1

Complexity management in the state estimation of multi-agent systems

Domitilla Del Vecchio[1] and Richard M. Murray[2]

University of Michigan[1], California Institute of Technology[2]

Abstract: This work addresses the problem of estimating the state in multi-agent decision and control systems. In particular, a novel approach to state estimation is developed that uses partial order theory in order to overcome some of the severe computational complexity issues arising in multi-agent systems. Within this approach, state estimation algorithms are developed, which enjoy proved convergence properties and are scalable with the number of agents. The dynamic evolution of the systems under study are characterized by the interplay of continuous and discrete variables. Continuous variables usually represent physical quantities such as position, velocity, voltage, and current, while the discrete variables usually represent quantities internal to the decision protocol that is used for coordination, communication, and control. Within the proposed state estimation approach, the estimation of continuous and discrete variables is developed in the same mathematical framework, as a joint continuous-discrete space is considered for the estimator. An application example is considered, involving the state estimation in competitive multi-robot systems.

Cooperative Control of Distributed Multiagent Systems. Edited by J.S. Shamma

This is a Book Title **Name of the Author/Editor**

1.1 Introduction

Logic and decision making are playing increasingly large roles in modern control systems, and virtually all modern control systems are implemented using digital computers. Examples include aerospace systems, transportation systems (air, automotive, and rail), communication networks (wired, wireless, and cellular), and supply networks (electrical power and manufacturing). The evolution of these systems is determined by the interplay of continuous dynamics and logic. The continuous variables can represent quantities such as position, velocity, acceleration, voltage, current, etc., while the discrete variables can represent the state of the decision and communication protocol that is used for coordination and control. Most of these systems are also multi-agent, in which an agent can be, for example, a wireless device, a micro-controller, a robot, a piece of machinery, a piece of hardware or software, or even a human. The need for understanding and analyzing the behavior of these systems is compelling. However, the coupling of continuous dynamics and logics and the multi-agent nature of these systems render the study of these systems interesting and complicated enough that new tools are needed for the sake of analysis and control. In particular, multi-agent systems are usually affected by the combinatorial explosion of the state space that renders most of the existing state estimation algorithms inapplicable.

The problem of estimating the state of a decision and control system has been addressed by several authors for control or as a means for solving monitoring or surveillance problems in distributed environments. In the hybrid systems literature, Bemporad et al. (1999) propose the notion of incremental observability for piecewise affine systems and construct a deadbeat observer that requires large amounts of computation. Balluchi et al. (2002) combine a *location* observer with a Luenberger observer to design hybrid observers that identify the location in a finite number of steps and converge exponentially to the continuous state. However, if the number of locations is large, as in the systems that we consider, such an approach is impracticable. In Balluchi et al. (2003), sufficient conditions for a linear hybrid system to be final state determinable are given. In Alessandri and Coletta (2001, 2003), Luenberger-like observers are proposed for hybrid systems where the system location is known. Vidal et al. (2002) derive sufficient and necessary conditions for observability of discrete time jump-linear systems, based on a simple rank test on the parameters of the model. In later work (Vidal et al. (2003)), these notions are generalized to the case of continuous time jump linear systems. For jump Markov linear systems, Costa and do Val (2002) derive a test for observability, and Cassandra et al. (1994) propose an approach to optimal control for partially observable Markov decision processes. For continuous time hybrid systems, Santis et al. (2003) propose a definition of observability based on the possibility of reconstructing the system state, and testable conditions for observability are provided.

In the discrete event literature, observability has been defined by Ramadge (1986), for example, who derives a test for current state observability. Oishi et al. (2003) derive a test for immediate observability in which the state of the system can be unambiguously reconstructed from the output associated with the current state and last and next events. Ozveren and Willsky (1990), Caines et al. (1991), and Caines and Wang (1995) propose discrete event observers based on the construction of the current-location observation tree that is impracticable when the number of locations is large, which is our case. Observability is also considered in the context of distributed monitoring and control in industrial automation, where agents are cooperating to perform system-level tasks such as failure detection and identification on the basis of local information (Rudie et al. (2003)). Diaz et al. (1994) consider observers for formal on-line validation of distributed systems, in which the on-line behavior is checked against a formal model. In the context of sensor networks, state estimation covers a fundamental role when solving surveillance and monitoring tasks in which the state usually has several components, such as the position of an agent, its identity, and its intent (see for example Collins et al. (2001) or Bui et al. (2002)).

The main contribution of this work is to design state estimators for multi-agent systems that overcome severe complexity issues encountered in previous work (Balluchi et al. (2002); Caines and Wang (1995); Caines et al. (1991)). These complexity issues render prohibitive the estimation problem for systems with a large discrete state space, which is often the case in multi-agent systems. Our point of view is that some of the complexity issues, such as those encountered in Caines et al. (1991) or Balluchi et al. (2002), can be avoided by finding a good way of representing the sets of interest and by finding a good way of computing maps on them. In this work, this is achieved by using partial order theory. Partial order theory has been historically used in theoretical computer science to prove properties about convergence of algorithms (Cousot and Cousot (1977)). It has also been used for studying controllability properties of finite state machines (Caines and Wei (1996)) and for tackling the state explosion problem in the verification of concurrent systems in Godefroid (1996). In this work, we exploit partial order theory to estimate the state in systems with a large discrete space resulting from the multi-agent nature of the system. In particular, given a system Σ defined on its space of variables, we extend it to a larger space of variables that has lattice structure to obtain an extended system $\tilde{\Sigma}$. Under certain properties verified by the extension $\tilde{\Sigma}$, an observer for system Σ can be constructed, which updates at each step only two variables. It updates the least and greatest element of the set of all values of variables compatible with the output sequence and with the dynamics of Σ. The structure of the obtained observer resembles the structure of the Luenberger observer (Luenberger (1971)) or a Kalman filter (Kalman (1960)) as it is obtained by "copying" the dynamics of the system Σ and by correcting it according to the measured output values. This work is concerned with the estimation of the discrete state in case the continuous state is measured, and with the estimation of the whole system state in case a cascade structure of the estimator is possible. We also show that a system is observable if and only if there is a lattice in which the extended system satisfies the requirements for the construction of the proposed estimator. Thus, our approach to state estimation is general.

The contents of this work are organized as follows. In Section 1.2, the RoboFlag Drill is introduced as our motivating example. In Section 1.3, basic definitions on partial orders and transition systems are reviewed. Section 1.4 formulates the discrete state estimation problem and Section 1.5 proposes a solution. In Section 1.6, the RoboFlag Drill is revisited. In Section 1.7, the generality of the approach is investigated and in Section 1.8 the proposed state estimation approach is extended to estimation of continuous and discrete variables for the case of a cascade form of the estimator.

1.2 Motivating Example

As a motivating example, we consider a task that represents a defensive maneuver for a robotic "capture the flag" game (D'Andrea et al. (2003)). We do not propose to devise a strategy that addresses the full complexity of the game. Instead, we examine the following very simple *drill* or exercise that we call “RoboFlag Drill.” Some number of blue robots with positions $(z_i, 0) \in \mathbb{R}^2$ (denoted by open circles) must defend their zone $\{(x, y) \in \mathbb{R}^2 \mid y \leq 0\}$ from an equal number of incoming red robots (denoted by filled circles). The positions of the red robots are $(x_i, y_i) \in \mathbb{R}^2$. An example for 8 robots is illustrated in Figure 1.1. The red robots move straight toward the blue robots’ defensive zone. The blue robots are each assigned to a red robot, and they coordinate to intercept the red robots. Let N represent the number of robots in each team. The robots start with an arbitrary (bijective) assignment $\alpha : \{1, ..., N\} \rightarrow \{1, ..., N\}$, where α_i is the red robot that blue robot i is required to intercept. At each step, each blue robot communicates with its neighbors and decides to either switch assignments with its left or right neighbor or keep its assignment. It is possible to show that the α assignment reaches the equilibrium value $(1, ..., N)$ (see Klavins and Murray (2004) or Klavins (2003) for details). We consider the problem of estimating the current assignment α given the motions of the blue robots, which might be of interest to, for example, the red robots in that they may use such information to determine a better strategy of attack. We do not consider the problem of how they would change their strategy in this work.

The RoboFlag Drill system can be specified by the following rules:

$$y_i(k+1) = y_i(k) - \delta \text{ if } y_i(k) \geq \delta \tag{1.1}$$

$$z_i(k+1) = z_i(k) + \delta \text{ if } z_i(k) < x_{\alpha_i(k)} \tag{1.2}$$

$$z_i(k+1) = z_i(k) - \delta \text{ if } z_i(k) > x_{\alpha_i(k)} \tag{1.3}$$

$$(\alpha_i(k+1), \alpha_{i+1}(k+1)) = (\alpha_{i+1}(k), \alpha_i(k)) \text{ if } x_{\alpha_i(k)} \geq z_{i+1}(k) \wedge x_{\alpha_{i+1}(k)} \leq z_{i+1}(k), \tag{1.4}$$

where we assume $z_i \leq z_{i+1}$ and $x_i < z_i < x_{i+1}$ for all k. Also, if none of the “if” statements above are verified for a given variable, the new value of the variable is equal to the old one. This system is a slight simplification of the original system described in Klavins (2003). In such a work in fact, two close robots might decide to swap their assignments even if they are moving in the same direction, while in the present case, two close robots swap their assignments only if they are moving one toward the other. Also, in Klavins (2003) the decision are taken sequentially first from the robots on the left and then from the robots on the right, and the decision are coordinated by a token that moves from left to right. In the present case, the decision protocol is completely decentralized. Equation (1.4) establishes that two robots trade their assignments if the current assignments cause them to go toward each other. The question we are interested in is the following: given the evolution of the measurable quantities z, x, y, can we build an estimator that tracks on-line the value of the assignment $\alpha(k)$? The value of $\alpha \in \text{perm}(N)$ determines the discrete state, i.e., $S = \text{perm}(N)$. The discrete state α determines also what has been called in previous work the location of the system (see Balluchi et al. (2002)). The number of possible locations is $N!$, that is, $|S| = N!$. This for $N \geq 8$ renders prohibitive the application of location observers based on the current location observation tree of Caines et al. (1991), used in Balluchi et al. (2002), and in Ozveren and Willsky (1990). At each step, the set of possible α values compatible with the current output and with the previously seen outputs can be so large as to render impractical its computation. As an example, we consider the situation depicted in Figure 1.1 (left) where $N = 8$. We see the blue robots $1, 3, 5$ going right and the others going left. From equations (1.2)–(1.3) with $x_i < z_i < x_{i+1}$ we deduce that the set of all possible $\alpha \in \text{perm}(N)$ compatible with this observation is such that $\alpha_i \geq i + 1$ for $i \in \{1, 2, 3\}$ and $\alpha_i \leq i$ for $i \in \{2, 4, 6, 7, 8\}$.

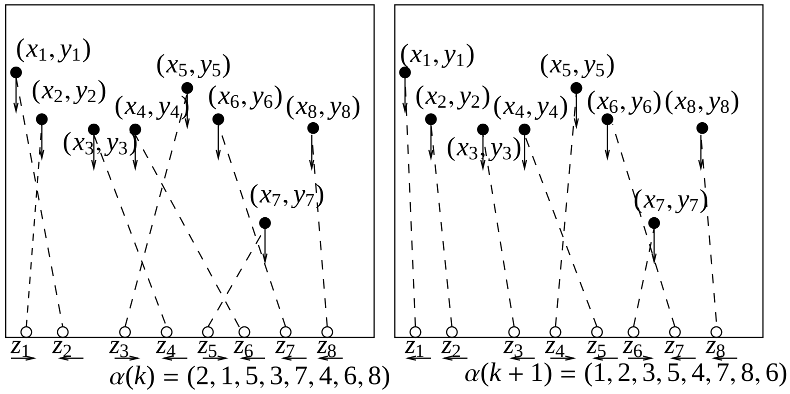

Figure 1.1 Example of the RoboFlag Drill with 8 robots per team. The dashed lines represent the assignment of each blue robot to red robot. The arrows denote the direction of motion of each robot.

The size of this set is 40320. According to the enumeration methods, this set needs to be mapped forward through the dynamics of the system to see what the values of α are at the next step that correspond to this output. Such a set is then intersected with the set of α values compatible with the new observation. To overcome the complexity issue that comes from the need of listing 40320 elements for performing such operations, we propose to represent a set by a lower L and an upper U elements according to some partial order. Then, we can perform the previously described operations only on L and U, two elements instead of 40320. This idea is developed in the following paragraph.

For this example, we can view $\alpha \in \mathbb{N}^N$. The set of possible assignments compatible with the observation of the z motion deduced from the equations (1.2)–(1.3), denoted $O_y(k)$, can be represented as an interval with the order established component-wise, see the diagram in Figure 1.2. The function $\tilde{f}$ that maps such a set forward, specified by the equations (1.4) with the assumption that $x_i < z_i < x_{i+1}$, simply swaps two adjacent robot assignments if these cause the two robots to move toward each other. Thus, it maps the set $O_y(k)$ to the set $\tilde{f}(O_y(k))$ shown in Figure 1.2, which can still be represented as an interval. When the new output measurement becomes available (Figure 1.1, right) we obtain the new set $O_y(k+1)$ reported in Figure 1.2. The sets $\tilde{f}(O_y(k))$ and $O_y(k+1)$ can be intersected by simply computing the supremum of their lower bounds and the infimum of their upper bounds. This way, we obtain the system that updates L and U, being L and U the lower and upper bounds of the set of all possible α compatible with the output sequence:

$$
\begin{aligned}
L(k+1) &= \tilde{f}\left(\sup(L(k), \inf O_y(k))\right) \\
U(k+1) &= \tilde{f}\left(\inf(U(k), \sup O_y(k))\right).
\end{aligned}
\tag{1.5}
$$

The variables $L(k)$ and $U(k)$ represent the lower and upper bound, respectively, of the set of all possible discrete state values compatible with the output sequence and with the system dynamics. The computational burden of this implementation is of the order of N if N is the number of robots. This computational burden is to be compared to $N!$, which is the computation requirement that we have with the enumeration approach. In the next section,

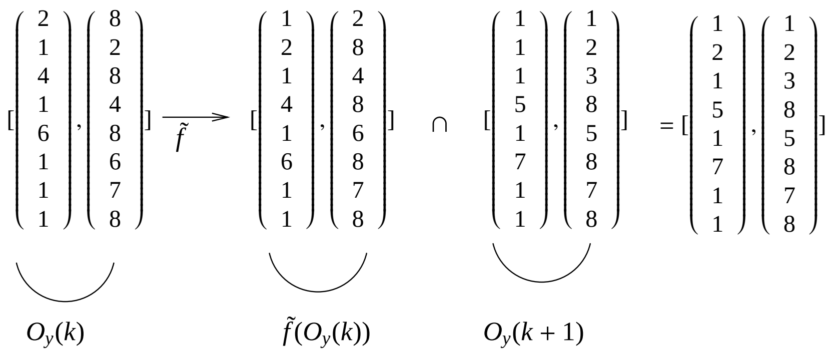

Figure 1.2 The observation of the z motion at step k gives the set of possible α, $O_y(k)$. At each step, the set is described by the lower and upper bounds of an *interval sublattice* in an appropriately defined lattice. Such set is then mapped through the system dynamics $\tilde{f}$ to obtain at step $k+1$ the set of α that are compatible also with the observation at step k. Such a set is then intersected with $O_y(k+1)$, which is the set of α compatible with the z motion observed at step $k+1$.

we introduce some basic notions on partial order theory and deterministic transition system models.

