Available online at www.sciencedirect.com
SCIENCE DIRECT
automatica
Automatica 42 (2006) 271 – 285
www.elsevier.com/locate/automatica

Discrete state estimators for systems on a lattice☆

Domitilla Del Vecchio[a,*], Richard M. Murray[b], Eric Klavins[c]

[a] *University of Michigan, Ann Arbor, USA*
[b] *California Institute of Technology, Pasadena, USA*
[c] *University of Washington, Seattle, USA*

Received 8 June 2004; received in revised form 6 April 2005; accepted 11 October 2005

Abstract

We address the problem of estimating discrete variables in a class of deterministic transition systems in which the continuous variables are available for measurement. We propose a novel approach to the estimation of discrete variables using lattice theory that overcomes some of the severe complexity issues encountered in previous work. The methodology proposed for the estimation of discrete variables is general as it is applicable to any observable system. Extensions generalize the approach to nondeterministic transition systems. The proposed estimator is finally constructed for a multi-robot system involving two teams competing against each other.

Keywords: Discrete state estimation; Lattice; Multi-agent systems; Computational complexity; Set representation

1. Introduction

In the last decade, hybrid system models have become very popular in the control community. The need for understanding the behavior of systems whose evolution is determined by the interplay of continuous dynamics and logic is compelling. In several applications, the coupling of continuous dynamics and decision protocols renders the system under study interesting and complicated enough that new mathematical tools are needed for the sake of analysis and control. Examples include the Internet, continuous plants controlled by digital controllers, multi-agent systems, biological systems, and many others. Issues such as controllability and observability arise naturally when trying to analyze the properties of these systems for control.

The problem of estimating and tracking the values of non-measurable variables in hybrid systems with reasonable computational effort is a challenging one. Bemporad, Ferrari-Trecate, and Morari (1999) show that observability properties are hard to check for hybrid systems and an observer is proposed that requires large amounts of computation. As a starting point, we consider the problem of estimating the discrete variable values when the continuous variables are available for measurement. This simplified scenario is already of practical interest as it is in the case of multi-robot systems. The continuous variables are quantities that we can measure directly, such as position and velocity, the discrete variables can represent the internal state of the decision and communication protocol that is used for coordination and control. We seek to construct a discrete state estimator with computational requirements comparable to that needed for simulating the system itself.

There is a wealth of research on observability and observer design for hybrid and discrete event systems. Bemporad et al. (1999) propose the notion of incremental observability for piecewise affine systems and propose a deadbeat observer that requires large amounts of computation. Balluchi, Benvenuti, Di Benedetto, and Sangiovanni-Vincentelli (2002) combine a *location* observer with a Luenberger observer to design hybrid observers that identify the location in a finite number of steps

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Rene Boel under the direction of Editor Ian Petersen.

* Corresponding author. Tel.: +1 734 761 4304; fax: +1 734 763 8041.
E-mail addresses: ddv@eecs.umich.edu (D. Del Vecchio), murray@cds.caltech.edu (R.M. Murray), klavins@ee.washington.edu (E. Klavins).

0005-1098/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2005.10.006

and converges 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, Benvenuti, Di Benedetto, and Sangiovanni-Vincentelli (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, Chiuso, and Soatto (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, Chiuso, Soatto, & Sastry, 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 test for observability, and Cassandra, Kaelbling, and Littman (1994) propose an approach to optimal control for partially observable Markov decision processes. For continuous time hybrid systems, De Santis, Di Benedetto, and Pola (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, which derive a test for current state observability. Oishi, Hwang, and Tomlin (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. Özveren and Willsky (1990), Caines, Greiner, and Wang (1991) and Caines and Wang (1995) propose discrete event observers based on the construction of the current-location observation tree that, as explored also in Del Vecchio and Klavins (2003), is impracticable when the number of locations is large, which is our case.

The main contribution of this paper is our approach to the estimation of the discrete variable values of a system (discrete state) that allows us to overcome some of the complexity issues encountered in previous work. In particular, given a system Σ whose discrete state needs to be estimated, we extend it to a lattice $(\chi, \leqslant)$, so that if the extended system $\tilde{\Sigma}$ and the lattice are *interval compatible*, an estimator $\hat{\Sigma}$ can be constructed that updates only two variables instead of an entire list of possible discrete states. These two variables are the lower and upper bounds of the set of possible discrete states compatible with the output sequence. In Section 2, we propose a multi-robot example to illustrate this idea. This approach to estimation is also general as it applies to any observable system in which the continuous variables are measured. In fact, we 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.

This paper is organized as follows. In Section 3, we review some basics on partial order theory and on observability. In Section 4, we formulate the problem that we seek to solve and a solution is proposed. Section 5 illustrates in detail the RoboFlag Drill system, its estimator is constructed, and complexity considerations are included. Section 6 proposes extensions to basic results that include the existence result for the estimator as well as the generalization of our arguments to nondeterministic systems.

2. Motivating example

As 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 $\{(x, y) \in \mathbb{R}^2 | y \leqslant 0\}$ from an equal number of incoming red robots (denoted by solid circles). The positions of the red robots are $(x_i, y_i) \in \mathbb{R}^2$. An example for eight robots is illustrated in Fig. 1. The red robots move straight toward the blue defensive zone. The blue robots are assigned each 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 $\alpha : \{1, \dots, N\} \to \{1, \dots, N\}$, 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 $(1, \dots, N)$ (see Klavins & Murray, 2004; 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 paper.

