different protocols and show that these protocols are guaranteed to converge to a Nash equilibrium of the resulting multi-player game. The performance of these protocols is compared through Monte Carlo simulations.

1.2.1.4 Inducing coordination

The problem of inducing coordination is to design individual agent control laws that use very limited descriptions of the behaviors of other agents. In terms of the discussion on predictive control, assume that two agents are not dynamically coupled but are coupled

⁷ A. Arsie, J.J. Enright, and E. Frazzoli, ‘On the value of information in dynamic multiple-vehicle routing problems’.

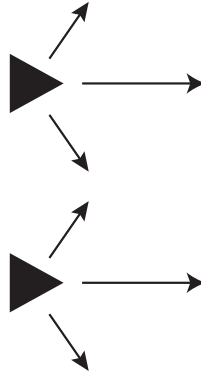


Figure 1.3 Multivehicle search.

through objective functions, as in Equations (1.2)–(1.3). A centralized control policy would have each agent’s control law as a function of the state of both agents, e.g.,

$$u_1 = g_1(x_1, x_2),$$

$$u_2 = g_2(x_1, x_2)$$

The objective in inducing coordination is to derive control laws that only use a suitably defined aggregate measure, denoted by $m_i(\cdot)$, of the behavior of the other agent, as in

$$u_1 = g_1(x_1, m_1(x_2)),$$

$$u_2 = g_2(x_2, m_2(x_1)).$$

Chapter 8⁸ considers the following scenario, illustrated in Figure 1.3. Two agents are traversing an unknown (gridded) map that models terrain with obstructed views. Each agent can measure the cost of travel for many steps ahead in a forward direction but has very limited view of the cost of travel in diagonal direction. The objective is for both vehicles to cross the map with minimum cost. This brings up the well-known common tension between exploration and exploitation. On the one hand, vehicles should explore the terrain in search of low cost paths. On the other hand, if vehicles are too far apart, then neither vehicle can take advantage of a low cost path found by the other. Chapter 8 computes the optimal policy in this scenario and shows that the optimal policy is characterized by an ideal separation. In terms of the above discussion, the separation factor serves as the ‘aggregate measure’ of the other vehicle’s performance. If a vehicle exceeds this separation, then it must be that it has found a path whose cost is below a threshold value.

Chapter 9⁹ investigates a linear version of the vehicle-target assignment problem illustrated in Figure 1.4 motivated by the ‘RoboFlag drill’ of Earl and D’Andrea (2002). As in Chapter 7, targets are regenerated stochastically. The approach here is that each agent makes a simplified model of the other agent in the form of a probability of being on

⁸ E. Feron and J. DeMot, ‘Optimal agent cooperation with local information’.

⁹ V.P.-W. Seah and J.S. Shamma, ‘Multiagent cooperation through egocentric modeling’.

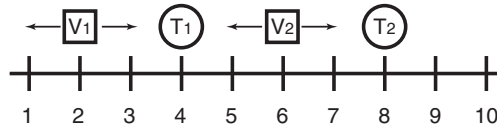


Figure 1.4 Linear vehicle-target assignment.

the left versus being on the right. This ‘probability’ becomes the aggregate measure of the behavior of the other agent. A semi-decentralized policy is computed through the following iterative process. First, an agent forms an empirical model of the behavior of the other agent. Second, an agent designs an optimal policy given this model. Once agents deploy the new policy, they remodel each other’s behavior and redesign their individual policies. The implication of convergence is a consistency condition. Namely, each agent’s behavior is consistent with how the agent is modeled by others. Furthermore, each agent’s local strategy is optimal with respect to how it models other agents. Chapter 9 derives conditions for convergence for the scenario in Figure 1.4, presents illustrative simulations, and establishes a connection to the learning in games approach.

1.2.2 Adversarial interactions

An important application for cooperative control is autonomous vehicle systems operating in a possibly hostile environment. Adversarial interactions highlight that these systems must address the possibility of other hostile vehicles that are also capable of strategic planning. For example in the vehicle-target assignment problems previously discussed, targets exhibited only scripted as opposed to strategic behavior. Targets were either stationary or they regenerated stochastically. In adversarial situation, targets may make strategic decisions related to mobility or visibility.