1.3 Basic Concepts

In this chapter, we review some basic notions that will be used throughout this work. First, we give some background on partial order and lattice theory in Section 1.3.1 (for more details the reader is referred to Davey and Priesteley (2002)). The theory of partial orders, while standard in computer science, may be less well known to the intended audience of this work. The class of deterministic transition systems is introduced in Section 1.3.2.

1.3.1 Partial Order Theory

A partial order is a set χ with a partial order relation "$\leq$", and we denote it by the pair $(\chi, \leq)$. For any $x, w \in \chi$, $\sup\{x, w\}$ is the smallest element that is larger than both x and w. In a similar way, $\inf\{x, w\}$ is the largest element that is smaller than both x and w. We define the *join* "$\vee$" and the *meet* "$\wedge$" of two elements x and w in χ as $x \vee w := \sup\{x, w\}$ and $x \wedge w := \inf\{x, w\}$ Also, if $S \subseteq \chi$, we have $\bigvee S := \sup\ S$, and $\bigwedge S := \inf\ S$.

Let $(\chi, \leq)$ be a partial order. If $x \wedge w \in \chi$ and $x \vee w \in \chi$ for any $x, w \in \chi$, then $(\chi, \leq)$ is a *lattice*. In Figure 1.3, we illustrate Hasse diagrams (Davey and Priesteley (2002)) showing partially ordered sets. From the diagrams, it is easy to tell when one element is less than another: $x < w$ if and only if there is a sequence of connected line segments moving upward from x to w. The partial order $(\chi, \leq)$ is a *chain* if for all $x, w \in \chi$, either $x \leq w$ or $w \leq x$, that is, any two elements are comparable. If instead any two elements are not comparable, i.e., $x \leq y$ if and only if $x = y$, $(\chi, \leq)$ is said to be an *anti-chain*. If $x < w$ and there is no other element in

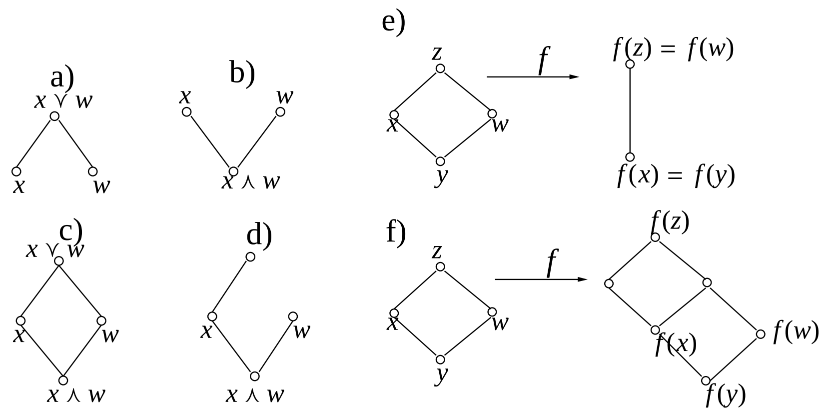

Figure 1.3 Left figure: In diagram a) and b), x and w are not related, but they have a join and a meet, respectively; in diagram c), we show a complete lattice; in diagram d), we show a partially ordered set that is not a lattice, since the elements x and w have a meet, but not a join. Right figure: In diagram e), we show a map that is, order preserving but not order embedding; in diagram f), we show an order embedding that is, not an order isomorphism (it is not onto).

between x and w, we write $x \ll w$. Let $(\chi, \leq)$ be a lattice and let $S \subseteq \chi$ be a non-empty subset of χ. Then, $(S, \leq)$ is a *sublattice* of χ if $a, b \in S$ implies that $a \vee b \in S$ and $a \wedge b \in S$. If any sublattice of χ contains its least and greatest elements, then $(\chi, \leq)$ is called *complete*. Any finite lattice is complete, but infinite lattices may not be complete, and hence the significance of the notion of a complete partial order (S. Abramsky (1994)). Given a complete lattice $(\chi, \leq)$, we will be concerned with a special kind of a sublattice called an *interval sublattice* defined as follows. Any interval sublattice of $(\chi, \leq)$ is given by $[L, U] = \{w \in \chi | L \leq w \leq U\}$ for $L, U \in \chi$. That is, this special sublattice can be represented by two elements only. For example, the interval sublattices of $(\mathbb{R}, \leq)$ are just the familiar closed intervals on the real line.

Let $(\chi, \leq)$ be a lattice with least element $\perp$ (the bottom). Then, $a \in \chi$ is called an *atom* of $(\chi, \leq)$ if $a > \perp$ and there is no element b such that $\perp < b < a$. The set of atoms of $(\chi, \leq)$ is denoted $\mathcal{A}(\chi, \leq)$. The *power lattice* of a set $\mathcal{U}$, denoted $(\mathcal{P}(\mathcal{U}), \subseteq)$, is given by the power set of $\mathcal{U}$, $\mathcal{P}(\mathcal{U})$ (the set of all subsets of $\mathcal{U}$), ordered according to the set inclusion $\subseteq$. The meet and join of the power lattice is given by intersection and union. The bottom element is the empty set, that is, $\perp = \emptyset$, and the top element is $\mathcal{U}$ itself, that is, $\top = \mathcal{U}$. Note that $\mathcal{A}(\mathcal{P}(\mathcal{U}), \subseteq) = \mathcal{U}$. Given a set P, we denote by $|P|$ its cardinality.

Definition 1.3.1 Let $(P, \leq)$ and $(Q, \leq)$ be partially ordered sets. A map $f : P \to Q$ is

(i) an *order preserving map* if $x \leq w \implies f(x) \leq f(w)$;

(ii) an *order embedding* if $x \leq w \iff f(x) \leq f(w)$;

(iii) an *order isomorphism* if it is order embedding and it maps P *onto* Q.

These different types of maps are shown in diagrams e) and f) of Figure 1.3. Every order isomorphism faithfully mirrors the structure of P onto Q.

Definition 1.3.2 If $(P, \leq)$ and $(Q, \leq)$ are lattices, then a map $f : P \to Q$ is said to be a *homomorphism* if f is *join-preserving* and *meet-preserving*, that is, for all $x, w \in P$ we have that $f(x \vee w) = f(x) \vee f(w)$ and $f(x \wedge w) = f(x) \wedge f(w)$.

Proposition 1.3.3 *(See Davey and Priesteley (2002)) If $f : P \to Q$ is a bijective homomorphism, then it is an order isomorphism.*

A partial order induces a notion of distance between elements in the space. Define the distance function on a partial order in the following way.

Definition 1.3.4 (Distance on a partial order) Let $(P, \leq)$ be a partial order. A distance d on $(P, \leq)$ is a function $d : P \times P \to \mathbb{R}$ such that the following properties are verified:

(i) $d(x, y) \geq 0$ for any $x,\ y \in P$ and $d(x, y) = 0$ if and only if $x = y$;

(ii) $d(x, y) = d(y, x)$;

(iii) if $x \leq y \leq z$ then $d(x, y) \leq d(x, z)$;

(iv) $d(x, z) \leq d(x, y) + d(y, z)$ (triangular inequality).

Since we will deal with a partial order on the space of the discrete variables and with a partial order on the space of the continuous variables, it is useful to introduce the Cartesian product of two partial orders as it can be found in S. Abramsky (1994).

Definition 1.3.5 (Cartesian product of partial orders) Let $(P_1, \leq)$ and $(P_2, \leq)$ be two partial orders. Their Cartesian product is given by $(P_1 \times P_2, \leq)$, where $P_1 \times P_2 = \{(x, y) \mid x \in P_1 \text{ and } y \in P_2\}$, and $(x, y) \leq (x', y')$ if and only if $x \leq x'$ and $y \leq y'$. For any $(p_1, p_2) \in P_1 \times P_2$ the standard projections $\pi_1 : P_1 \times P_2 \to P_1$ and $\pi_2 : P_1 \times P_2 \to P_2$ are such that $\pi_1(p_1, p_2) = p_1$ and $\pi_2(p_1, p_2) = p_2$.

One can easily verify that the projection operators preserve the order. In this work we will also deal with *approximations* of sets and elements of a partial order. We thus give the following definition.

Definition 1.3.6 (Upper and lower approximation) Let P_1 and P_2 be two sets with $P_1 \subseteq P_2$ and $(P_2, \leq)$ a partial order. For any $x \in P_2$, we define the *lower and upper approximations* of x in P_1 as $a_L(x) := \max_{(P_2,\leq)}\{w \in P_1 \mid w \leq x\}$ and $a_U(x) := \min_{(P_2,\leq)}\{w \in P_1 \mid w \geq x\}$. If such lower and upper approximations exist for any $x \in P_2$, then the partial order $(P_2, \leq)$ is said to be *closed with respect to P_1*.

One can verify that the lower and upper approximation functions are order preserving. This means that for any $x_1, x_2 \in P_2$ with $x_1 \leq x_2$, then $a_L(x_1) \leq a_L(x_2)$ and $a_U(x_1) \leq a_U(x_2)$. In this section, we have given some basic definitions on partial order and lattice theory. In the next section, we introduce the class of models that we are going to consider in this work. These are transition systems with output.

1.3.2 Deterministic Transition Systems

The class of systems we are concerned with are deterministic, infinite state systems with output. The following definition introduces such a class.

Definition 1.3.7 (Deterministic transition systems) A *deterministic transition system* (DTS) is the tuple $\Sigma = (S, \mathcal{Y}, F, g)$, where S is a set of states with $s \in S$; $\mathcal{Y}$ is a set of outputs with $y \in \mathcal{Y}$; $F : S \to S$ is the state transition function; $g : S \to \mathcal{Y}$ is the output function.

An execution of Σ is any sequence $\sigma = \{s(k)\}_{k\in\mathbb{N}}$ such that $s(0) \in S$ and $s(k+1) = F(s(k))$ for all $k \in \mathbb{N}$. The set of all executions of Σ is denoted $\mathcal{E}(\Sigma)$. An output sequence of Σ is denoted $y = \{y(k)\}_{k\in\mathbb{N}}$, with $y(k) = g(\sigma(k))$, for $\sigma \in \mathcal{E}(\Sigma)$.

Definition 1.3.8 Given a deterministic transition system $\Sigma = (S, \mathcal{Y}, F, g)$, two executions σ_1, σ_2 in $\mathcal{E}(\Sigma)$ are *distinguishable* if there exists a k such that $g(\sigma_1(k)) \neq g(\sigma_2(k))$.

Definition 1.3.9 (Observability) The deterministic transition system $\Sigma = (S, \mathcal{Y}, F, g)$ is said to be *observable* if any two different executions $\sigma_1, \sigma_2 \in \mathcal{E}(\Sigma)$ are distinguishable.

From this definition, we deduce that if a system Σ is observable, any two different initial states will give rise to two executions σ_1 and σ_2 with different output sequences. Thus, the initial states can be distinguished by looking at the output sequence.

1.4 Problem Formulation

The deterministic transition systems Σ we defined in the previous section are quite general. In this section, we restrict our attention to systems with a specific structure. In particular, for a system $\Sigma = (S, \mathcal{Y}, F, g)$ we suppose that (i) $S = \mathcal{U} \times \mathcal{Z}$ with $\mathcal{U}$ a finite set and $\mathcal{Z}$ a finite dimensional space; (ii) $F = (f, h)$, where $f : \mathcal{U} \times \mathcal{Z} \to \mathcal{U}$ and $h : \mathcal{U} \times \mathcal{Z} \to \mathcal{Z}$; (iii) $y = g(\alpha, z) := z$, where $\alpha \in \mathcal{U}$, $z \in \mathcal{Z}$, $y \in \mathcal{Y}$, and $\mathcal{Y} = \mathcal{Z}$. The set $\mathcal{U}$ is a set of logic states and $\mathcal{Z}$ is a set of measured states or physical states, as one might find in a robot system. In the case of the example given in Section 1.2, $\mathcal{U} = \text{perm}(N)$ and $\mathcal{Z} = \mathbb{R}^N$, the function f is represented by equations (1.4) and the function h is represented by equations (1.2)–(1.3). In the sequel, we will denote with abuse of notation this class of deterministic transition systems by $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$, in which we associate to the tuple $(\mathcal{U}, \mathcal{Z}, f, h)$, the equations:

$$\begin{aligned} \alpha(k+1) &= f(\alpha(k), z(k)) \\ z(k+1) &= h(\alpha(k), z(k)) \\ y(k) &= z(k), \end{aligned} \tag{1.6}$$

in which $\alpha \in \mathcal{U}$ and $z \in \mathcal{Z}$. An execution of the system Σ in equations (1.6) is a sequence $\sigma = \{\alpha(k), z(k)\}_{k\in\mathbb{N}}$. The output sequence is $\{y(k)\}_{k\in\mathbb{N}} = \{z(k)\}_{k\in\mathbb{N}}$. Given an execution σ of the system Σ, we denote the α and z sequences corresponding to such an execution by $\{\sigma(k)(\alpha)\}_{k\in\mathbb{N}}$ and $\{\sigma(k)(z)\}_{k\in\mathbb{N}}$, respectively.