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

$$y_i(k+1) = y_i(k) - \delta \quad \text{if } y_i(k) \geqslant \delta, \tag{1}$$

$$z_i(k+1) = z_i(k) + \delta \quad \text{if } z_i(k) < x_{\alpha_i(k)}, \tag{2}$$

$$z_i(k+1) = z_i(k) - \delta \quad \text{if } z_i(k) > x_{\alpha_i(k)}, \tag{3}$$

$$(\alpha_i(k+1), \alpha_{i+1}(k+1)) = (\alpha_{i+1}(k), \alpha_i(k)) \quad \text{if } x_{\alpha_i(k)} \geqslant z_{i+1}(k) \wedge x_{\alpha_{i+1}(k)} \leqslant z_{i+1}(k), \tag{4}$$

where we assume $z_i \leqslant z_{i+1}$ 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).

Eq. (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 what has been called in previous work the location of the system (see Balluchi et al., 2002). The number of possible locations is $N!$, which, for $N \geqslant 8$, renders prohibitive the application of location observers

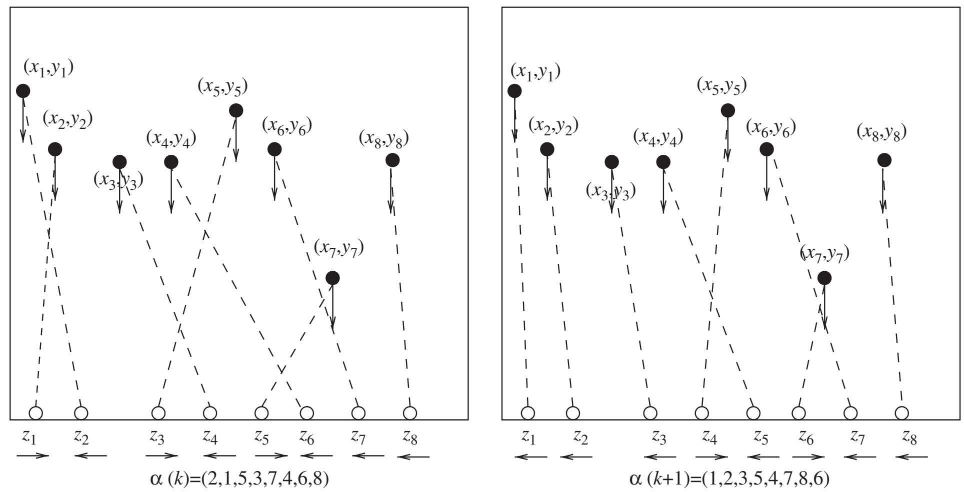

Fig. 1. Example of the RoboFlag Drill with eight robots per team.

based on the current-location observation tree as described in Caines et al. (1991) and used in Balluchi et al. (2002), Özveren and Willsky (1990), or discrete state observers based on similar concepts as the one in Del Vecchio and Klavins (2003). At each step, the set of possible α values compatible with the current output and with the previously seen outputs can be so large to render impractical its computation. As an example, we consider the situation depicted in Fig. 1 (left) where $N = 8$. We see the blue robots 1, 3, 5 going right and the others going left. From Eqs. (2)–(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 $\alpha_i \geqslant i + 1$ for $i \in \{1, 2, 3\}$ and $\alpha_i \leqslant i$ for $i \in \{2, 4, 6, 7, 8\}$. The size of this set is 40 320. According to the current-location observation tree method, this set needs to be mapped forward through the dynamics of the system to see what are the values of α 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 40 320 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 40 320. 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 Eqs. (2)–(3), denoted $O_y(k)$, can be represented as an interval with the order established component-wise, see the diagram in Fig. 2. The function $\tilde{f}$ that maps such a set forward, specified by Eqs. (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 Fig. 2, which can still be represented as an interval. When the new output measurement becomes available (Fig. 1, right) we obtain the new set $O_y(k+1)$ reported in Fig. 2. The sets $\tilde{f}(O_y(k))$ and $O_y(k+1)$ can be intersected by simply computing the maximum of their

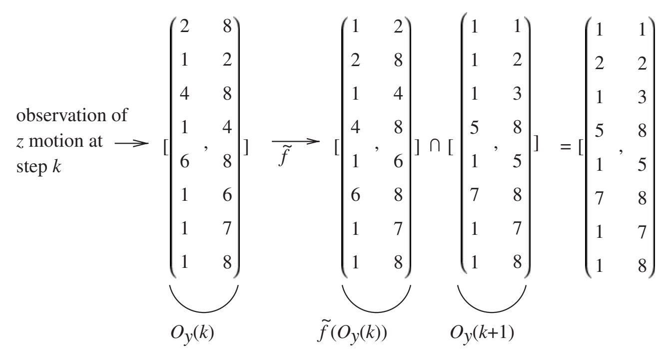

Fig. 2. The observation of the z motion at step k gives the set of possible α, $O_y(k)$. At each step, the set is described by the lower and upper bounds of a *interval sublattice* in an appropriately defined lattice. Such set is then mapped through the system dynamics ($\tilde{f}$) to obtain at step $k+1$ 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 $k+1$.