1.2.2.1 Trajectory planning with adversaries

Both Chapters 10¹⁰ and 11¹¹ look at a particular version of vehicle target assignment called ‘Roboflag drill’ (see Earl and D’Andrea 2002) illustrated in Figure 1.5. Mobile attackers seek to penetrate a defense zone. However, an attacker is disabled if it is ‘tagged’ by a defender. Suppose that attacker paths were known in advance. Then it is possible to set up an optimization problem to compute defender trajectories to intercept attacker positions. The general form of this optimization is a mixed integer linear program (MILP). MILP optimization is similar to linear programming, except that some (or all) variables are constrained to be integers. An illustrative form is

$$\begin{aligned} & \max c^T x \\ & \text{subject to } Ax \leq b, x_i \in \{0, 1\}. \end{aligned}$$

¹⁰ M.G. Earl and R. D’Andrea, ‘Multi-vehicle cooperative control using mixed integer linear programming’.

¹¹ G.C. Chasparis and J.S. Shamma, ‘LP-based multi-vehicle path planning with adversaries’.

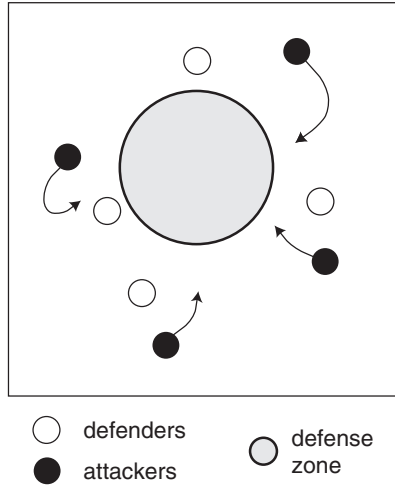


Figure 1.5 RoboFlag drill.

This problem would be a linear program if the constraint set on x were changed to the interval condition $x_i \in [0, 1]$. MILP optimization can be significantly more computationally demanding than LP optimization. However, there are effective commercial solvers. The integer constraints in RoboFlag drill stem from certain binary states in the problem set-up, such as whether or not an attacker has been intercepted. Chapter 10 goes beyond the assumption of a known attacker trajectory. Rather, the assumption is a known *strategy*, which can be viewed as a guidance law, of the attackers.

Chapter 11 addresses a very similar framework. However, the approach taken is to avoid MILP formulations in favor of LP formulations. This is achieved by a predictive control implementation of the defender trajectory planning. The procedure is as follows. First, a simplified model for trajectory planning is constructed. In this simplification, certain discrete elements of the model are ignored, thereby allowing an LP computation of a defender trajectories. An initial phase of the computed trajectory is implemented, at which point the process is repeated. As in Chapter 11, there is an assumed strategy of the attackers. However, this strategy is based on myopic predictions of the attacker trajectories and updated as part of the predictive control iteration. Since the method is based on computationally simpler LP, it allows one to endow both attackers and defenders with equal strategic capability. That is, both attackers and defenders optimize online based on revised strategy predictions of their opponent.

1.2.2.2 Adversary forecasting

An important consideration in both of the above approaches was the lack of knowledge, and hence assumptions on, the strategy, or ‘intent’, of the opponent. In order to compute an optimal response, one must make some sort of model of opponent strategies. Recent work (Mannor *et al.* 2006) addresses this issue in the framework of learning in games. The set-up is as follows. Two players \mathcal{P}_1 and \mathcal{P}_2 repeatedly play a matrix game over times $t = \{0, 1, 2, \dots\}$. Player \mathcal{P}_i selects its action $a_i(t)$ from its action set \mathcal{A}_i . Each player

receives a utility (reward) $U_i(a_i(t), a_{-i}(t))$. (Recall that $-i$ denotes the ‘other’ player.) An important assumption is that players do *not* know each other’s utility function. In particular, a zero-sum structure as in