From the measurement of the output sequence, which in our case coincides with the evolution of the continuous variables, we want to construct a discrete state estimator: a system $\hat{\Sigma}$ that takes as input the values of the measurable variables and asymptotically tracks the value of the variable α. We thus define in the following definition a deterministic transition system with input.

Definition 1.4.1 (Deterministic transition system with input) A deterministic transition system with input is a tuple $(S, \mathcal{I}, \mathcal{Y}, F, g)$ in which S is a set of states; $\mathcal{I}$ is a set of inputs; $\mathcal{Y}$ is a set of outputs; $F : S \times \mathcal{I} \to S$ is a transition function; $g : S \times \mathcal{I} \to \mathcal{Y}$ is an output function.

An alternative to simply maintaining a list of all possible values for α is next proposed. Specifically, if the set $\mathcal{U}$ can be immersed in a larger set χ whose elements can be related by an order relation $\leq$, we can represent a subset of $(\chi, \leq)$ as an interval sublattice $[L, U]$. Let "id" denote the identity operator. We formulate the discrete state estimation problem on a lattice as follows.

Problem 1 (Discrete state estimator on a lattice) Given the deterministic transition system $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$, find a deterministic transition system with input $\hat{\Sigma} = (\chi \times \chi, \mathcal{Y} \times \mathcal{Y}, \chi \times \chi, (f_1, f_2), \text{id})$, with $f_1 : \chi \times \mathcal{Y} \times \mathcal{Y} \to \chi$, $f_2 : \chi \times \mathcal{Y} \times \mathcal{Y} \to \chi$, $\mathcal{U} \subseteq \chi$, with $(\chi, \leq)$ a lattice, represented by the equations

$$L(k+1) = f_1(L(k), y(k), y(k+1))$$
$$U(k+1) = f_2(U(k), y(k), y(k+1)),$$

with $L(k) \in \chi$, $U(k) \in \chi$, $L(0) := \bigwedge \chi$, $U(0) := \bigvee \chi$, such that

(i) $L(k) \leq \alpha(k) \leq U(k)$ (correctness);

(ii) $|[L(k+1), U(k+1)]| \leq |[L(k), U(k)]|$ (non-increasing error);

(iii) There exists $k_0 > 0$ such that for any $k \geq k_0$ we have $[L(k), U(k)] \cap \mathcal{U} = \alpha(k)$ (convergence).

1.5 Problem Solution

For finding a solution to Problem 1, we need to find the functions f_1 and f_2 defined on a lattice $(\chi, \leq)$ such that $\mathcal{U} \subseteq \chi$ for some finite lattice χ. We propose in the following definitions a way of extending a system Σ defined on $\mathcal{U}$ to a system $\tilde{\Sigma}$ defined on χ with $\mathcal{U} \subseteq \chi$. Moreover, as we have seen in the motivating example, we want to represent the set of possible α values compatible with an output measurement as an interval sublattice in $(\chi, \leq)$. We thus define the $\tilde{\Sigma}$ transition classes, with each transition class corresponding to a set of values in χ compatible with an output measurement. We define the partial order $(\chi, \leq)$ and the system $\tilde{\Sigma}$ to be interval compatible if such equivalence classes are interval sublattices and $\tilde{\Sigma}$ preserves their structure.

Definition 1.5.1 (Extended system) Given the deterministic transition system $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$, an *extension of* Σ *on* χ, with $\mathcal{U} \subseteq \chi$ and $(\chi, \leq)$ a finite lattice, is any system $\tilde{\Sigma} = (\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ in which $\tilde{f} : \chi \times \mathcal{Z} \to \chi$ is such that $\tilde{f}|_{\mathcal{U}\times\mathcal{Z}} = f$ and $\tilde{h} : \chi \times \mathcal{Z} \to \mathcal{Z}$ is such that $\tilde{h}|_{\mathcal{U}\times\mathcal{Z}} = h$.

Definition 1.5.2 (Transition sets) Let $\tilde{\Sigma} = (\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ be a deterministic transition system. The non empty sets $T_{(z^1,z^2)}(\tilde{\Sigma}) = \{w \in \chi \mid z^2 = \tilde{h}(w, z^1)\}$, for $z^1, z^2 \in \mathcal{Z}$, are named the $\tilde{\Sigma}$-*transition sets*.

Each $\tilde{\Sigma}$-transition set contains all of $w \in \chi$ values that allow the transition from z^1 to z^2 through $\tilde{h}$. It will also be useful to define the transition class $\mathcal{T}_i(\tilde{\Sigma})$, which corresponds to multiple transition sets, as transition sets obtained by different pairs (z^1, z^2) can define the same set in χ.

Definition 1.5.3 (Transition classes) The set $\mathcal{T}(\tilde{\Sigma}) = \{\mathcal{T}_1(\tilde{\Sigma}), ..., \mathcal{T}_M(\tilde{\Sigma})\}$, with $\mathcal{T}_i(\tilde{\Sigma})$ such that

(i) for any $\mathcal{T}_i(\tilde{\Sigma}) \in \mathcal{T}(\tilde{\Sigma})$ there are $z^1, z^2 \in \mathcal{Z}$ such that $\mathcal{T}_i(\tilde{\Sigma}) = T_{(z^1,z^2)}(\tilde{\Sigma})$;

(ii) for any $T_{(z^1,z^2)}(\tilde{\Sigma})$ there is $j \in \{1, ..., M\}$ such that $T_{(z^1,z^2)}(\tilde{\Sigma}) = \mathcal{T}_j(\tilde{\Sigma})$;

is the *set of* $\tilde{\Sigma}$*-transition classes*.

Note that $T_{(z^1,z^2)}$ and $T_{(z^3,z^4)}$ might be the same set even if $(z^1, z^2) \neq (z^3, z^4)$: in the RoboFlag Drill example introduced in Section 1.2, if robot j is moving right, the set of possible values of α_j is $[j + 1, N]$ independently of the values of $z_j(k)$. Thus, $T_{(z^1,z^2)}$ and $T_{(z^3,z^4)}$ can define the same set that we call $\mathcal{T}_i(\tilde{\Sigma})$ for some i. Also, the transition classes $\mathcal{T}_i(\tilde{\Sigma})$ are not necessarily equivalence classes as they might not be pairwise disjoint. However, for the RoboFlag Drill it is the case that the transition classes are pairwise disjoint, and thus they partition the lattice $(\chi, \leq)$ in equivalence classes.

Definition 1.5.4 (Output set) Given the extension $\tilde{\Sigma} = (\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ of the deterministic transition system $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$ on the lattice $(\chi, \leq)$, and given an output sequence $\{y(k)\}_{k\in\mathbb{N}}$ of Σ, the set $O_y(k) := \{w \in \chi \mid \tilde{h}(w, y(k)) = y(k + 1)\}$ is the *output set* at step k.

Note that by definition, for any k, $O_y(k) = T_{(y(k),y(k+1))}(\tilde{\Sigma})$, and thus it is equal to $\mathcal{T}_i(\tilde{\Sigma})$ for some $i \in \{1, ..., M\}$. By definition of the extended functions ($\tilde{h}|_{\mathcal{U}\times\mathcal{Z}} = h$), this output set contains also all of the values of α compatible with the same output pair.

Definition 1.5.5 (Interval compatibility) Given the extension $\tilde{\Sigma} = (\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ of the system $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$ on the lattice $(\chi, \leq)$, the pair $(\tilde{\Sigma}, (\chi, \leq))$ is said to be *interval compatible* if

(i) each $\tilde{\Sigma}$-transition class, $\mathcal{T}_i(\tilde{\Sigma}) \in \mathcal{T}(\tilde{\Sigma})$, is an interval sublattice of $(\chi, \leq)$, that is, $\mathcal{T}_i(\tilde{\Sigma}) = [\bigwedge\mathcal{T}_i(\tilde{\Sigma}), \bigvee\mathcal{T}_i(\tilde{\Sigma})]$;

(ii) $\tilde{f} : (\mathcal{T}_i(\tilde{\Sigma}), z) \to [\tilde{f}(\bigwedge\mathcal{T}_i(\tilde{\Sigma}), z), \tilde{f}(\bigvee\mathcal{T}_i(\tilde{\Sigma}), z)]$ is an order isomorphism for any $i \in \{1, ..., M\}$ and for any $z \in \mathcal{Z}$.

The following theorem gives the main result, which proposes a solution to Problem 1.

Theorem 1.5.6 *Assume that the deterministic transition system* $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$ *is observable. If there is a lattice* $(\chi, \leq)$*, such that the pair* $(\tilde{\Sigma}, (\chi, \leq))$ *is interval compatible, then the deterministic transition system with input* $\hat{\Sigma} = (\chi \times \chi, \mathcal{Z} \times \mathcal{Z}, \chi \times \chi, (f_1, f_2), id)$ *with*

$$f_1(L(k), y(k), y(k + 1)) = \tilde{f}\left(L(k) \vee \bigwedge O_y(k), y(k)\right)$$
$$f_2(U(k), y(k), y(k + 1)) = \tilde{f}\left(U(k) \wedge \bigvee O_y(k), y(k)\right)$$

solves Problem 1.

Proof. In order to prove the statement of the theorem, we need to prove that the system

$$\begin{aligned} L(k + 1) &= \tilde{f}(L(k) \vee \bigwedge O_y(k), y(k)) \\ U(k + 1) &= \tilde{f}(U(k) \wedge \bigvee O_y(k), y(k)) \end{aligned} \tag{1.7}$$

with $L(0) = \bigwedge \chi$, $U(0) = \bigvee \chi$ is such that properties (i)–(iii) of Problem 1 are satisfied. For simplicity of notation, we omit the dependence of $\tilde{f}$ on its second argument.

Proof of (i): This is proved by induction on k. Base case: for $k = 0$ we have that $L(0) = \bigwedge \chi$ and that $U(0) = \bigvee \chi$, so that $L(0) \leq \alpha(0) \leq U(0)$. Induction step: we assume that $L(k) \leq \alpha(k) \leq U(k)$ and we show that $L(k+1) \leq \alpha(k+1) \leq U(k+1)$. Note that $\alpha(k) \in O_y(k)$. This, along with the assumption of the induction step, implies that

$$L(k) \vee \bigwedge O_y(k) \leq \alpha(k) \leq U(k) \wedge \bigvee O_y(k).$$

This last relation also implies that there is x such that $x \geq L(k) \vee \bigwedge O_y(k)$ and $x \leq \bigvee \ O_y(k)$. This in turn implies that

$$L(k) \vee \bigwedge O_y(k) \leq \bigvee O_y(k).$$

This in turn implies that $L(k) \vee \bigwedge O_y(k) \in O_y(k)$. Because of this, because (by analogous reasonings) $U(k) \wedge \bigvee O_y(k) \in O_y(k)$, and because the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible, we can use the isomorphic property of $\tilde{f}$ (property (ii) of Definition 1.5.5), which leads to

$$\tilde{f}(L(k) \vee \bigwedge O_y(k)) \leq \alpha(k+1) \leq \tilde{f}(U(k) \wedge \bigvee O_y(k)).$$

This relationship combined with equation (1.7) proves (i).

Proof of (ii): This can be shown by proving that for any $w \in [L(k+1), U(k+1)]$ there is $z \in [L(k), U(k)]$ such that $w = \tilde{f}(z)$. By equation (1.7), $w \in [L(k+1), U(k+1)]$ implies that

$$\tilde{f}(L(k) \vee \bigwedge O_y(k)) \leq w \leq \tilde{f}(U(k) \wedge \bigvee O_y(k)). \tag{1.8}$$

In addition, we have that

$$\bigwedge O_y(k) \leq L(k) \vee \bigwedge O_y(k)$$

and

$$U(k) \wedge \bigvee O_y(k) \leq \bigvee O_y(k).$$

Because the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible, by virtue of the isomorphic property of $\tilde{f}$ (property (ii) of Definition 1.5.5), we have that

$$\tilde{f}(\bigwedge O_y(k)) \leq \tilde{f}(L(k) \vee \bigwedge O_y(k))$$

and

$$\tilde{f}(U(k) \wedge \bigvee O_y(k)) \leq \tilde{f}(\bigvee O_y(k)).$$

This, along with relation (1.8), implies that

$$w \in [\tilde{f}(\bigwedge O_y(k)), \tilde{f}(\bigvee O_y(k))].$$

From this, using again the order isomorphic property of $\tilde{f}$, we deduce that there is $z \in O_y(k)$ such that $w = \tilde{f}(z)$. This with relation (1.8) implies that

$$L(k) \vee \bigwedge O_y(k) \leq z \leq U(k) \wedge \bigvee O_y(k),$$

which in turn implies that $z \in [L(k), U(k)]$.

Proof of (iii): We proceed by contradiction. Thus, assume that for any k_0 there exists a $k \geq k_0$ such that $\{\alpha(k), \beta_k\} \subseteq [L(k), U(k)] \cap \mathcal{U}$ for some $\beta_k \neq \alpha(k)$ and $\beta_k \in \mathcal{U}$. By the proof of part (ii) we also have that β_k is such that $\beta_k = \tilde{f}(\beta_{k-1})$ for some $\beta_{k-1} \in [L(k-1), U(k-1)]$.

We want to show that in fact $\beta_{k-1} \in [L(k-1), U(k-1)] \cap \mathcal{U}$. If this is not the case, we can construct an infinite sequence $\{k_i\}_{i\in\mathbb{N}^+}$ such that $\beta_{k_i} \in [L(k_i), U(k_i)] \cap \mathcal{U}$ with $\beta_{k_i} = \tilde{f}(\beta_{k_i-1})$ and $\beta_{k_i-1} \in [L(k_i-1), U(k_i-1)] \cap (\chi - \mathcal{U})$. Notice that $|[L(k_1-1), U(k_1-1)] \cap (\chi - \mathcal{U})| = M < \infty$. Also, we have

$$|[L(k_1), U(k_1)] \cap (\chi - \mathcal{U})| < |[L(k_1-1), U(k_1-1)] \cap (\chi - \mathcal{U})|.$$

This is due to the fact that $\tilde{f}(\beta_{k_1-1}) \notin [L(k_1), U(k_1)] \cap (\chi - \mathcal{U})$, and to the fact that each element in $[L(k_1), U(k_1)] \cap (\chi - \mathcal{U})$ comes from one element in $[L(k_1-1), U(k_1-1)] \cap (\chi - \mathcal{U})$ (proof of (ii) and because $\mathcal{U}$ is invariant under $\tilde{f}$). Thus we have a strictly decreasing sequence of natural numbers $\{|[L(k_i-1), U(k_i-1)] \cap (\chi - \mathcal{U})|\}$ with initial value M. Since M is finite, we reach the contradiction that $|[L(k_i-1), U(k_i-1)] \cap (\chi - \mathcal{U})| < 0$ for some i. Therefore, $\beta_{k-1} \in [L(k-1), U(k-1)] \cap \mathcal{U}$.