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:

$$L(k+1) = \tilde{f}(\max(L(k), \inf O_y(k))),$$

$$U(k+1) = \tilde{f}(\min(U(k), \sup O_y(k))). \tag{5}$$

As it will be shown in detail in the paper, the update laws in Eqs. (5) have, among others, the property that $[L(k), U(k)] \cap \text{perm}(N)$ tends to $\alpha(k)$. Letting $V(k) = |[L(k), U(k)] \cap \text{perm}(N)|$, Fig. 3 shows convergence plots $V(k)$ for the estimator compared to the convergence plots $E(k) = (1/N)\sum_{i=1}^{N} |\alpha_i(k) - i|$ of the assignment protocol to its equilibrium $(1, \ldots, N)$.

This example gives an idea of how complexity issues can be overcome with the aid of some partial order structure. In particular, the function $\tilde{f}$ has the property of preserving the

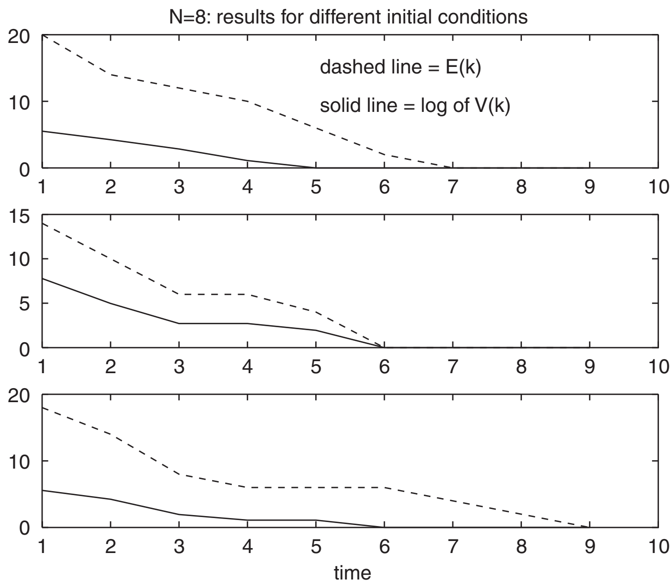

Fig. 3. Convergence plots for the estimator ($V(k)$) compared to the convergence plots of the assignment protocol to its equilibrium ($E(k)$).

interval structure of the sets of interest: this is a key property that allows to use only upper bounds and lower bounds for computation purposes. In a more general setting, one would like to know what are the properties of a system that allow such simplifications. By using partial order theory, which is introduced in the next section, we address this question.

3. Basic concepts

To construct the estimator introduced in the previous section, which updates lower and upper bounds of the set of all possible discrete variable values compatible with the output sequence, we make use of tools from partial order (or lattice) theory (Davey & Priesteley, 2002). The theory of partial orders, while standard in computer science, may be less well known to the intended audience of the paper. Therefore, we briefly review the basic definitions and notation we will use before proceeding to the main body of the paper.

3.1. Partial order theory

A partial order is a set χ with a partial order relation "$\leqslant$", and we denote it by the pair $(\chi, \leqslant)$. For any $x, w \in \chi$, the $\sup\{x, w\}$ is the smallest element that is larger than both x and w. In a similar way, the $\inf\{x, w\}$ is the largest element that is smaller than both x and w. We define the *join* "$\curlyvee$" and the *meet* "$\curlywedge$" of two elements x and w in χ as (1) $x \curlyvee w = \sup\{x, w\}$ and $x \curlywedge w = \inf\{x, w\}$; (2) if $S \subseteq \chi$, $\bigvee S = \sup S$ and $\bigwedge S = \inf S$.

Let $(\chi, \leqslant)$ be a partial order. If $x \curlywedge w \in \chi$ and $x \curlyvee w \in \chi$ for any $x, w \in \chi$, then $(\chi, \leqslant)$ is a *lattice*. In Fig. 4, we illustrate Hasse diagrams (Davey & Priesteley, 2002) showing partially ordered sets. From the diagram, 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.

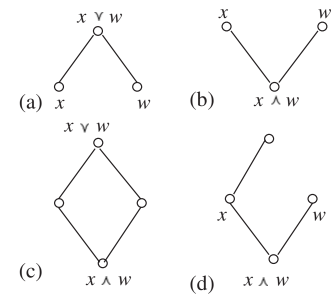

Fig. 4. In diagram (a) and (b), x and w are not related, but they have a join and a meet, respectively. In diagram (c), we show a complete lattice. In diagram (d), we show a partially ordered set that is not a lattice, since the elements x and w have a meet, but not a join.

Let $(\chi, \leqslant)$ be a partial order. Then, $(\chi, \leqslant)$ is a *chain* if for all $x, w \in \chi$, either $x \leqslant w$ or $w \leqslant x$, that is any two elements are comparable. If instead any two elements are not comparable, i.e. $x \leqslant y$ if and only if $x = y$, $(\chi, \leqslant)$ is said to be an *anti-chain*.