$$U_1(a_1, a_2) = -U_2(a_2, a_1)$$

is not assumed. This restriction can be interpreted as a lack of knowledge of the intent of the other player. To deal with this lack of knowledge, players make forecasts of each other’s actions. More precisely, given the action history $(a_2(0), a_2(1), \dots, a_2(t-1))$ of player \mathcal{P}_2 , player \mathcal{P}_1 constructs a forecast $f_2(t)$ which can be interpreted as a probability distribution on the anticipated $a_2(t)$. Similarly, player \mathcal{P}_2 creates a forecast, $f_1(t)$, of $a_1(t)$. The action of each player is then to maximize its expected utility based on this forecast, i.e.,

$$a_1(t) = \max_{a_1 \in \mathcal{A}_1} \mathbf{E}_{a_2(t) \sim f_2(t)} [U_1(a_1, a_2(t))]$$

$$a_2(t) = \max_{a_2 \in \mathcal{A}_2} \mathbf{E}_{a_1(t) \sim f_1(t)} [U_2(a_2, a_1(t))].$$

We use the notation \sim to denote the probability distribution over which the expectation operation is performed. Of course, this procedure relies heavily on how players forecast each other’s actions. The forecasting approach in (Mannor *et al.* 2006) is one of ‘calibrated forecasting’. In brief, calibrated forecasts are a ‘universal’ approach to construct statistically consistent (i.e., calibrated) forecasts of a discrete sequence in the absence of a model of the generator of the sequence (Foster and Vohra 1997). Unfortunately, these forecasting schemes, while universal, are not computationally feasible. In (Mannor *et al.* 2006), universality is sacrificed for computational efficiency. An alternative forecasting scheme, termed ‘tracking forecast’, is constructed that results in calibrated forecasts for special classes of sequence generators. This special class include many forms of learning algorithms. The final result is the ability in repeated play to optimally respond to an opponent based on observations of past play but without knowledge of the opponent’s utility function, and hence, intent.

1.2.2.3 Effects of information patterns

Chapter 12¹² considers a very different problem in adversarial interactions that is at the heart of the distributed information patterns prevalent in cooperative control problems. The problem set-up one of linear systems with a combination of stochastic and deterministic inputs. The system dynamics are of the form

$$\dot{x} = Ax + B_0d + B_1w + B_2u$$

$$y = Cx + n$$

The inputs d and n represent stochastic process noise and measurement noise, respectively. The inputs w and u are opposing adversarial inputs, where w seeks to maximize

¹² A. Swarup and J.L. Speyer, ‘Characterization of LQG differential games with different information patterns’.

a specified cost function that u seeks to minimize. The control u has access to the output measurement y . A standard assumption in worst case paradigms such as robust control is that the adversary, w , being ‘nature’, is all knowing. That is, it has full access to the state and to the measurement of the controller. Chapter 12 departs from this set-up by assuming that the maximizer, w , can measure the state x but *not* the noisy output y . It turns out that such seemingly innocuous differences can significantly alter the resulting constructions of optimal controllers. (A classic illustration is the Witsenhausen example (Ho 1980; Witsenhausen 1968).) For example, under standard assumptions, optimal controllers are of the same dimension as the state dynamics. Under non-standard assumptions, such as the above setting, it is not known whether optimal controllers are even finite dimensional, let alone the same order as the state dynamics. Chapter 12 shows that under mild technical assumptions, controllers remain finite dimensional even in this non-standard information scenario.

1.2.3 Uncertain evolution

An operating environment, even if not hostile, introduces significant uncertainty in the operation of autonomous vehicle systems. Uncertain evolution means that cooperative control solutions may encompass both estimation and adaptation methods to overcome environmental uncertainty.

1.2.3.1 Hybrid mode estimation

Let us take the position of an evading mobile target. The construction of an evasion trajectory would benefit from knowledge of the target assignments of vehicles (cf., the previous discussion on unknown opponent strategies and intent). Since there is no explicit communication between vehicles and targets, a target must estimate a vehicle’s assignment based on some sort of assumed model. In this setting, and for other cooperative control problems as well, such an underlying model often combines both continuous dynamics (e.g., vehicle motions) and discrete dynamics (e.g., target assignments), resulting in the now standard ‘hybrid’ system form.