Thus for any k_0 there is $k \geq k_0$ such that $\{\alpha(k), \beta_k\} \subseteq [L(k), U(k)] \cap \mathcal{U}$, with $\beta_k = f(\beta_{k-1})$ for some $\beta_{k-1} \in [L(k-1), U(k-1)] \cap \mathcal{U}$. Also, from the proof of part (ii) we have that $\beta_{k-1} \in O_y(k-1)$. As a consequence, there exists $\bar{k} > 0$ such that $\{\beta_{k-1}, z(k-1)\}_{k\geq\bar{k}} = \sigma_1$ and $\{\alpha(k-1), z(k-1)\}_{k\geq\bar{k}} = \sigma_2$ are two executions of Σ sharing the same output. This contradicts the observability assumption.

Corollary 1.5.7 *If the extended system $\tilde{\Sigma}$ of an observable system Σ is observable, then the estimator $\hat{\Sigma}$ given in Theorem 1.5.6 solves Problem 1 with $L(k) = U(k) = \alpha(k)$ for $k \geq k_0$.*

1.6 Example: The RoboFlag Drill

The RoboFlag Drill has been described in Section 1.2. In this section, we revisit the example by showing that it is observable with measurable variables z, and by finding a lattice and a system extension that can be used for constructing the estimator proposed in Theorem 1.5.6. The system specification is given in equations (1.1,1.2,1.3,1.4). In particular, the rules in equations (1.2,1.3) model the function $h : \mathcal{U} \times \mathcal{Z} \to \mathcal{Z}$ that updates the continuous variables, and the rules in equations (1.4) model the function $f : \mathcal{U} \times \mathcal{Z} \to \mathcal{U}$ that updates the discrete variables. In this example, we have $\mathcal{U} = \text{perm}(N)$ the set of permutations of N elements, and $\mathcal{Z} = \mathbb{R}^N$. Thus, the RoboFlag system is given by $\Sigma = (\text{perm}(N), \mathbb{R}^N, f, h)$, in which the variables $z \in \mathbb{R}^N$ are measured.

RoboFlag Drill Observation Problem. Given initial values for x and y and the values of z corresponding to an execution of $\Sigma = (\text{perm}(N), \mathbb{R}^N, f, h)$, determine the value of α during that execution.

Before constructing the estimator for the system $\Sigma = (\text{perm}(N), \mathbb{R}^N, f, h)$, we show in the following proposition that such a system is observable.

Proposition 1.6.1 *The system $\Sigma = (perm(N), \mathbb{R}^N, f, h)$ represented by the rules (1.2-1.3-1.4) with measurable variables z is observable.*

Proof. Given any two executions σ_1 and σ_2 of Σ, for proving observability, it is enough to show that if $\{\sigma_1(k)(\alpha)\}_{k\in\mathbb{N}} \neq \{\sigma_2(k)(\alpha)\}_{k\in\mathbb{N}}$, then $\{\sigma_1(k)(z)\}_{k\in\mathbb{N}} \neq \{\sigma_2(k)(z)\}_{k\in\mathbb{N}}$. Since the measurable variables are the z_i's, their direction of motion is also measurable. Thus, we consider the vector of directions of motion of the z_i as output. Let $g(\sigma(k))$ denote such a vector at step k for the execution σ. It is enough to show that there is a k such that $g(\sigma_1(k)) \neq g(\sigma_2(k))$. Note that, in any execution of Σ, the α trajectory reaches the equilibrium value $[1, ..., N]$, and therefore there is a step $\bar{k}$ at which $f(\sigma_1(\bar{k})) = f(\sigma_2(\bar{k}))$ and $\sigma_1(\bar{k})(\alpha) \neq \sigma_2(\bar{k})(\alpha)$. As a consequence the system is observable if $g(\sigma_1(k)) \neq g(\sigma_2(k))$. Therefore it is enough to prove that for any $\alpha \neq \beta$, for $\alpha, \beta \in \mathcal{U}$, we have $g(\alpha, z) = g(\beta, v) \implies f(\alpha, z) \neq f(\beta, v)$, where $z, v \in \mathbb{R}^N$. Thus, $g(\alpha) = g(\beta)$ by (1.2-1.3) implies that (1) $z_i < x_{\alpha_i} \iff v_i < x_{\beta_i}$ and (2) $z_i \geq x_{\alpha_i} \iff v_i \geq x_{\beta_i}$. This implies that $x_{\alpha_i} \geq z_{i+1} \wedge x_{\alpha_{i+1}} \leq z_{i+1} \Leftrightarrow x_{\beta_i} \geq v_{i+1} \wedge x_{\beta_{i+1}} \leq v_{i+1}$. By denoting $\alpha' = f(\alpha, z)$ and $\beta' = f(\beta, v)$, we have that $(\alpha'_i, \alpha'_{i+1}) = (\alpha_{i+1}, \alpha_i) \Leftrightarrow (\beta'_i, \beta'_{i+1}) = (\beta_{i+1}, \beta_i)$. Hence if there exists an i such that $\alpha_i \neq \beta_i$, then there exists a j such that $\alpha'_j \neq \beta'_j$, and therefore $f(\alpha, z) \neq f(\beta, v)$.

1.6.1 RoboFlag Drill Estimator

We next construct the estimator proposed in Theorem 1.5.6 in order to estimate and track the value of the assignment α in any execution. To accomplish this, we find a lattice $(\chi, \leq)$ in which to immerse the set $\mathcal{U}$ and an extension $\tilde{\Sigma}$ of the system Σ to χ, so that the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible. We choose as χ the set of vectors in $\mathbb{N}^N$ with coordinates $x_i \in [1, N]$, that is,

$$\chi = \{x \in \mathbb{N}^N : x_i \in [1, N]\}. \tag{1.9}$$

For the elements in χ, we use the vector notation, that is, $x = (x_1, ..., x_N)$. The partial order that we choose on such a set is given by

$$\forall x, w \in \chi, \;\; x \leq w \text{ if } x_i \leq w_i \; \forall i. \tag{1.10}$$

As a consequence, the join and the meet between any two elements in χ are given by

$$\forall \; x, w \;\; \in \chi, \;\; v = x \vee w \text{ if } v_i = \max\{x_i, w_i\},$$
$$\forall \; x, w \;\; \in \chi, \;\; v = x \wedge w \text{ if } v_i = \min\{x_i, w_i\}.$$

With this choice,we have $\bigvee \chi = (N, ..., N)$ and $\bigwedge \chi = (1, ..., 1)$. The pair $(\chi, \leq)$ with the order defined by (1.10) is clearly a lattice. The set $\mathcal{U}$ is the set of all permutations of N elements and it is a subset of χ. All of the elements in $\mathcal{U}$ form an anti-chain of the lattice, that is, any two elements of $\mathcal{U}$ are not related by the order $(\chi, \leq)$. In the reminder of this section, we denote by w the variables in χ not specifying whether w is in $\mathcal{U}$, and we denote by α the variables in $\mathcal{U}$. The function $h : \text{perm}(N) \times \mathbb{R}^N \to \mathbb{R}^N$ can be naturally extended to χ as

$$\begin{aligned} z_i(k+1) &= z_i(k) + \delta \text{ if } z_i(k) < x_{w_i(k)} \\ z_i(k+1) &= z_i(k) - \delta \text{ if } z_i(k) > x_{w_i(k)} \end{aligned} \tag{1.11}$$

for $w \in \chi$. The rules (1.11) specify $\tilde{h} : \chi \times \mathbb{R}^N \to \mathbb{R}^N$, and one can check that $\tilde{h}|_{\mathcal{U} \times \mathcal{Z}} = h$. In an analogous way $f : \text{perm}(N) \times \mathbb{R}^N \to \text{perm}(N)$ is extended to χ as

$$(w_i(k+1), w_{i+1}(k+1)) = (w_{i+1}(k), w_i(k)) \text{ if } x_{w_i(k)} \geq z_{i+1}(k) \wedge x_{w_{i+1}(k)} \leq z_{i+1}(k), \tag{1.12}$$

for $w \in \chi$. The rules (1.12) model the function $\tilde{f} : \chi \times \mathbb{R}^N \to \chi$, and one can check that $\tilde{f}|_{\mathcal{U}\times\mathcal{Z}} = f$. Therefore, the system $\tilde{\Sigma} = (\chi, \mathbb{R}^N, \tilde{f}, \tilde{h})$ is the extended system of $\Sigma = (\text{perm}(N), \mathbb{R}^N, f, h)$ (see Definition 1.5.1). The following proposition shows that the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible.

Proposition 1.6.2 *The pair $(\tilde{\Sigma}, (\chi, \leq))$, where $\Sigma = (perm(N), \mathbb{R}^N, f, h)$ is represented by the rules (1.2-1.3-1.4), and $(\chi, \leq)$ is given by (1.9)-(1.10), is interval compatible.*

Proof. We need to show properties (i) and (ii) of Definition 1.5.5. To simplify notation, we neglect the dependence of $\tilde{f}$ on its second argument.

Proof of property (i): By (1.11) we have that $T_{(z^1,z^2)}(\tilde{\Sigma})$ is not empty if for any i we have $z_i^2 = z_i^1 + \delta$, $z_i^2 = z_i^1 - \delta$, or $z_i^2 = z_i^1$. Thus

$$T_{(z^1,z^2)}(\tilde{\Sigma}) = \begin{cases} \{w \mid x_{w_i} > z_i^1, \}, & \text{if } z_i^2 = z_i^1 + \delta \\ \{w \mid x_{w_i} < z_i^1, \}, & \text{if } z_i^2 = z_i^1 - \delta \\ \{w \mid x_{w_i} = z_i^1, \}, & \text{if } z_i^2 = z_i^1. \end{cases} \tag{1.13}$$

Because we assumed that $x_i < z_i < x_{i+1}$, we have that $x_{w_i} > z_i$ if and only if $w_i > i$ and $x_{w_i} < z_i$ if and only if $w_i < i$. This, along with relations (1.13) and Definition 1.5.3, imply (i).

Proof of property (ii): To show that $\tilde{f} : \mathcal{T}_i(\tilde{\Sigma}) \to [\tilde{f}(\bigwedge \mathcal{T}_i(\tilde{\Sigma})), \tilde{f}(\bigvee \mathcal{T}_i(\tilde{\Sigma}))]$ is an order isomorphism we show: a) that it is onto; and b) that it is order embedding. a) To show that it is onto, we show directly that $f(\mathcal{T}_i(\tilde{\Sigma})) = [\tilde{f}(\bigwedge \mathcal{T}_i(\tilde{\Sigma})), \tilde{f}(\bigvee \mathcal{T}_i(\tilde{\Sigma}))]$. We omit the dependence on $\tilde{\Sigma}$ to simplify notation. From the proof of (i), we deduce that the sets $\mathcal{T}_i$ are of the form $\mathcal{T}_i = (\mathcal{T}_{i,1}, ..., \mathcal{T}_{i,N})$, with $\mathcal{T}_{i,j} \in \{[1, j], [j+1, N], [j, j]\}$. Denote by $\tilde{f}(\mathcal{T}_i)_j$ the jth coordinate set of $\tilde{f}(\mathcal{T}_i)$. By equations (1.12) we derive that $\tilde{f}(\mathcal{T}_i)_j \in \{\mathcal{T}_{i,j}, \mathcal{T}_{i,j-1}, \mathcal{T}_{i,j-1}\}$. We consider the case where $\tilde{f}(\mathcal{T}_i)_j = \mathcal{T}_{i,j-1}$; the other cases can be treated in analogous way. If $\tilde{f}(\mathcal{T}_i)_j = \mathcal{T}_{i,j-1}$ then $\tilde{f}(\mathcal{T}_i)_{j-1} = \mathcal{T}_{i,j}$. Denoting $\bigwedge \mathcal{T}_i = l$ and $\bigvee \mathcal{T}_i = u$, with $l = (l_1, ..., l_N)$ and $u = (u_1, ..., u_N)$, we have also that $\tilde{f}(l)_j = l_{j-1}$, $\tilde{f}(l)_{j-1} = l_j$, $\tilde{f}(u)_j = u_{j-1}$, $\tilde{f}(u)_{j-1} = u_j$. Thus, $\tilde{f}(\mathcal{T}_i)_j = [\tilde{f}(l)_j, \tilde{f}(u)_j]$ for all j. This in turn implies that $\tilde{f}(\mathcal{T}_i) = [\tilde{f}(l), \tilde{f}(u)]$, which is what we wanted to show. b) To show that $\tilde{f} : \mathcal{T}_i \to [\tilde{f}(\bigwedge \mathcal{T}_i), \tilde{f}(\bigvee \mathcal{T}_i)]$ is order embedding, it is enough to note again that $\tilde{f}(\mathcal{T}_i)$ is obtained by switching $\mathcal{T}_{i,j}$ with $\mathcal{T}_{i,j+1}$, $\mathcal{T}_{i,j-1}$, or leaving it as $\mathcal{T}_{i,j}$. Therefore if $w \leq v$ for $w, v \in \mathcal{T}_i$ then $\tilde{f}(w) \leq \tilde{f}(v)$ since coordinate-wise we will compare the same numbers. By the same reasoning the reverse is also true, that is, if $\tilde{f}(w) \leq \tilde{f}(v)$ then $w \leq v$.