Let $(\chi, \leqslant)$ be a lattice and let $S \subseteq \chi$ be a nonempty subset of χ. Then, $(S, \leqslant)$ is a *sublattice* of χ if $a, b \in S$ implies that $a \curlyvee b \in S$ and $a \curlywedge b \in S$. If any sublattice of χ contains its least and greatest elements, then $(\chi, \leqslant)$ 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 (CPO). Given a complete lattice $(\chi, \leqslant)$, we will be concerned with a special kind of a sublattice called an *interval sublattice* defined as follows. Any interval sublattice of $(\chi, \leqslant)$ is given by $[L, U] = \{w \in \chi \mid L \leqslant w \leqslant U\}$ for $L, U \in \chi$. That is, this special sublattice can be represented by only two elements. For example, the intervals of $(\mathbb{R}, \leqslant)$ are just the familiar closed intervals on the real line.

Let $(\chi, \leqslant)$ be a lattice with least element $\perp$. Then, $a \in \chi$ is called an *atom* if $a > \perp$ and there is no element b such that $\perp < b < a$. The set of atoms of $(\chi, \leqslant)$ is denoted $\mathscr{A}(\chi, \leqslant)$.

The *power lattice* of a set $\mathscr{U}$, denoted $(\mathscr{P}(\mathscr{U}), \subseteq)$, is given by the power set of $\mathscr{U}$, $\mathscr{P}(\mathscr{U})$ (the set of all subsets of $\mathscr{U}$), 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 $\mathscr{U}$ itself, that is, $\top = \mathscr{U}$. Note that $\mathscr{A}(\mathscr{P}(\mathscr{U}), \subseteq) = \mathscr{U}$. An example is illustrated in Fig. 5. Given a set P, we denote by $|P|$ its cardinality. Next, we give some definitions about maps on partial orders.

Let $(P, \leqslant)$ and $(Q, \leqslant)$ be partially ordered sets. A map $f : P \to Q$ is (i) an *order preserving map* if $x \leqslant w \Rightarrow f(x) \leqslant f(w)$; (ii) an *order embedding* if $x \leqslant w \iff f(x) \leqslant f(w)$; (iii) an *order isomorphism* if it is order embedding and it maps P *onto* Q. If $(P, \leqslant)$ and $(Q, \leqslant)$ 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 $f(x \curlyvee w) = f(x) \curlyvee f(w)$ and $f(x \curlywedge w) = f(x) \curlywedge f(w)$.

Proposition 1 (*see Davey & Priesteley, 2002*). *If $f : P \to Q$ is a bijective homomorphism, then it is an order isomorphism.*

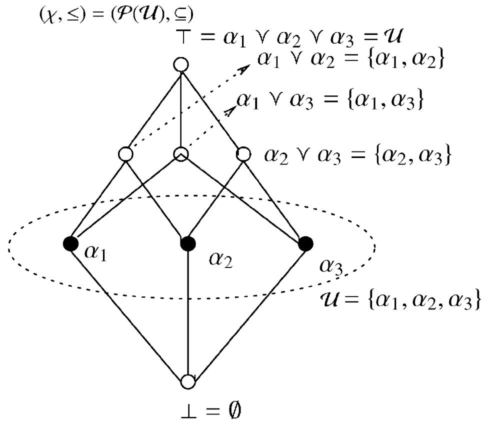

Fig. 5. Power lattice $(\chi, \leqslant)$ of a set $\mathscr{U}$ composed by three elements.

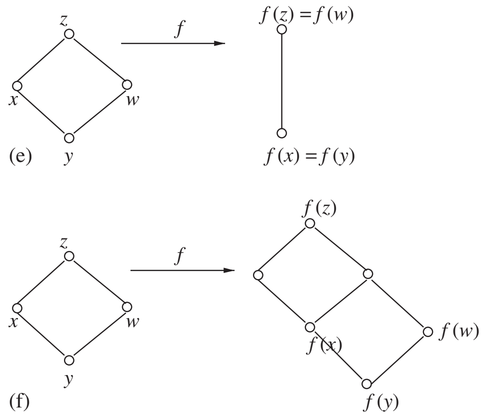

Fig. 6. Diagram (e) shows a map that is order preserving but not order embedding. Diagram (f) shows an order embedding that is not an order isomorphism: any two elements maintain the same order relation, but in between z and w there is nothing, while in between $f(z)$ and $f(w)$ some other elements appear (it is not onto).

Every order isomorphic map faithfully mirrors the structure of P onto Q. In Fig. 6 (right), we show some examples. The notion of order preserving map can be generalized to the case in which the map is nondeterministic, that is, it maps an element to a set of possible elements. With a slight abuse of the term "order preserving", we also make the following nonstandard definition. Let $x, w \in \chi$, with $(\chi, \leqslant)$ a lattice, $x \leqslant w$, and $f : \chi \to \mathscr{P}(\chi)$. We say that f is *order preserving* if $\bigvee f(x) \leqslant \bigvee f(w)$ and $\bigwedge f(x) \leqslant \bigwedge f(w)$.

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 3.1 (*Deterministic transition systems*). A *deterministic transition system* (DTS) is the tuple $\Sigma = (S, \mathscr{Y}, F, g)$, where

(i) S is a set of states with $s \in S$;
(ii) $\mathscr{Y}$ is a set of outputs with $y \in \mathscr{Y}$;
(iii) $F : S \to S$ is the state transition function;
(iv) $g : S \to \mathscr{Y}$ is the output function.