The general assumed model is

$$\begin{aligned}x(t+1) &= A(q(t))x(t) + Bw(t) \\ y(t) &= Cx(t) + n(t)\end{aligned}$$

where w and n are stochastic process and measurement noise, respectively. The parameter $q(t)$ takes its values from a finite discrete set \mathcal{Q} . The problem of mode estimation is to infer $q(t)$ based on observations $y(t)$.

Chapters 13 and 14 provide complementary approaches to address this problem. Chapter 13¹³ takes an information theoretic approach. It is assumed that the mode switches, $q(t)$, are described by a Markov process with associated transition probabilities. The connection to information theory is that the mode history can be viewed as a ‘message’,

¹³ N.C. Martins and M.A. Dahleh, ‘Modal estimation of jump linear systems: an information theoretic viewpoint’.

and the system dynamics and observations can be viewed as a ‘channel’. The information theoretic connection allows one to provide bounds on mode estimation errors. Chapter 14¹⁴ takes a more explicit approach. The evolution dynamics of the mode is augmented onto the state dynamics, thereby producing a higher order nonlinear system. Approximate nonlinear filtering methods can then be applied to the augmented system to derive an online mode estimate.

1.2.3.2 Autonomous evolution of languages

The a long-term goal of cooperative control is to enable agents to explore unknown environments, communicate their findings to one another, and ultimately report back to human observers. A particularly ambitious goal is that this exploration will occur in highly unknown ‘first encounter’ environments. Such a scenario can be contrasted with a more standard search and rescue operation, where the uncertainty is much more structured. There is an unknown location to be specified, but the general environment of the search is well understood. By contrast, in first encounter environments, many basic elements of the environment are not known, and so agents must learn to communicate about findings that were not ‘pre-programmed’ into their communication protocol.

Chapter 15¹⁵ makes an interesting connection between this scenario and the evolution of languages. Different agents will encounter different discoveries, thereby altering their internal representations of the environment. The problem of language evolution involves determining whether initially homogenous languages will evolve into mutually unintelligible dialects or stay mutually understandable. The underlying graphical structure of communication among agents plays an important role in determining the ultimate outcome of this process. Interestingly, there are strong connections to the aforementioned consensus problem as well, cf. Equation (1.1).

1.2.4 Complexity management

Cooperative control problem formulations, if taken at face value, often result in computationally prohibitive solutions. Complexity management seeks to alleviate the inherent computational complexity through exploitation of special structures or the introduction of effective approximations.

Many of the problems discussed have some element of complexity management by virtue of an imposed distributed structure. For example, much of the task assignment work involved deriving effective solutions while bypassing an explicit optimization of the assignment problem. Chapters 16–17 take a more explicit view at complexity management.

Chapter 16¹⁶ revisits the problem of hybrid mode estimation, but for multiple agents. A motivating problem is estimating a *collective* assignment profile through observations of a group of vehicles. A complementary approach to stochastic methods in estimation

¹⁴ D. Choukroun and J.L. Speyer, ‘Conditionally-linear filtering for mode estimation in jump-linear systems’.

¹⁵ Y. Lee, T.C. Collier, C.E. Taylor, and E.P. Stabler, ‘Cohesion of languages in grammar networks’.

¹⁶ D. Del Vecchio and R.M. Murray, ‘Complexity management in the state estimation of multi-agent systems’.

is set-valued estimation (e.g., Milanese and Vicino 1991; Schweppe 1973; Shamma and Tu 1997, 1999). The main idea is to compute the entire *set* of values (as opposed to a single estimate) of an unknown quantity that are consistent with available observations to date. For example, for linear systems with bounded process disturbances and measurement noises, this amounts to constructing the set of all possible states based on output measurements. Not surprisingly, the exact characterization of such a set in the absence of any additional structure can be computationally prohibitive. Chapter 16 addresses this problem by superimposing a lattice structure on the state dynamics. The benefit of this lattice structure is that set-valued estimates can be expressed as generalized intervals, as in $lb(t) \leq x(t) \leq ub(t)$, where $lb(t)$ and $ub(t)$ denote lower bounds and upper bounds from the lattice structure, respectively. As new measurements are taken, these lower and upper bounds are recomputed. The approach is demonstrated on the problem of estimating an assignment profile based on vehicle motions. This also falls under the heading of hybrid mode estimation, but without a stochastic structure imposed on mode dynamics.