The estimator $\hat{\Sigma}$ given in Theorem 1.5.6 is thus represented by the following rules

$$l_i(k+1) = i+1 \;\text{ if }\; z_i(k+1) = z_i(k) + \delta \tag{1.14}$$

$$l_i(k+1) = 1 \;\text{ if }\; z_i(k+1) = z_i(k) - \delta \tag{1.15}$$

$$L_{i,y}(k+1) = \max\{L_i(k), l_i(k+1)\} \tag{1.16}$$

$$(L_i(k+1), L_{i+1}(k+1)) = (L_{i+1,y}(k+1), L_{i,y}(k+1))$$
$$\text{if } x_{L_{i,y}(k+1)} \geq z_{i+1}(k) \wedge x_{L_{i+1,y}(k+1)} \leq z_{i+1}(k) \tag{1.17}$$

$$u_i(k+1) = N \;\text{ if }\; z_i(k+1) = z_i(k) + \delta \tag{1.18}$$

$$u_i(k+1) = i \;\text{ if }\; z_i(k+1) = z_i(k) - \delta \tag{1.19}$$

$$U_{i,y}(k+1) = \min\{U_i(k), u_i(k+1)\} \tag{1.20}$$

$$(U_i(k+1), U_{i+1}(k+1)) = (U_{i+1,y}(k+1), U_{i,y}(k+1))$$
$$\text{if } x_{U_{i,y}(k+1)} \geq z_{i+1}(k) \wedge x_{U_{i+1,y}(k+1)} \leq z_{i+1}(k) \tag{1.21}$$

initialized with $L(0) = \bigwedge \chi$ and $U(0) = \bigvee \chi$. Rules (1.14-1.15) and (1.18-1.19) take the output information z and set the lower and upper bound of $O_y(k)$, respectively. Rules (1.16) and (1.20) compute the lower and upper bound of the intersection $[L(k), U(k)] \cap O_y(k)$, respectively. Finally, rules (1.17) and (1.21) compute the lower and upper bound of the set $\tilde{f}([L(k), U(k)] \cap O_y(k))$, respectively. Also note that the rules in (1.14-1.21) are obtained by "copying" the rules in (1.12) and correcting them by means of the output information.

1.6.2 Complexity of the RoboFlag Drill Estimator

The amount of computation required for updating L and U according to rules (1.14)–(1.21) is proportional to the amount of computation required for updating the variables α in system Σ. In fact, we have $2N$ rules, $2N$ variables, and $2N$ computations of "max" and "min" of values in $\mathbb{N}$. Therefore, the computational complexity of the algorithm that generates $L(k)$ and $U(k)$ is proportional to $2N$, which is of the same order as the complexity of the algorithm that generates the α trajectories. As established by property (iii) of Problem 1, the function of k given by $|[L(k), U(k)] \cap \mathcal{U} - \alpha(k)|$ tends to zero. This function is useful for analysis purposes, but it is not necessary to compute it at any point in the estimation algorithm proposed in equation (1.14-1.21). However, since $L(k)$ does not converge to $U(k)$ once the algorithm has converged, i.e., when $|[L(k), U(k)] \cap \mathcal{U}| = 1$, we cannot find the value of $\alpha(k)$ from the values of $U(k)$ and $L(k)$ directly. Instead of computing directly $[L(k), U(k)] \cap \mathcal{U}$, we carry out a simple algorithm, which in the case of the RoboFlag Drill example takes at most $(N^2 + N)/2$ steps and takes as inputs $L(k)$ and $U(k)$ and gives as output $\alpha(k)$ if the algorithm has converged. This is formally explained in the following paragraph.

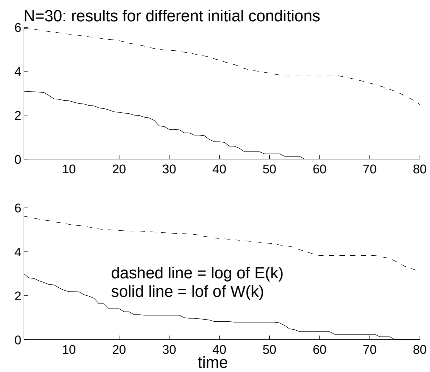

Figure 1.4 Example with N=30: note that the function $W(k)$ is always non-increasing, and its logarithm is converging to zero.

Algorithm 1 (Refinement algorithm) Let $c_i = [L_i, U_i]$. Then the algorithm $(m_1, ..., m_N) = \mathit{Refine}(c_1, ..., c_N)$, which takes assignment sets $c_1, ..., c_N$ and produces assignment sets $m_1, ..., m_N$, is such that if $m_i = \{k\}$ then $k \notin m_j$ for any $j \neq i$.

This algorithm takes as input the sets c_i and removes singletons occurring at one coordinate set from all of the other coordinate sets. By construction, it follows that $m_i \subseteq c_i$. It does this iteratively: if in the process of removing one singleton, a new one is created in some other coordinate set, then such a singleton is also removed from all of the other coordinate sets. The refinement algorithm has two useful properties. First, the sets m_i are equal to the α_i when $[L, U] \cap \mathcal{U} = \alpha$. Second, the cardinality of the sets $m_i(k)$ is non-increasing with the time step k. As a final remark, note that the sets m_i are not used in the update laws of the estimation algorithm, they are just computed at each step from the set $[L, U]$ in order to extract α from it when the algorithm has converged.

1.6.3 Simulation Results

The RoboFlag Drill system represented in the rules (1.2), (1.3), and (1.4) has been implemented in Matlab together with the estimator reported in the rules (1.14-1.21). Figure 1.4 shows the behavior of the quantities $E(k) = \frac{1}{N}\sum_{i=1}^{N} |\alpha_i(k) - i|$ and $W(k) = \frac{1}{N}\sum_{i=1}^{N} |m_i(k)|$. The function $E(k)$ is a function of α, and it is not increasing along the executions of the system $\Sigma = (\text{perm}(N), \mathbb{R}^N, f, h)$. This quantity is showing the rate of convergence of the α assignment to its equilibrium $(1, ..., N)$. We show in Figure 1.4 the logarithm of $E(k)$ and the logarithm of $W(k)$, which is non-increasing and converging to one, that is, the sets $(m_1(k), ..., m_N(k))$ converge to $\alpha(k) = (\alpha_1(k), ..., \alpha_N(k))$. In the same figure, we notice that when $W(k)$ converges to one, $E(k)$ has not converged to zero yet. This shows that the estimator is faster than the dynamics of the system under study.

1.7 Existence of Discrete State Estimators on a Lattice

In this section, we show that if the system $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$ is observable, there always exists a lattice $(\chi, \leq)$ such that the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible. The size of the set χ is singled out as a cause of complexity, and a worst case size is computed. In particular, the worst case size of the lattice never exceeds the size of the observer tree proposed in Caines et al. (1991). For systems in which $\mathcal{U}$ can be immersed in a space equipped with algebraic properties, as is the case for the RoboFlag Drill, a preferred lattice structure $(\chi, \leq)$ exists where joins and meets can be efficiently computed and represented by exploiting the algebra. For these systems, the estimation methodology proposed in this work reduces complexity with respect to enumeration methods. Consider the deterministic transition system $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$. In order to show the link between the original system and its extension, it is useful to define the Σ-transition sets and the Σ-transition classes as defined for the extended system $\tilde{\Sigma} = (\mathcal{U}, \mathcal{Z}, \tilde{f}, \tilde{h})$ in Definition 1.5.2 and Definition 1.5.3, respectively.

Definition 1.7.1 The non-empty sets $T_{(z^1,z^2)}(\Sigma) = \{\alpha \in \mathcal{U} \mid z^2 = h(\alpha, z^1)\}$, for $z^1, z^2 \in \mathcal{Z}$, are named the Σ-*transition sets*.

Definition 1.7.2 The set $\mathcal{T}(\Sigma) = \{\mathcal{T}_1(\Sigma), ..., \mathcal{T}_M(\Sigma)\}$, with $\mathcal{T}_i(\Sigma)$ such that

(i) for any $\mathcal{T}_i(\Sigma) \in \mathcal{T}(\Sigma)$ there is $(z^1, z^2) \in \mathcal{Z}$ such that $\mathcal{T}_i(\Sigma) = T_{(z^1,z^2)}(\Sigma)$;

(ii) for any $z^1, z^2 \in \mathcal{Z}$ for which $T_{(z^1,z^2)}(\Sigma)$ is not empty, there is $j \in \{1, ..., M\}$ such that $T_{(z^1,z^2)}(\Sigma) = \mathcal{T}_j$;

is the *set of* Σ-*transition classes*.

Note that the set $\mathcal{T}(\Sigma)$ is finite as we assumed at the beginning that the set $\mathcal{U}$ is finite. Each Σ-transition set $T_{(z_1,z^2)}(\Sigma)$ contains all of α values in $\mathcal{U}$ that allow the transition from z^1 to z^2 through the function h. Note also that for any $z^1, z^2 \in \mathcal{Z}$ we have $T_{(z^1,z^2)}(\Sigma) \subseteq T_{(z^1,z^2)}(\tilde{\Sigma})$ because $\tilde{h}|_{\mathcal{U}\times\mathcal{Z}} = h$ and $\mathcal{U} \subseteq \chi$. This in turn implies that $\mathcal{T}_i(\Sigma) \subseteq \mathcal{T}_i(\tilde{\Sigma})$.

Lemma 1.7.3 *Consider the deterministic transition system $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$. If Σ is observable, then $f : (\mathcal{T}_j(\Sigma), z) \to f(\mathcal{T}_j(\Sigma), z)$ is one to one for any $j \in \{1, .., M\}$ and for any $z \in \mathcal{Z}$.*

Proof. We have to show that if $\alpha_a \neq \alpha_b$ and $\alpha_a, \alpha_b \in \mathcal{T}_j(\Sigma)$ for some j, then $f(\alpha_a, z) \neq f(\alpha_b, z)$. Suppose instead that $f(\alpha_a, z) = f(\alpha_b, z)$, this means that the two executions σ_a, σ_b starting at $\sigma_a(0) = (\alpha_a, z)$ and $\sigma_b(0) = (\alpha_b, z)$ have the same output sequence, but they are different. This means that they are not distinguishable, and therefore the system is not observable. This contradicts the assumption.

This lemma is used in the following theorem, which shows the link between observability and extensibility of the system Σ into a system $\tilde{\Sigma}$ that is interval compatible with a lattice $(\chi, \leq)$.

Theorem 1.7.4 *Consider the deterministic transition system $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$. If Σ is observable, then there exists a complete lattice $(\chi, \leq)$ with $\mathcal{U} \subseteq \chi$, such that the extension $\tilde{\Sigma} = (\tilde{f}, \tilde{h}, \chi, \mathcal{Z})$ of Σ on χ is such that $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible.*

Proof. We show the existence of a lattice $(\chi, \leq)$ and of an extended system $\tilde{\Sigma} = (\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ with $(\tilde{\Sigma}, (\chi, \leq))$ an interval compatible pair by construction. Define $\chi := \mathcal{P}(\mathcal{U})$, and $(\chi, \leq) := (\mathcal{P}(\mathcal{U}), \subseteq)$. To define $\tilde{h}$, define the sublattices $(\mathcal{T}_i(\tilde{\Sigma}), \leq)$ of $(\chi, \leq)$ for $i \in \{1, ..., M\}$, by $(\mathcal{T}_i(\tilde{\Sigma}), \leq) := (\mathcal{P}(\mathcal{T}_i(\Sigma)), \subseteq)$ as shown in Figure 1.5. As a consequence, for any given $z^1, z^2 \in \mathcal{Z}$ such that $z^2 = h(\alpha, z^1)$ for $\alpha \in \mathcal{T}_i(\Sigma)$ for some i, we define $z^2 = \tilde{h}(w, z^1)$ for any $w \in \mathcal{T}_i(\tilde{\Sigma})$. Clearly, $\tilde{h}|_{\mathcal{U}\times\mathcal{Z}} = h$, and $\mathcal{T}_i(\tilde{\Sigma})$ for any i is an interval sublattice of the form $\mathcal{T}_i(\tilde{\Sigma}) = [\perp, \bigvee\mathcal{T}_i(\tilde{\Sigma})]$. The function $\tilde{f}$ is defined in the following way. For any $x, w \in \chi$ and $\alpha \in \mathcal{U}$ we define

$$\begin{cases} \tilde{f}(x \vee w) & := \tilde{f}(x) \vee \tilde{f}(w) \\ \tilde{f}(x \wedge w) & := \tilde{f}(x) \wedge \tilde{f}(w) \\ \tilde{f}(\perp) & := \perp \\ \tilde{f}(\alpha) & := f(\alpha), \end{cases} \tag{1.22}$$

in which we have omitted the dependence on z to simplify notation. We prove first that $\tilde{f} : \mathcal{T}_i(\tilde{\Sigma}) \to [\perp, \tilde{f}(\bigvee\mathcal{T}_i(\tilde{\Sigma}))]$ is onto. We have to show that for any $w \in [\perp, \tilde{f}(\bigvee\mathcal{T}_i(\tilde{\Sigma}))]$, with $w \neq \perp$, there is $x \in [\perp, \bigvee\mathcal{T}_i(\tilde{\Sigma})]$ such that $w = \tilde{f}(x)$. Since $\bigvee\mathcal{T}_i(\tilde{\Sigma}) = \alpha_1 \vee ... \vee \alpha_p$ for $\{\alpha_1, ..., \alpha_p\} = \mathcal{T}_i(\Sigma)$, we have also that $\tilde{f}(\bigvee\mathcal{T}_i(\tilde{\Sigma})) = f(\alpha_1) \vee .... \vee f(\alpha_p)$ by virtue of equations (1.22). Because $w \leq \tilde{f}(\bigvee\mathcal{T}_i(\tilde{\Sigma}))$, we have that $w = f(\alpha_{j_1}) \vee ... \vee f(\alpha_{j_m})$ for $j_k \in \{1, ..., p\}$ and $m < p$. This in turn implies, by equations (1.22), that $w = \tilde{f}(\alpha_{j_1} \vee ... \vee \alpha_{j_m})$. Since $x := \alpha_{j_1} \vee ... \vee \alpha_{j_m} < \bigvee\mathcal{T}_i(\tilde{\Sigma})$, we have proved that $w = \tilde{f}(x)$ for $x \in \mathcal{T}_i(\tilde{\Sigma})$. Second, we notice that $\tilde{f} : \mathcal{T}_i(\tilde{\Sigma}) \to [\perp, \tilde{f}(\bigvee\mathcal{T}_i(\tilde{\Sigma}))]$ is one to one because of Lemma 1.7.3. Thus, we have proved that $\tilde{f} : \mathcal{T}_i(\tilde{\Sigma}) \to [\perp, \tilde{f}(\bigvee\mathcal{T}_i(\tilde{\Sigma}))]$ is a bijection, and by equations (1.22) it is also a homomorphism. We then apply Proposition 1.3.3 to obtain the result.