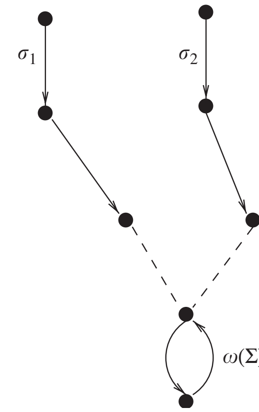

Fig. 7. Executions σ_2 and σ_3 are weakly equivalent according to Definition 3.5 while σ_1 is not weakly equivalent to either σ_2 or σ_3.

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 $\mathscr{E}(\Sigma)$.

Definition 3.2. Let $\Sigma = (S, \mathscr{Y}, F, g)$ be a deterministic transition system. The set $\Omega \subset S$ is the *ω^+-limit set* of Σ, denoted $\omega(\Sigma)$, if it is the smallest subset of S such that for all $\sigma = \{s(k)\}_{k \in \mathbb{N}}$

(i) If $s(k) \in \Omega$ and $s(k+1) = F(s(k))$, then $s(k+1) \in \Omega$.
(ii) For each $\sigma \in \mathscr{E}(\Sigma)$, there exists k_σ such that $\sigma(k_\sigma) \in \Omega$.

Definition 3.3. Given a deterministic transition system $\Sigma = (S, \mathscr{Y}, F, g)$, two executions $\sigma_1, \sigma_2 \in \mathscr{E}(\Sigma)$ are *distinguishable* if there exists a k such that $g(\sigma_1(k)) \neq g(\sigma_2(k))$.

Definition 3.4 (*Observability*). The deterministic transition system $\Sigma = (S, \mathscr{Y}, F, g)$ is said to be *observable* if any two different executions $\sigma_1, \sigma_2 \in \mathscr{E}(\Sigma)$ 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. However, there are systems for which two different initial states cannot be distinguished, but the states at some later step can. We introduce a weaker notion of observability analogous to *detectability* (Sontag, 1998) that accounts for this distinction.

Definition 3.5. Given a deterministic transition system $\Sigma = (S, \mathscr{Y}, F, g)$, two executions $\sigma_1, \sigma_2 \in \mathscr{E}(\Sigma)$ are *weakly equivalent*, denoted $\sigma_1 \sim \sigma_2$, if there exists k^* such that $\sigma_1(k^*) \notin \omega(\Sigma)$ and $\sigma_1(k) = \sigma_2(k)$ for all $k \geqslant k^*$.

In Fig. 7, we show examples of equivalent and not equivalent system executions.

Definition 3.6 (*Weak observability*). A deterministic transition system $\Sigma = (S, \mathscr{Y}, F, g)$ is *weakly observable* if whenever $\sigma_1 \nsim \sigma_2$ then there is k such that $g(\sigma_1(k)) \neq g(\sigma_2(k))$.

In the next section, we propose the estimator construction for observable systems, and in Section 6 we generalize the results obtained for observable systems to the case the system is weakly observable.

4. Estimator construction

In this section, we restrict the class of systems we consider to those in which the continuous variables are measurable. The discrete state estimation problem is then stated as the problem of finding suitable update laws for the upper and lower bounds of the set of all possible discrete variable values compatible with the output sequence. A solution to this problem is proposed in Theorem 4.1.

4.1. 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 $\Sigma=(S, \mathscr{Y}, F, g)$ we suppose that (i) $S=\mathscr{U}\times\mathscr{Z}$ with $\mathscr{U}$ a finite set and $\mathscr{Z}$ a finite dimensional space; (ii) $F=(f, h)$, where $f : \mathscr{U}\times\mathscr{Z} \to \mathscr{U}$ and $h : \mathscr{U}\times\mathscr{Z} \to \mathscr{Z}$; (iii) $y=g(\alpha, z) := z$, where $\alpha \in \mathscr{U}$, $z \in \mathscr{Z}$, $y \in \mathscr{Y}$, and $\mathscr{Y} = \mathscr{Z}$. The set $\mathscr{U}$ is a set of logic states and $\mathscr{Z}$ 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 2, $\mathscr{U}=\mathrm{perm}(N)$ and $\mathscr{Z}=\mathbb{R}^N$, the function f is represented by Eqs. (4) and the function h is represented by Eqs. (2)–(3). In the sequel, we will denote this class of DTS by $\Sigma=\mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ where we associate to the tuple $(\mathscr{U}, \mathscr{Z}, f, h)$, the equations:

$$\begin{aligned}\alpha(k+1) &= f(\alpha(k), z(k)),\\ z(k+1) &= h(\alpha(k), z(k)),\\ y(k) &= z(k),\end{aligned} \tag{6}$$

where $\alpha \in \mathscr{U}$ and $z \in \mathscr{Z}$. An execution of the system Σ in Eqs. (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 4.1 (*Deterministic transition system with input*). A deterministic transition system with input is a tuple $(S, \mathscr{I}, \mathscr{Y}, F, g)$ in which
(i) S is a set of states;
(ii) $\mathscr{I}$ is a set of inputs;
(iii) $\mathscr{Y}$ is a set of outputs;
(iv) $F : S \times \mathscr{I} \to S$ is a transition function;
(v) $g : S \to \mathscr{Y}$ is an output function.