Chapter 17¹⁷ discusses complexity management from the perspective of scenario representation. The objective is to derive representations of a multivehicle scenario for a human operator that are both descriptive and not overly detailed. The challenge is to find an appropriate level of granularity and to represent this level of detail in a graphically appealing manner. The specific framework taken here is abstraction-based patch models. Rather than specify locations and trajectories of vehicles, patch models specify sets of possibilities. The tradeoff is that the omission of details results in the admission of auxiliary scenarios. By having a variable level of resolution, auxiliary scenarios can be pruned. In many ways, the framework resembles the set-valued estimation of Chapter 16. The main difference is that the set of possibilities is deliberately constructed as a representation device as opposed to the outcome of an estimation process.

1.3 FUTURE DIRECTIONS

We conclude this opening chapter with some comments on future directions in cooperative control. It should be clear from the present discussion that the scope of cooperative control admits multiple problem formulations for addressing multiple aspects of the underlying issues. And so, rather than suggest specific open questions, we will offer some general directions that have received relatively limited attention.

- *Strategic decision-making*: Much of the effort in cooperative control has been devoted to multivehicle trajectory planning and motion execution, such as formation keeping, rendezvous, or area coverage. Less attention has been given to *strategic* decision-making. Strategic decision-making entails determining, coordinating and executing a plan of attack in response to real-time data as events unfold. In terms of a feedback hierarchy, strategic decision-making involves higher level planning, while multivehicle motion planning is at a lower level (with conventional vehicular control at the lowest level). Some of the work in vehicle-task assignment falls under strategic decision-making, but in more general settings, one may not have an underlying geometric framework to exploit.

¹⁷ V. G. Rao, S. Goldfarb and R. D'Andrea, 'Abstraction-based command and control with patch models'.

- *Increasing role of game theory*: The use of game theory in cooperative control has largely been limited to the concept of Nash equilibrium in zero sum or general sum games. Game theory, by definition, is concerned with the interactions between multiple decision-makers, and hence it is clearly relevant to cooperative control. There are a variety of important concepts, such as mechanism design, bargaining, coalition theory, and correlated equilibrium that are not yet widely known. A challenge in exploiting these concepts is that they were developed with the intention of being models of social phenomena, whereas cooperative control applications involve engineered systems and require prescriptive synthesis tools.
- *Benchmark scenarios and problems*: A collection of benchmark scenarios would help to compare different approaches in the broad scope of cooperative control research. The terminology benchmark ‘scenario’ rather than benchmark ‘problem’ is deliberate here. For example, we have used vehicle target assignment as a benchmark scenario for much of the discussion in this chapter. Various research results were discussed in terms of the relevant component of this benchmark scenario. Benchmark problems are much more structured, where competing methodologies can be compared quantitatively. Benchmark problems, stemming from benchmark scenarios would also be very valuable. Here, one faces the usual challenge of defining specific problems that are neither trivially concocted nor intractable.

ACKNOWLEDGEMENTS

Research supported by AFOSR/MURI grant #F49620-01-1-0361, NSF grant #ECS-0501394, and ARO grant #W911NF-04-1-0316.