Theorem 1.7.4 shows that an observable system admits a lattice and a system extension that satisfy interval compatibility by constructing them in a way similar to the way one shows that a stable dynamical system has a Lyapunov function. However, the lattice constructed in the proof of the theorem is impractical for the implementation of the estimator of Theorem 1.5.6 when the size of $\mathcal{U}$ is large because the size of the representation of the elements of χ is large as well. However, one does not need to have $\chi = \mathcal{P}(\mathcal{U})$, but it is enough to have in χ the elements that the estimator needs. With this consideration, the following proposition computes the worst case size of χ.

Proposition 1.7.5 *Consider the system* $\Sigma = (\mathcal{U}, \mathcal{Z}, f, h)$, *with* $f : \mathcal{U} \to \mathcal{U}$. *Assume that the sets* $\{\mathcal{T}_1(\Sigma), ..., \mathcal{T}_m(\Sigma)\}$ *are all disjoint. Then there exist an extension* $\tilde{\Sigma} = (\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ *with* $|\chi| \leq 2|\mathcal{U}|^2$.

For the proof, see (DelVecchio and Murray (2004)).

The size of χ gives an idea of how many values of joins and meets need to be stored. In the case of the RoboFlag example with $N = 4$ robots per team, the size of $\mathcal{P}(\mathcal{U})$ is 16778238, while the worst case size given in Proposition 1.7.5 is 576, and the size of the lattice χ proposed in Section 1.6.1 is $4^4 = 256$. Thus the estimate given by Proposition 1.7.5 significantly reduces the size of χ given by $\mathcal{P}(\mathcal{U})$. Note that the size of the lattice proposed in Section 1.6.1 is smaller than 576 because there are pairs of elements that have the same join, for example, the pairs $(3, 1, 4, 2)$, $(4, 2, 1, 3)$, and $(4, 2, 1, 3)$, $(2, 1, 4, 3)$ have the same join, that is, $(4, 2, 4, 3)$. In the next section, we extend the result of Theorem 1.5.6 to the estimation of continuous and discrete variables in case an estimator in cascade form is possible.

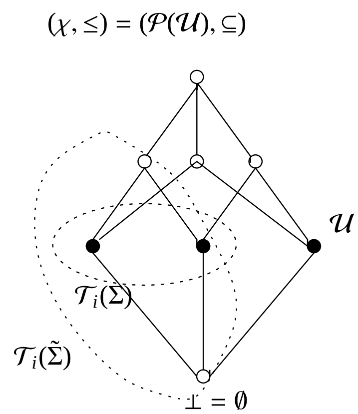

Figure 1.5 Example of $(\mathcal{P}(\mathcal{U}), \subseteq)$ with $\mathcal{U}$ composed by three elements (solid circles).

1.8 Extensions to the Estimation of Discrete and Continuous Variables

In this section, we consider systems in which the continuous variables are not directly measured as it was in the previous sections. As an example, consider the RoboFlag Drill, in which now the blue robots move according to a second order dynamics, of which only the position is measured. In order to formally define the structure of the systems under study, we define the feedback interconnection of two systems with input.

Definition 1.8.1 Consider the two systems with input $\Sigma_1 = (S_1, \mathcal{I}_1, \mathcal{Y}_1, F_1, g_1)$ and $\Sigma_2 = (S_2, \mathcal{I}_2, \mathcal{Y}_2, F_2, g_2)$, in which $\mathcal{I}_1 = \mathcal{Y}_2$, $\mathcal{I}_2 = \mathcal{Y}_1$, and $g_1 : S_1 \to \mathcal{Y}_1$. The *feedback interconnection* of Σ_1 with Σ_2, denoted by $\Sigma_1 \circ_f \Sigma_2$, is the deterministic transition system given by $\Sigma_1 \circ_f \Sigma_2 := (S_1 \times S_2, \mathcal{Y}_2, (F_1', F_2'), g_2')$, in which for any $s_1 \in S_1$ and any $s_2 \in S_2$, we have $F_1'(s_1, s_2) = F_1(s_1, g_2(s_2, g_1(s_1)))$, $F_2'(s_1, s_2) = F_2(s_2, g_1(s_1))$, and $g_2'(s_1, s_2) = g_2(s_2, g_1(s_1))$.

The output of the feedback interconnection $\Sigma_1 \circ_f \Sigma_2$ is the output of Σ_2. Let $\mathcal{U}$ be a finite discrete set, $\mathcal{Z}$ an infinite possibly dense set, and $\mathcal{Y}$ a finite or infinite set. In this section, we consider systems that are the feedback interconnection of a system that updates the discrete variable dynamics and of a system that updates the continuous variables dynamics. These systems have the form $\Sigma = \Sigma_1 \circ_f \Sigma_2$, in which $\Sigma_1 = (\mathcal{U}, \mathcal{Y}, \mathcal{U}, f, \mathrm{id}_1)$ and $\Sigma_2 = (\mathcal{Z}, \mathcal{U}, \mathcal{Y}, h, g)$. We have denoted by id_1 the function that attaches s to a pair $(s, u) \in S \times \mathcal{I}$. Thus, we will have that $\Sigma_1 \circ_f \Sigma_2 = (\mathcal{U} \times \mathcal{Z}, \mathcal{Y}, (f', h'), g')$, in which for any $\alpha, z \in \mathcal{U} \times \mathcal{Z}$ we have that $f'(\alpha, z) = f(\alpha, g(z, \alpha))$, $h'(\alpha, z) = h(z, \alpha)$, and $g'(\alpha, z) = g(z, \alpha)$. We represent these systems by means of the following update equations

$$\alpha(k+1) = f(\alpha(k), y(k)) \tag{1.23}$$
$$z(k+1) = h(z(k), \alpha(k)) \tag{1.24}$$
$$y(k) = g(z(k), \alpha(k)).$$

The Σ-*transition sets* (Definition 1.7.1) take the form

$$T_{(y_1,y_2)}(\Sigma) = \{\alpha \in \mathcal{U} \mid \exists\, z \in \mathcal{Z} \text{ such that } y_1 = g(z,\alpha) \text{ and } y_2 = g(f(\alpha, y_1), h(z,\alpha))\}$$

with $y_1, y_2 \in \mathcal{Y}$. We denote the property that allows to distinguish two different initial values of the discrete state α of the Σ independently of the continuous state z by independent discrete state observability.

Definition 1.8.2 The system $\Sigma = \Sigma_1 \circ_f \Sigma_2$ is said to be *independent discrete state observable* if it is observable and for any output sequence $\{y(k)\}_{k\in\mathbb{N}}$, we have that for any two executions $\sigma_1, \sigma_2 \in \mathcal{E}(\Sigma)$ such that $\{\sigma_1(k)(\alpha)\}_{k\in\mathbb{N}} \neq \{\sigma_2(k)(\alpha)\}_{k\in\mathbb{N}}$, there is $k > 0$ such that $\sigma_1(k)(\alpha) \in T_{(y(k),y(k+1))}(\Sigma)$ and $\sigma_2(k)(\alpha) \notin T_{(y(k),y(k+1))}(\Sigma)$.

This property will allow to construct a discrete-continuous state estimator that is a cascade interconnection of a discrete state estimator as we have developed in Section 1.5, and a continuous state estimator. Before introducing such an estimator, we define the cascade interconnection of two systems with input.

Definition 1.8.3 Consider the two systems with input $\Sigma_1 = (S_1, \mathcal{I}_1, \mathcal{Y}_1, F_1, g_1)$ and $\Sigma_2 = (S_2, \mathcal{I}_2, \mathcal{Y}_2, F_2, g_2)$, in which $\mathcal{I}_2 = \mathcal{Y}_1$. The *cascade interconnection*, denoted $\Sigma_1 \circ_c \Sigma_2$, is the deterministic transition system with input given by $\Sigma_1 \circ_c \Sigma_2 := (S_1 \times S_2, \mathcal{I}_1, \mathcal{Y}_2, (F_1', F_2'), g_2')$, such that for any $s_1 \in S_1$, $s_2 \in S_2$ and $u_1 \in \mathcal{I}_1$ we have that $F_1'(s_1, s_2, u_1) = F_1(s_1, u_1)$, $F_2'(s_1, s_2, u_1) = F_2(s_2, g_1(s_1, u_1))$ and $g_2'(s_1, s_2, u_1) = g_2(s_2, g_1(s_1, u_1))$.

Consider the deterministic transition system $\Sigma = \Sigma_1 \circ_f \Sigma_2$, with output sequence $\{y(k)\}_{k\in\mathbb{N}}$. From the measurement of the output sequence, we construct a cascade state estimator: A system $\hat{\Sigma} = \hat{\Sigma}_1 \circ_c \hat{\Sigma}_2$, in which $\hat{\Sigma}_1$ takes as input the values of the output of Σ and asymptotically tracks the value of the variables α, while $\hat{\Sigma}_2$ takes as input the discrete state estimates and the output of Σ to asymptotically track the value of z. The cascade state estimation problem is formally introduced as follows.

Problem 2 (Cascade state estimator) Given the deterministic transition system $\Sigma = \Sigma_1 \circ_f \Sigma_2$, find the cascade interconnection $\hat{\Sigma} = \hat{\Sigma}_1 \circ_c \hat{\Sigma}_2$, in which $\hat{\Sigma}_1$ is as given in Theorem 1.5.6 and $\hat{\Sigma}_2 = (\mathcal{L} \times \mathcal{L}, \chi \times \chi \times \mathcal{Y} \times \mathcal{Y}, \chi \times \chi \times \mathcal{Z}_E \times \mathcal{Z}_E, (f_3, f_4), (g_1, g_2, g_3, g_4))$ where $f_3 : \mathcal{L} \times \chi \times \chi \times \mathcal{Y} \times \mathcal{Y} \to \mathcal{L}$, $f_4 : \mathcal{L} \times \chi \times \chi \times \mathcal{Y} \times \mathcal{Y} \to \mathcal{L}$, $g_1 : \chi \to \chi$, $g_2 : \chi \to \chi$, $g_1 = g_2 = \mathrm{id}$, $g_3 : \mathcal{L} \to \mathcal{Z}_E$, and $g_4 : \mathcal{L} \to \mathcal{Z}_E$, $\mathcal{U} \subseteq \chi$, $(\chi, \leq)$ a lattice, $\mathcal{Z} \subseteq \mathcal{Z}_E$ with $(\mathcal{Z}_E, \leq)$ a lattice, $\chi \times \mathcal{Z}_E \subseteq \mathcal{L}$, $(\mathcal{L}, \leq)$ a lattice, such that for any execution $\sigma = \{(\alpha(k), z(k))\}_{k\in\mathbb{N}}$ of Σ with output sequence $\{y(k)\}_{k\in\mathbb{N}}$ the update laws

$$\begin{aligned} L(k+1) &= f_1(L(k), y(k), y(k+1)) \\ U(k+1) &= f_2(U(k), y(k), y(k+1)) \\ q_L(k+1) &= f_3(q_L(k), L(k), U(k), y(k), y(k+1)) \qquad (1.25) \\ q_U(k+1) &= f_4(q_U(k), L(k), U(k), y(k), y(k+1)), \end{aligned}$$

in which $L(0) := \bigwedge \chi$, $U(0) := \bigvee \chi$, $q_L(0) = \bigwedge \mathcal{L}$, $q_U(0) = \bigvee \mathcal{L}$, and $z_L(k) = g_3(q_L(k))$, and $z_U(k) = g_4(q_U(k))$, have properties (i)−−(iii) of Problem 1 and

(i') $z_L(k) \leq z(k) \leq z_U(k)$ (correctness);

(ii') $d(z_L(k), z_U(k)) \leq \gamma(|[L(k), U(k)]|)$, with γ a monotonically decreasing function of its argument (non-increasing error);

(iii') there exists $k_0' > 0$ such that $d(z_{L'}(k), z_{U'}(k)) = 0$ for any $k \geq k_0'$, where $L'(k) = \bigwedge([L(k), U(k)] \cap \mathcal{U})$, $U'(k) = \bigvee([L(k), U(k)] \cap \mathcal{U})$, and

$$q_{L'}(k+1) = f_3(q_{L'}(k), L'(k), U'(k), y(k), y(k+1))$$
$$q_{U'}(k+1) = f_4(q_{U'}(k), L'(k), U'(k), y(k), y(k+1))$$
$$z_{L'}(k) = \bigwedge g_3\left([q_{L'}(k), q_{U'}(k)] \cap (\mathcal{U} \times \mathcal{Z})\right) \tag{1.26}$$
$$z_{U'}(k) = \bigvee g_4\left([q_{L'}(k), q_{U'}(k)] \cap (\mathcal{U} \times \mathcal{Z})\right) \tag{1.27}$$

with $q_{L'}(0) = q_L(0)$ and $q_{U'}(0) = q_U(0)$, for some distance function "d" (convergence).

□

The variables L and U have the same meaning as they had in Section 1.5. The variables z_L and z_U instead represent the lower and upper bounds in $(\mathcal{Z}_E, \leq)$ of the set of all possible continuous variable values that are compatible with the output sequence, with the system dynamics established by Σ, and with the set of possible discrete variable values. The variables q_L and q_U are auxiliary variables that are needed to model the coupling of the continuous and discrete state dynamics. They represent the lower and upper bounds in $(\mathcal{L}, \leq)$ of the set of all possible pairs (α, z) compatible with the output sequence, with the system dynamics, and with the set of possible discrete variable values. The distance function "d" has been left unspecified for the moment, as its form depends on the particular partial order chosen $(\mathcal{Z}_E, \leq)$. In the case in which $\mathcal{Z}_E = \mathcal{Z}$ and $\mathcal{Z} = \mathbb{R}^n$ with the order established component-wise, the distance can be the classical euclidean distance, for example. To determine a solution to Problem 2, we introduce the notion of extension of a system to a joint continuous-discrete partial order.