In Problem 1 below, we specify what the elements of this tuple are when the DTS with input is a discrete state estimator of a DTS $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$. First, note that the set $\mathscr{U}$ does not have a natural metric associated with it. As a consequence, a way to track the value of α is to list, at each step k, the set of all possible α values that are compatible with the observation and with the system dynamics given in (6). This has been done already in Del Vecchio and Klavins (2003), for example, where the estimate is a list of possible values that the estimator has to update when a new measurement becomes available. This method leads to computational issues when the set to be listed is large.

In this paper, we propose an alternative to simply maintaining a list of all possible values for α. We propose to find a representation of the set so that the estimator updates the representation of the set rather than the whole set itself. In particular, if the set $\mathscr{U}$ can be immersed in a larger set χ whose elements can be related by an order relation $\leqslant$, we could represent a subset of $(\chi, \leqslant)$ as an interval sublattice $[L, U]$ (see Section 3.1). 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 $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$, find a deterministic transition system with input $\hat{\Sigma}=(\chi\times\chi, \mathscr{Z}\times\mathscr{Z}, \chi\times\chi, (f_1, f_2), \mathrm{id})$, with $f_1 : \chi\times\mathscr{Z}\times\mathscr{Z} \to \chi$, $f_2 : \chi\times\mathscr{Z}\times\mathscr{Z} \to \chi$, $\mathscr{U} \subseteq \chi$, with $(\chi, \leqslant)$ 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) $L(k)\leqslant\alpha(k)\leqslant U(k)$ (correctness).
(ii) $|[L(k+1), U(k+1)]|\leqslant|[L(k), U(k)]|$ (nonincreasing error).
(iii) There exists $k_0 > 0$ such that for any $k\geqslant k_0$ we have $[L(k), U(k)] \cap \mathscr{U} = \alpha(k)$ (convergence).

In the example shown in Section 2, we had that

$$f_1(L(k), y(k), y(k+1)) = \tilde{f}(\max(L(k), \inf O_y(k)))$$

$$f_2(U(k), y(k), y(k+1)) = \tilde{f}(\min(U(k), \sup O_y(k))),$$

where $O_y(k)$ is the set of possible α compatible with the output measurement at step k. Thus, in the following section we define the output sets O_y and we explain what are desirable properties of such sets, which will turn out to be interval sublattices. Also, in the example proposed we have $\mathscr{U} = \mathrm{perm}(N)$, and χ the set of vectors in $\mathbb{N}^N$ with components $x_i \in [1, N]$. The order is established component-wise, so that $(\chi, \leqslant)$ is a complete lattice. The function $\tilde{f}$ is defined on $(\chi, \leqslant)$, it coincides with f on $\mathscr{U}$, and it preserves the structure of the interval sublattices in $(\chi, \leqslant)$. With $\tilde{f}$, we extend the system defined on $\mathscr{U}$ to a system defined on χ. This extended system is going to be formally defined in the following section, and its desirable properties on the lattice $(\chi, \leqslant)$ will be introduced as well.

4.2. Problem solution

For finding a solution to Problem 1, we need to find the functions f_1 and f_2 defined on a lattice $(\chi, \leqslant)$ such that $\mathscr{U} \subseteq \chi$ for some finite lattice χ. We propose in the following definitions a way of extending a system Σ defined on $\mathscr{U}$ to a system $\tilde{\Sigma}$ defined on χ with $\mathscr{U} \subseteq \chi$. 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 $(\chi, \leqslant)$. 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 $(\chi, \leqslant)$ and the system $\tilde{\Sigma}$ to be interval compatible if such equivalence classes are interval sublattices and $\tilde{\Sigma}$ preserves their structure. On the basis of such notions, Theorem 4.1 below gives a possible solution to Problem 1.

Definition 4.2 (*Extended system*). Given the deterministic transition system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$, an *extension of* Σ *on* χ, with $\mathscr{U} \subseteq \chi$ and $(\chi, \leqslant)$ a complete lattice, is any system $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$, such that
(i) $\tilde{f} : \chi \times \mathscr{Z} \to \chi$ and $\tilde{f}|_{\mathscr{U}\times\mathscr{Z}} = f$;
(ii) $\tilde{h} : \chi \times \mathscr{Z} \to \mathscr{Z}$ and $\tilde{h}|_{\mathscr{U}\times\mathscr{Z}} = h$.

Definition 4.3 (*Transition sets*). Let $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ be a deterministic transition system. The nonempty sets $T_{(z^1,z^2)}(\tilde{\Sigma}) = \{w \in \chi | z^2 = \tilde{h}(w, z^1)\}$, for $z^1, z^2 \in \mathscr{Z}$, 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 $\tilde{h}$.

Definition 4.4 (*Transition classes*). The *set of* $\tilde{\Sigma}$-*transition classes* is given by $\mathscr{T}(\tilde{\Sigma}) = \{\mathscr{T}_1(\tilde{\Sigma}), \ldots, \mathscr{T}_M(\tilde{\Sigma})\}$, with $\mathscr{T}_i(\tilde{\Sigma})$ such that

(i) For any $\mathscr{T}_i(\tilde{\Sigma}) \in \mathscr{T}(\tilde{\Sigma})$ there are $z^1, z^2 \in \mathscr{Z}$ such that $\mathscr{T}_i(\tilde{\Sigma}) = T_{(z^1,z^2)}(\tilde{\Sigma})$.
(ii) For any $T_{(z^1,z^2)}(\tilde{\Sigma})$ there is $j \in \{1, \ldots, M\}$ such that $T_{(z^1,z^2)}(\tilde{\Sigma}) = \mathscr{T}_j(\tilde{\Sigma})$.