REFERENCES

- Ahuja RK, Kumar A, Jha K and Orlin JB 2003 Exact and heuristic methods for the weapon-target assignment problem. Technical Report #4464-03, MIT, Sloan School of Management Working Papers.
- Arslan G, Marden JR and Shamma JS 2007 Autonomous vehicle-target assignment : A game theoretical formulation. *ASME Journal of Dynamic Systems, Measurement and Control* **129**(5) 584–596.
- Beard RW, McLain TW, Goodrich MA and Anderson EP 2002 Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation* **18**(6), 911–922.
- Bemporad A and Morari M 1999 Robust model predictive control: A survey. In *Robustness in Identification and Control* (ed. Garully A, Tesi A and Vicino A), vol. **245**. Springer-Verlag, Berlin, Lecture Notes in Control and Information Sciences.
- Blondel VD, Hendrickx JM, Olshevsky A and Tsitsiklis J 2005 Convergence in multiagent coordination, consensus, and flocking. In *44th IEEE Conf. on Decision and Control*, pp. 2996–3000.
- Earl M and D’Andrea R 2002 A study in cooperative control: The Roboflag drill. In *Proceedings of the American Control Conference*, Anchorage, Alaska.
- Foster D and Vohra R 1997 Calibrated learning and correlated equilibrium. *Games and Economic Behavior* **21**, 40–55.
- Fudenberg D and Levine D 1998 *The Theory of Learning in Games*. MIT Press, Cambridge, MA.

- Ho YC 1980 Team decision theory and information structures. *Proceedings of the IEEE* **68**(6), 644–654.
- Jadbabaie A, Lin J and Morse AS 2003 Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control* **48**(6), 988–1001.
- Kashyap A, Basar T and Srikant R 2006 Consensus with quantized information updates. In *45th IEEE Conference on Decision and Control*, pp. 2728–2733.
- Mannor S, Shamma JS and Arslan G 2007 Online calibrated forecasts: Memory efficiency versus universality for learning in games. *Machine Learning*. Special issue on Learning and Computational Game Theory **129**(5) 585–596.
- Marden JR, Arslan G and Shamma JS 2007 Connections between cooperative control and potential games illustrated on the consensus problem. In *Proceedings of the European Control Conference*, 4604–4611.
- Milanese M and Vicino V 1991 Optimal estimation theory for dynamic systems with set membership uncertainty: An overview. *Automatica* **27**, 997–1009.
- Moreau L 2004 Stability of continuous-time distributed consensus algorithms. In *43rd IEEE Conference on Decision and Control*, pp. 3998–4003.
- Murphey R 1999 Target-based weapon target assignment problems. In *Nonlinear Assignment Problems: Algorithms and Applications* (ed. Pardalos P and Pitsoulis L). Kluwer Academic Publishers, Dordrecht, pp. 39–53.
- Murray R 2006 Recent research in cooperative control of multi-vehicle systems. Submitted to *ASME Journal of Dynamic Systems, Measurement, and Control*.
- Olfati-Saber R and Murray RM 2004 Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control* **AC-49**(9), 1520–1533.
- Olfati-Saber R, Fax JA and Murray RM 2007 Consensus and cooperation in networked multi-agent systems. In *Proceedings of the IEEE* **95**(1), 215–233.
- Panai L and Luke S 2005 Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems* **11**, 387–434.
- Schweppe F 1973 *Uncertain Dynamic Systems*. Prentice-Hall, Englewood Cliffs, NJ.
- Shamma J and Tu KY 1997 Approximate set-valued observers for nonlinear systems. *IEEE Transactions on Automatic Control* **AC-42**(5), 648–658.
- Shamma J and Tu KY 1999 Set-valued observers and optimal disturbance rejection. *IEEE Transactions on Automatic Control* **AC-44**(2), 253–264.
- Tsitsiklis JN, Bertsekas DP and Athans M 1986 Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control* **35**(9), 803–812.
- Witsenhausen H 1968 A counterexample in stochastic optimum control. *SIAM Journal of Control* **6**(1), 131–147.
- Wolpert D and Tumer K 1999 An overview of collective intelligence In *Handbook of Agent Technology* (ed. Bradshaw JM) AAAI Press/MIT Press, Cambridge, MA.
- Xiao L and Boyd S 2004 Fast linear iterations for distributed averaging. *Systems and Control Letters* **53**(1): 65–78.
- Xiao L and Boyd S 2005 A scheme for robust distributed sensor fusion based on average consensus. In *Information Processing in Sensor Networks*, pp. 63–70.
- Young H 2006 *Strategic Learning and its Limits*. Oxford University Press, Oxford.