Definition 1.8.4 Consider the system $\Sigma = \Sigma_1 \circ_f \Sigma_2$ with $\Sigma_1 = (\mathcal{U}, \mathcal{Y}, \mathcal{U}, f, \mathrm{id}_1)$ and $\Sigma_2 = (\mathcal{Z}, \mathcal{U}, \mathcal{Y}, h, g)$. Let $(\chi, \leq)$, $(\mathcal{Z}_E, \leq)$, and $(\mathcal{L}, \leq)$ be partial orders with $\mathcal{U} \subseteq \chi$, $\mathcal{Z} \subseteq \mathcal{Z}_E$, and $\chi \times \mathcal{Z}_E \subseteq \mathcal{L}$. The system extension is defined as $\tilde{\Sigma} = (\mathcal{L}, \mathcal{Y}, \tilde{F}, \tilde{G})$, in which

(i) $\tilde{F} : \mathcal{L} \to \mathcal{L}$ with $\tilde{F}|_{\mathcal{U}\times\mathcal{Z}} = (f', h')$ and $\mathcal{L} - (\mathcal{U} \times \mathcal{Z})$ is invariant under $\tilde{F}$;

(ii) $\tilde{G} : \mathcal{L} \to \mathcal{Y}$ with $\tilde{G}|_{\mathcal{U}\times\mathcal{Z}} = g'$;

(iii) $\tilde{\Sigma}|_{\chi\times\mathcal{Z}_E} = \tilde{\Sigma}_1 \circ_f \tilde{\Sigma}_2$, in which $\tilde{\Sigma}_1 = (\chi, \mathcal{Y}, \chi, \tilde{f}, \mathrm{id}_1)$ and $\tilde{\Sigma}_2 = (\mathcal{Z}_E, \chi, \mathcal{Y}, \tilde{h}, \tilde{g})$, with $\tilde{f}|_{\mathcal{U}\times\mathcal{Y}} = f$, $\tilde{h}|_{\mathcal{Z}\times\mathcal{U}} = h$, and $\tilde{g}|_{\mathcal{Z}\times\mathcal{U}} = g$;

(iv) the partial order $(\mathcal{L}, \leq)$ is closed with respect to $\chi \times \mathcal{Z}_E$.

The $\tilde{\Sigma}$-transition sets thus take the following form. For any $y_1, y_2 \in \mathcal{Y}$, $T_{(y_1,y_2)}(\tilde{\Sigma}) = \{w \in \chi \mid \exists\, z \in \mathcal{Z}$ such that $y_2 = \tilde{g}(\tilde{F}(w, z))$ and $y_1 = \tilde{g}(w, z)\}$. The $\tilde{\Sigma}$-transition sets correspond to the set of all possible values of $w \in \chi$ compatible with two consecutive outputs of the extended system $\tilde{\Sigma}$. The *output set* is again given by $O_y(k) := T_{(y(k),y(k+1))}(\tilde{\Sigma})$. The next definition introduces the notion of interval compatibility of the tuple $(\tilde{\Sigma}_1, \tilde{\Sigma}, (\chi, \leq))$.

Definition 1.8.5 The tuple $(\tilde{\Sigma}_1, (\chi, \leq))$ is said to be *interval compatible* if the following are verified

(i) for any $y_1, y_2 \in \mathcal{Y}$, we have that $T_{y_1,y_2}(\tilde{\Sigma}) = [\wedge T_{y_1,y_2}(\tilde{\Sigma}), \vee T_{y_1,y_2}(\tilde{\Sigma})]$;

(ii) the extension $\tilde{\Sigma}_1$ is such that $\tilde{f} : (T_{y_1,y_2}(\tilde{\Sigma}), y_1) \to [\tilde{f}(\bigwedge T_{y_1,y_2}(\tilde{\Sigma}), y_1), \tilde{f}(\bigvee T_{y_1,y_2}(\tilde{\Sigma}), y_1)]$ is an order isomorphism.

In order to determine the set of variable values in $\mathcal{L}$ of the extended system that are compatible with an output pair y_1, y_2 and with a set of possible discrete variable values $[w_1, w_2] \subseteq T_{y_1,y_2}(\tilde{\Sigma})$, we introduce the notion of induced output set.

Definition 1.8.6 Consider the system $\tilde{\Sigma} = (\mathcal{L}, \mathcal{Y}, \tilde{F}, \tilde{g})$ and a transition set $T_{(y_1,y_2)}(\tilde{\Sigma})$ for some $y_1, y_2 \in \mathcal{Y}$. For any $w_1, w_2 \in T_{(y_1,y_2)}(\tilde{\Sigma})$ with $w_1 \leq w_2$, the sets

$$I_{(y_1,y_2)}^{[w_1,w_2]} = \{q \in \mathcal{L} \mid \pi_1 \circ a_L(q) \geq w_1,\ \pi_1 \circ a_U(q) \leq w_2,\ y_2 = \tilde{g}(\tilde{F}(q)),\ \text{and } y_1 = \tilde{g}(q)\}$$

are named the induced output sets of $\tilde{\Sigma}$ induced by an interval $[w_1, w_2] \subseteq T_{y_1,y_2}(\tilde{\Sigma})$.

The meaning of an induced output set is the following. The set $I_{(y_1,y_2)}^{[w_1,w_2]}$ is the set of all possible values of $q \in \mathcal{L}$ that are compatible with two output measurements y_1, y_2 *and* whose upper and lower approximations in $\chi \times \mathcal{Z}_E$ have the discrete component contained in the set $[w_1, w_2]$. One can easily verify that if $[w_1, w_2] \subseteq T_{y_1,y_2}(\tilde{\Sigma})$, then $\{(\alpha, z) \mid g(\alpha, z) = y_1$ and $g(f(\alpha, y_1), h(\alpha, z)) = y_2\}$ with $\alpha \in [w_1, w_2]$ is contained in $I_{(y_1,y_2)}^{[w_1,w_2]}$. Next, a definition similar to interval compatibility is introduced for the induced output sets.

Definition 1.8.7 The pair $(\tilde{\Sigma}, (\mathcal{L}, \leq))$ is said to be *induced interval compatible* if $(\tilde{\Sigma}_1, (\chi, \leq))$ is interval compatible and for any $[w_1, w_2] \subseteq T_{y_1,y_2}(\tilde{\Sigma})$ for $y_1, y_2 \in \mathcal{Y}$, we have that

(i) $\tilde{F} : ([\wedge I_{(y_1,y_2)}^{[w_1,w_2]}, \vee I_{(y_1,y_2)}^{[w_1,w_2]}]) \to [\tilde{F}(\wedge I_{(y_1,y_2)}^{[w_1,w_2]}), \tilde{F}(\vee I_{(y_1,y_2)}^{[w_1,w_2]})]$ is order preserving;

(ii) $\tilde{F} : ([\wedge I_{(y_1,y_2)}^{[\alpha,\alpha]}, \vee I_{(y_1,y_2)}^{[\alpha,\alpha]}]) \to [\tilde{F}(\wedge I_{(y_1,y_2)}^{[\alpha,\alpha]}), \tilde{F}(\vee I_{(y_1,y_2)}^{[\alpha,\alpha]})]$ is an order isomorphism;

(iii) for any $[w_1, w_2] \subseteq T_{y_1,y_2}(\tilde{\Sigma})$, we have that $d\left(\pi_2 \circ a_L \circ \tilde{F}(\wedge I_{(y_1,y_2)}^{[w_1,w_2]}), \pi_2 \circ a_U \circ \tilde{F}(\vee I_{(y_1,y_2)}^{[w_1,w_2]})\right) \leq \gamma(|[w_1, w_2]|)$, for some distance function "d," and $\gamma : \mathbb{N} \to \mathbb{R}$ a monotonic function of its argument.

Item (i) and (ii) of this definition require that the extended function $\tilde{F}$ has order preserving properties on the induced output sets. Item (iii) establishes that the distance between the lower and upper bounds of the interval sublattice in $(\mathcal{Z}_E, \leq)$ induced by an interval $[w_1, w_2] \in \chi$ is bounded by a monotonic function of the size of $[w_1, w_2]$. When $(y_1, y_2) = (y(k), y(k+1))$ in the above definitions, in which $\{y(k)\}_{k\in\mathbb{N}}$ is an output sequence of Σ, we will use the notation $(y(k), y(k+1)) := Y(k)$ so that $I_{y(k),y(k+1)}^{[w_1,w_2]} = I_{Y(k)}^{[w_1,w_2]}$. A solution to Problem 2 is proposed in the following theorem.

Theorem 1.8.8 *Let* $\{y(k)\}_{k\in\mathbb{N}}$ *be the output sequence corresponding to an execution of* Σ. *Let* $\hat{\Sigma}_1$ *be as in Theorem 1.5.6. Consider the additional system with input* $\hat{\Sigma}_2 = (\mathcal{L} \times \mathcal{L}, \chi \times \chi \times$

$\mathcal{Y}\times\mathcal{Y}, \mathcal{Z}_E\times\mathcal{Z}_E, (f_3, f_4), (g_1, g_2, g_3, g_4))$ with $f_3 : \mathcal{L}\times\chi\times\chi\times\mathcal{Y}\times\mathcal{Y}\to\mathcal{L}$, $f_4 : \mathcal{L}\times\chi\times\chi\times\mathcal{Y}\times\mathcal{Y}\to\mathcal{L}$, $g_3 : \mathcal{L}\to\mathcal{Z}_E$, and $g_4 : \mathcal{L}\to\mathcal{Z}_E$ given by

$$f_3\left(q_L(k), L(k), U(k), y(k), y(k+1)\right) = \tilde{F}\left(q_L(k)\vee\bigwedge I_{Y(k)}^{[L^*(k),U^*(k)]}\right) \tag{1.28}$$

$$f_4\left(q_U(k), L(k), U(k), y(k), y(k+1)\right) = \tilde{F}\left(q_U(k)\wedge\bigvee I_{Y(k)}^{[L^*(k),U^*(k)]}\right) \tag{1.29}$$

$$g_3(q_L(k)) = \pi_2\circ a_L(q_L(k)) \tag{1.30}$$

$$g_4(q_U(k)) = \pi_2\circ a_U(q_U(k)) \tag{1.31}$$

in which $L(k), U(k), y(k), y(k+1)$ is the output of $\hat{\Sigma}_1$, $L^*(k) = \bigwedge O_y(k)\vee L(k)$, $U^*(k) = \bigvee O_y(k)\wedge U(k)$. If Σ is independent discrete state observable and $(\tilde{\Sigma}, (\mathcal{L}, \leq))$ is induced interval compatible, $\hat{\Sigma}_1\circ_c\hat{\Sigma}_2$ solves Problem 2.

Proof. Properties (i)–(iii) follow directly from Theorem 1.5.6. The proof of (i)'–(iii)' proceeds as follows. The proof of (i)' exploits the order preserving properties of $\tilde{F}$, a_L, a_U, and π_2. The proof of (ii)' exploits the property of induced order compatibility and the definition of distance on a partial order. The proof of (iii)' uses directly the observability of system Σ.

Proof of (i)'. We use induction argument on k. Initially, $q_L(0) = \bigwedge\mathcal{L}$ and $q_U(0) = \bigvee\mathcal{L}$. Therefore, we have that $q_L(0)\leq(\alpha(0), z(0))\leq q_U(0)$. Next, we show that if $q_L(k)\leq(\alpha(k), z(k))\leq q_U(k)$ then also $q_L(k+1)\leq(\alpha(k+1), z(k+1))\leq q_U(k+1)$. Since $\alpha(k)\in[L^*(k), U^*(k)]\subseteq T_{y(k),y(k+1)}(\tilde{\Sigma})$ and also $(\alpha(k), z(k))\in\{(\alpha, z)\mid g(\alpha, z) = y(k)$ and $g(f(\alpha, y(k)), h(\alpha, z)) = y(k+1)\}$, it follows that $(\alpha(k), z(k))\in I_{Y(k)}^{[L^*(k),U^*(k)]}$. As a consequence, we have that

$$q_L(k)\vee\bigwedge I_{Y(k)}^{[L^*(k),U^*(k)]}\leq(\alpha(k), z(k))\leq q_U(k)\wedge\bigvee I_{Y(k)}^{[L^*(k),U^*(k)]}.$$

By property (i) of Definition 1.8.7 (order preserving property) it follows that $\tilde{F}(q_L(k)\vee\bigwedge I_Y^{[L^*,U^*]})\leq(\alpha(k+1), z(k+1))\leq\tilde{F}(q_U(k)\wedge\bigvee I_Y^{[L^*,U^*]})$, which in turn implies by equations (1.28) and (1.29) that $q_L(k+1)\leq(\alpha(k+1), z(k+1))\leq q_U(k+1)$. We are thus left to show that $q_L(k)\leq(\alpha(k), z(k))$ implies that $\pi_2\circ a_L(q_L(k))\leq z(k)$ and that $(\alpha(k), z(k))\leq q_U(k)$ implies that $z(k)\leq\pi_2\circ a_U(q_U(k))$. These are true as $\pi_2\circ a_L$ and $\pi_2\circ a_U$ are order preserving maps, $\pi_2\circ a_L(\alpha(k), z(k)) = z(k)$, and $\pi_2\circ a_U(\alpha(k), z(k)) = z(k)$.