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 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 $\mathscr{T}_i(\tilde{\Sigma})$ for some i. Also, the transition classes $\mathscr{T}_i(\tilde{\Sigma})$ 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 $(\chi, \leqslant)$ in equivalence classes.

Definition 4.5 (*Output set*). Given the extension $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ of the deterministic transition system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ on the lattice $(\chi, \leqslant)$, 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 $\mathscr{T}_i(\tilde{\Sigma})$ for some $i \in \{1, \ldots, M\}$. The output set at step k is the set of all possible w values that are compatible with the pair $(y(k), y(k+1))$. By definition of the extended functions $(\tilde{h}|_{\mathscr{U}\times\mathscr{Z}} = h)$, this output set contains also all of the values of α compatible with the same output pair.

Definition 4.6 (*Interval compatibility*). Given the extension $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ of the system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ on the lattice $(\chi, \leqslant)$, the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is said to be *interval compatible* if

(i) Each $\tilde{\Sigma}$-transition class, $\mathscr{T}_i(\tilde{\Sigma}) \in \mathscr{T}(\tilde{\Sigma})$, is an interval sublattice of $(\chi, \leqslant)$, i.e., $\mathscr{T}_i(\tilde{\Sigma}) = [\bigwedge \mathscr{T}_i(\tilde{\Sigma}), \bigvee \mathscr{T}_i(\tilde{\Sigma})]$.
(ii) $\tilde{f} : (\mathscr{T}_i(\tilde{\Sigma}), z) \to [\tilde{f}(\bigwedge \mathscr{T}_i(\tilde{\Sigma}), z), \tilde{f}(\bigvee \mathscr{T}_i(\tilde{\Sigma}), z)]$ is an order isomorphism for any $i \in \{1, \ldots, M\}$ and for any $z \in \mathscr{Z}$.

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

Theorem 4.1. *Assume that the deterministic transition system* $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ *is observable. If there is a lattice* $(\chi, \leqslant)$, *such that the pair* $(\tilde{\Sigma}, (\chi, \leqslant))$ *is interval compatible*, *then the deterministic transition system with input* $\hat{\Sigma} = (\chi \times \chi, \mathscr{Z} \times \mathscr{Z}, \chi \times \chi, (f_1, f_2), \mathrm{id})$ *with*

$$f_1(L(k), y(k), y(k+1)) = \tilde{f}\left(L(k) \curlyvee \bigwedge O_y(k), y(k)\right),$$

$$f_2(U(k), y(k), y(k+1)) = \tilde{f}\left(U(k) \curlywedge \bigvee O_y(k), y(k)\right)$$

solves Problem 1.

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

$$L(k+1) = \tilde{f}\left(L(k) \curlyvee \bigwedge O_y(k), y(k)\right),$$
$$U(k+1) = \tilde{f}\left(U(k) \curlywedge \bigvee O_y(k), y(k)\right), \tag{7}$$

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 $L(0) \leqslant \alpha(0) \leqslant U(0)$. Induction step: we assume that $L(k) \leqslant \alpha(k) \leqslant U(k)$ and we show that $L(k+1) \leqslant \alpha(k+1) \leqslant U(k+1)$. Note that $\alpha(k) \in O_y(k)$. This, along with the assumption of the induction step, implies that

$$L(k) \curlyvee \bigwedge O_y(k) \leqslant \alpha(k) \leqslant U(k) \curlywedge \bigvee O_y(k).$$

Because we have that $L(k) \curlyvee \bigwedge O_y(k) \in O_y(k)$, and $U(k) \curlywedge \bigvee O_y(k) \in O_y(k)$, and the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible, we can use the isomorphic property of $\tilde{f}$ (property (ii) of Definition 4.6), which leads to

$$\tilde{f}\left(L(k) \curlyvee \bigwedge O_y(k)\right) \leqslant \alpha(k+1) \leqslant \tilde{f}\left(U(k) \curlywedge \bigvee O_y(k)\right).$$

This relationship combined with Eq. (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 Eq. (7), $w \in [L(k+1), U(k+1)]$ implies that

$$\tilde{f}\left(L(k) \curlyvee \bigwedge O_y(k)\right) \leqslant w \leqslant \tilde{f}\left(U(k) \curlywedge \bigvee O_y(k)\right). \tag{8}$$

In addition, we have that

$$\bigwedge O_y(k) \leqslant L(k) \curlyvee \bigwedge O_y(k)$$

and

$$U(k) \curlywedge \bigvee O_y(k) \leqslant \bigvee O_y(k).$$

Because the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible, by virtue of the isomorphic property of $\tilde{f}$ (property (ii) of Definition 4.6), we have that

$$\tilde{f}\left(\bigwedge O_y(k)\right) \leqslant \tilde{f}\left(L(k) \curlyvee \bigwedge O_y(k)\right)$$

and

$$\tilde{f}\left(U(k) \curlywedge \bigvee O_y(k)\right) \leqslant \tilde{f}\left(\bigvee O_y(k)\right).$$