Proof of (ii)'. Since $\tilde{F}$ is order preserving on the induced transition sets, we have (neglecting the dependence on k) that

$$\tilde{F}(\bigwedge I_Y^{[L^*,U^*]})\leq\tilde{F}(q_L\vee\bigwedge I_Y^{[L*,U^*]})\text{ and }\tilde{F}(\bigvee I_Y^{[L^*,U^*]})\geq\tilde{F}(q_L\wedge\bigvee I_Y^{[L*,U^*]}).$$

Since $\pi_2\circ a_L$ and $\pi_2\circ a_U$ are also order preserving, by using property (iii) of the distance function (Definition 1.3.4), we have that

$$d\left(\pi_2\circ a_L\circ\tilde{F}(q_L\vee\bigwedge I_Y^{[L^*,U^*]}), \pi_2\circ a_U\circ\tilde{F}(q_U\wedge\bigvee I_Y^{[L^*,U^*]})\right)\leq$$
$$d\left(\pi_2\circ a_L\circ\tilde{F}(\bigwedge I_Y^{[L^*,U^*]}), \pi_2\circ a_U\circ\tilde{F}(\bigvee I_Y^{[L^*,U^*]})\right).$$

By property (iii) of Definition 1.8.7, we have that

$$d\left(\pi_2\circ a_L\circ\tilde{F}(\bigwedge I_Y^{[L^*,U^*]}), \pi_2\circ a_U\circ\tilde{F}(\bigvee I_Y^{[L^*,U^*]})\right)\leq\gamma([L^*, U^*]). \tag{1.32}$$

Since $\tilde{f}([L^*, U^*], y) = [\tilde{f}(L^*, y), \tilde{f}(U^*, y)]$, $\tilde{f}(L^*(k), y(k)) = L(k+1)$, and $\tilde{f}(U^*(k), y(k)) = U(k+1)$, we have by the order isomorphism property of $\tilde{f}$ that $|\tilde{f}([L^*, U^*], y(k))| = |[L^*, U^*]| = |[L(k+1), U(k+1)]|$. Thus $\gamma(|[L^*, U^*]|) = \gamma([[L(k+1), U(k+1)]])$. This along with equation (1.32) completes the proof.

Proof of (iii)'. For $k > k_0$, $L'(k) = \alpha(k) = U'(k)$ because $[L(k), U(k)] \cap \mathcal{U} = \alpha(k)$. As a consequence, $q_{L'}(k+1) = \tilde{F}(q_{L'}(k) \vee \wedge I_{Y(k)}^{[\alpha(k),\alpha(k)]})$ and $q_{U'}(k+1) = \tilde{F}(q_{U'}(k) \wedge \vee I_{Y(k)}^{[\alpha(k),\alpha(k)]})$. By property (ii) of Definition 1.8.7, it follows that for any $k > k_0$ we have that for any $q' \in [q_{L'}(k+1), q_{U'}(k+1)]$ there is $q \in [q_{L'}(k), q_{U'}(k)]$ such that $q' = \tilde{F}(q)$. Also, $\mathcal{L} - (\mathcal{U} \times \mathcal{Z})$ is invariant under $\tilde{F}$ and $\tilde{F}|_{\mathcal{U}\times\mathcal{Z}} = (f', h')$, Therefore, it is also true that for any $(\alpha', z') \in [q'_L(k+1), q'_U(k+1)] \cap (\mathcal{U} \times \mathcal{Z})$ there is $(\alpha, z) \in [q_{L'}(k), q_{U'}(k)] \cap (\mathcal{U} \times \mathcal{Z})$ such that $(\alpha', z') = (f(\alpha, y(k)), h(\alpha, z))$. In addition, we have that such (α, z) is in the induced transition set, that is, $(\alpha, z) \in I_{Y(k)}^{[\alpha(k),\alpha(k)]}$. Thus in turn implies that $g(\alpha, z) = y(k)$. Since this is true for any $k \geq k_0$, if for any k we have that $[q'_L(k), q'_U(k)] \cap (\mathcal{U} \times \mathcal{Z})$ contains more than one element, it means that there are at least two executions of Σ, $\sigma_1 \neq \sigma_2$, such that $g(\sigma_1) = g(\sigma_2)$. This contradicts observability of Σ. Thus, it must be that there is $k'_0 > k_0$ such that for $k \geq k'_0$ we have that $[q'_L(k), q'_U(k)] \cap (\mathcal{U} \times \mathcal{Z}) = (\alpha(k), z(k))$. As a consequence, by virtue of equations (1.26,1.27,1.30,1.31) we also have that $z_{L'}(k) = z_{U'}(k)$ for all $k \geq k'_0$.

1.8.1 RoboFlag Drill with continuous dynamics

A version of the RoboFlag Drill system of Section 1.2 is considered in which now the defender robots have partially measured second order dynamics instead of having fully measured first order dynamics. To describe this dynamics, each defender robot motion is denoted by variable $z_i \in \mathbb{R}^2$, in which $z_{i,1}$ is the position. We consider the problem of estimating the current assignment α and continuous variables z_i given measured positions $z_{i,1}$ of the defender robots. The function $h : \mathcal{U} \times \mathcal{Z} \to \mathcal{Z}$ that updates the z variables is given by

$$
\begin{aligned}
z'_{i,1} &= (1-\beta)z_{i,1} - \beta z_{i,2} + 2\beta x_{\alpha_i} \\
z'_{i,2} &= (1-\lambda)z_{i,2} + \lambda x_{\alpha_i}
\end{aligned}
\tag{1.33}
$$

for any i. The set $\mathcal{Z}$ is such that $z_{i,1} \in [x_i, x_{i+1}]$ and $z_{i,2} \in [x_i, x_{i+1}]$ for any i, which is guaranteed if β and λ are assumed sufficiently small. This means that each defender moves toward the x position of the assigned attacker with second order damped dynamics. The continuous variables are $z = (z_{1,1}, z_{1,2}, ..., z_{N,1}, z_{N,2}) \in \mathcal{Z}$, with output $g(z) = (z_{1,1}, ..., z_{N,1}) \in \mathcal{Y}$. Thus, $\Sigma_1 = (\mathcal{U}, \mathcal{Y}, \mathcal{U}, f, \mathrm{id}_1)$ and $\Sigma_2 = (\mathcal{Z}, \mathcal{U}, h, g)$, in which f and $\mathcal{U}$ are the same as in Section 1.2 and h is now given by equations (1.33). The overall system is given by the feedback interconnection of Σ_1 with Σ_2, that is, $\Sigma = \Sigma_1 \circ_f \Sigma_2$. The partial order $(\chi, \leq)$ and extension $\tilde{f}$ are the same as the ones defined in equations (1.9) and (1.10). We define $\mathcal{L} = \chi \times \mathcal{Z}$ and $\tilde{F} = (\tilde{f}, \tilde{h})$, in which $\tilde{h} : \chi \times \mathcal{Z} \to \mathcal{Z}$ is the same as h in equations (1.33) but with $x \in \chi$ in place of $\alpha \in \mathcal{U}$. The partial order in $\mathcal{Z}$ is established by $z^a \leq z^b$ for $z^a, z^b \in \mathcal{Z}$ if $z^a_{i,2} \leq z^b_{i,2}$. One can verify that the order on each $z_{i,2}$ is preserved by the dynamics in equations (1.33). In fact, if $z^{(1)}_{i,2} < z^{(2)}_{i,2}$ and $w^{(1)}_i \leq w^{(2)}_i$ then $(1-\lambda)z^{(1)}_{i,2} + \lambda x_{w^{(1)}_i} \leq (1-\lambda)z^{(2)}_{i,2} + \lambda x_{w^{(2)}_i}$ because $x_{w^{(1)}_i} \leq x_{w^{(2)}_i}$ whenever $w^{(1)}_i \leq w^{(2)}_i$, and because $(1-\lambda) > 0$. With such partial order on $\mathcal{Z}$, one can show that the assumptions of Theorem 1.8.8 are verified. The estimator in Theorem 1.8.8 has been implemented. For the continuous state estimator, let $z_L, z_U \in \mathbb{R}^N$ represent the lower and upper bound of $(z_{1,2}, ..., z_{N,2})$. Figure 1.6 illustrates the estimator performance, in which

$W(k) = \sum_{i=1}^{N} |m_i(k)|$, where $|m_i(k)|$ is the cardinality of the sets $m_i(k)$ that are the sets of possible α_i for each component obtained from the sets $[L_i, U_i]$ by removing iteratively a singleton occurring at component i by all other components. When $[L(k), U(k)] \cap \text{perm}(N)$ has converged to α, then $m_i(k) = \alpha_i(k)$. The distance function for $z, x \in \mathbb{R}^N$ is defined by $d(x, z) = \sum_{i=1}^{N} \text{abs}(z_i - x_i)$. The function $V(k)$ in the plot is defined by $V(k) = \frac{1}{2} \sum_{i=1}^{N} (x_{U_i(k)} - x_{L_i(k)})$, and it is always non increasing. A possible candidate for γ is thus $\gamma(|[L, U]|) := |[L, U]| \sup_i |x_{U_i} - x_{L_i}|$. Note that even if the discrete state has not converged yet, the continuous state estimation error after $k = 8$ is close to zero.

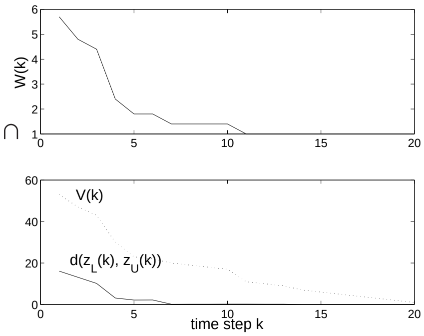

Figure 1.6 Estimator performance with $N = 10$ agents.

1.9 Conclusions

In this work, we have presented an approach to state estimation in decision and control systems, which reduces complexity by using partial order theory. The developed algorithms enjoy scalability properties that are substantial in multi-agent systems. This has been done for estimating the discrete state once the continuous state is measured and for estimating both discrete and continuous state when an estimator in cascade form is possible. Partial order theory has proved to be a useful tool borrowed from theoretical computer science to address this issue, and it was nicely merged with classical control theory to reach our goal. The question of how to deal with the estimation problem for both the continuous and the discrete state when an estimator in cascade form is not possible will be addressed in our future work. In particular, we will take advantage of the “cooperative” and “competitive” nature of the multi-agent systems under study that naturally imposes order preserving dynamics in suitable partial orders. The dynamic feedback control problem will also be addressed in our future work by using partial order theory. For the case of the RoboFlag Drill, we will design a dynamic controller for the attackers, which on the basis of the state estimates, decides the next action to take in order to win the game. These are mainly synthesis issues, however we will also explore how partial order theory can be employed to solve analysis problems such as the computation of escape sets and controlled invariant sets with low computational load.

Bibliography

Alessandri A and Coletta P 2001 Design of Luenberger observer for a class of hybrid linear systems *Lecture Notes in Computer Science* 2034, M. D. Di Benedetto, A. Sangiovanni-Vincentelli Eds. Springer Verlag, pp. 7–18.

Alessandri A and Coletta P 2003 Design of observer for switched discrete-time linear systems *American Control Conference*, pp. 2785–2790.

Balluchi A, Benvenuti L, Benedetto MDD and Sangiovanni-Vincentelli A 2002 Design of observers for hybrid systems *Lecture Notes in Computer Science* 2289,C. J. Tomlin and M. R. Greensreet Eds. Springer Verlag, pp. 76–89.

Balluchi A, Benvenuti L, Benedetto MDD and Sangiovanni-Vincentelli A 2003 Observability of hybrid systems *Conf. on Decision and Control*, pp. 1159–1164.

Bemporad A, Ferrari-Trecate G and Morari M 1999 Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control* **45**, 1864–1876.

Bui HH, Venkatesh S and West G 2002 Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research* **17**, 451–499.

Caines PE and Wang S 1995 Cocolog: A conditional observer and controller logic for finite machines. *SIAM J. Control and Optimization* pp. 1687–1715.

Caines PE and Wei Y 1996 The hierarchical lattices of a finite machine. *Systems and Control Letters* **25**, 257–263.

Caines PE, Greiner R and Wang S 1991 Classical and logic-based dynamic observers for finite automata. *IMA J. of Mathematical Control and Information* pp. 45–80.

Cassandra AR, Kaelbling LP and Littman ML 1994 Acting optimally in partially observable stochastic domains *Proc. 12th Conference on Artificial Intelligence*, pp. 1023–1028, Seattle, WA.

Collins RT, Lipton AJ, Fujiyoshi H and Kanade T 2001 Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE* **89**(10), 1456–1477.

Costa EF and do Val JBR 2002 On the observability and detectability of continuous-time jump linear systems. *SIAM Journal on Control and Optimization* **41**, 1295–1314.

Cousot P and Cousot R 1977 Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints *Conference Record of the Sixth Annual ACM Symposium on Principles of Programming Languages*, pp. 238–252, Los Angeles.

D'Andrea R, Murray RM, Adams JA, Hayes AT, Campbell M and Chaudry A 2003 The RoboFlag Game *American Control Conference*, pp. 661–666.

Davey BA and Priesteley HA 2002 *Introduction to Lattices and Order*. Cambridge University Press.

DelVecchio D and Murray RM 2004 Existence of discrete state estimators for hybrid systems on a lattice *Conf. on Decision and Control*.

Diaz M, Juanole G and Courtiat J 1994 Observer-a concept for formal on-line validation of distributed systems. *IEEE Trans. on Software Engineering* **20**, 900–913.

Godefroid P 1996 *Partial-order methods for the verification of concurrent systems* Lecture notes in computer science. Springer.

Kalman RE 1960 A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering* **82**(Series D), 35–45.

Klavins E 2003 A formal model of a multi-robot control and communication task *Conf. on Decision and Control*, pp. 4133–4139, Hawaii.

Klavins E and Murray RM 2004 Distributed algorithms for cooperative control. *Pervasive Computing* **3**, 56–65.

Luenberger DG 1971 An introduction to observers. *IEEE Transactions on Automatic Control* **AC-16:6**, 596–602.

Oishi M, Hwang I and Tomlin C 2003 Immediate observability of discrete event systems with application to user-interface design *Conf. on Decision and Control*, pp. 2665 – 2672, Hawaii.

Ozveren CM and Willsky AS 1990 Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control* **35**(7), 797–806.

Ramadge PJ 1986 Observability of discrete event systems *Conf. on Decision and Control*, pp. 1108–1112, Athens, Greece.

Rudie K, Lafortune S and Ling F 2003 Minimal communication in a distributed discrete event system. *IEEE Trans. on Automatic Control* **48**, 957–975.

S. Abramsky AJ 1994 *Domain Theory* Handbook of Logic in Computer Science Volume 3.

Santis ED, Benedetto MDD and Pola G 2003 On observability and detectability of continuous-time linear switching systems *Conf. on Decision and Control*, pp. 5777–5782.

Vidal R, Chiuso A and Soatto S 2002 Observability and identifiability of jump linear systems *Conf. on Decision and Control*, pp. 3614 – 3619, Las Vegas.

Vidal R, Chiuso A, Soatto S and Sastry S 2003 Observability of linear hybrid systems *Lecture Notes in Computer Science* 2623, O. Maler and A. Pnueli Eds. Springer Verlag, pp. 526–539.