This, along with relations (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 (8) implies that

$$L(k) \curlyvee \bigwedge O_y(k) \leqslant z \leqslant U(k) \curlywedge \bigvee O_y(k),$$

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

Proof of (iii): We proceed by contradiction. Thus, assume that for any k_0 there exists a $k \geqslant k_0$ such that $\{\alpha(k), \beta_k\} \subseteq [L(k), U(k)] \cap \mathscr{U}$ for some $\beta_k \neq \alpha(k)$ and $\beta_k \in \mathscr{U}$. 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 $\beta_{k-1} \in [L(k-1), U(k-1)] \cap \mathscr{U}$. If this is not the case, we can construct an infinite sequence $\{k_i\}_{i \in \mathbb{N}^+}$ such that $\beta_{k_i} \in [L(k_i), U(k_i)] \cap \mathscr{U}$ with $\beta_{k_i} = \tilde{f}(\beta_{k_i - 1})$ and $\beta_{k_i - 1} \in [L(k_i - 1), U(k_i - 1)] \cap (\chi - \mathscr{U})$. Notice that $|[L(k_1 - 1), U(k_1 - 1)] \cap (\chi - \mathscr{U})| = M < \infty$. Also, we have

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

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

Thus, for any k_0 there is $k \geqslant k_0$ such that $\{\alpha(k), \beta_k\} \subseteq [L(k), U(k)] \cap \mathscr{U}$, with $\beta_k = f(\beta_{k-1})$ for some $\beta_{k-1} \in [L(k-1), U(k-1)] \cap \mathscr{U}$. 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 $\{\beta_{k-1}, z(k-1)\}_{k \geqslant \bar{k}} = \sigma_1$ and $\{\alpha(k-1), z(k-1)\}_{k \geqslant \bar{k}} = \sigma_2$ are two executions of Σ sharing the same output. This contradicts the observability assumption. □

Corollary 4.1. *If the extended system $\tilde{\Sigma}$ of an observable system Σ is observable, then the estimator $\hat{\Sigma}$ given in Theorem* 4.1 *solves Problem* 1 *with $L(k) = U(k) = \alpha(k)$ for $k \geqslant k_0$.*

Proof. The proof proceeds by contradiction. Assume that for any $k_0 \geqslant 0$ there is $k \geqslant k_0$ such that $\{\alpha(k), \beta_k\} \subseteq [L(k), U(k)]$ for some β_k. By the proof of (ii) of Theorem 4.1, we have that $\beta_k = \tilde{f}(\beta_{k-1})$ for $\beta_{k-1} \in [L(k-1), U(k-1)]$ and $\beta_{k-1} \in O_y(k-1)$. Thus, $\sigma_1 = \{\beta_{k-1}, z(k-1)\}_{k \in \mathbb{N}}$ and $\sigma_2 = \{\alpha(k-1), z(k-1)\}_{k \in \mathbb{N}}$ are two executions of $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$that share the same output sequence. This contradicts the observability of the system $\tilde{\Sigma}$. □

An example in which Theorem 4.1 holds but Corollary 4.1 does not is provided by the RoboFlag Drill introduced in Section 2. In fact, if we allow the assignments to be in $\mathbb{N}^N$, there are different executions compatible with the same output sequence.

5. Example: the RoboFlag Drill

The RoboFlag Drill has been described in Section 2. In this section, we revisit it by finding a lattice and a system extension that can be used for constructing the estimator proposed in Theorem 4.1. We define $x = (x_1, \ldots, x_N)$, $z = (z_1, \ldots, z_N)$, $\alpha = (\alpha_1, \ldots, \alpha_N)$. The complete RoboFlag specification is given by the program in rules (1)–(4). In particular, the rules in (2)–(3) model the function $h : \mathscr{U} \times \mathscr{Z} \to \mathscr{Z}$ that updates the continuous variables, and the rules in (4) model the function $f : \mathscr{U} \times \mathscr{Z} \to \mathscr{U}$ that updates the discrete variables. In this example, we have $\mathscr{U} = \text{perm}(N)$ the set of permutations of N elements, and $\mathscr{Z} = \mathbb{R}^N$. Thus, the RoboFlag system is given by $\Sigma = \mathscr{S}(\text{perm}(N), \mathbb{R}^N, f, h)$, and the variables $z \in \mathbb{R}^N$ are measured. The variables x are treated as known parameters.

Problem 2 (*RoboFlag Drill Observation Problem*). Given initial values for x and y and the values of z corresponding to an execution of $\Sigma = \mathscr{S}(\text{perm}(N), \mathbb{R}^N, f, h)$, determine the value of α during that execution.

It can be shown that the system $\Sigma = \mathscr{S}(\text{perm}(N), \mathbb{R}^N, f, h)$ reported in rules (1)–(4) with measured variable z is observable (the interested reader is deferred to Del Vecchio & Murray, 2004, for details). Then, the estimator is constructed in the following section.

5.1. RoboFlag Drill estimator

In this section, we construct the estimator proposed in Theorem 4.1 in order to estimate and track the value of the assignment α in any execution. To accomplish this, we find a lattice $(\chi, \leqslant)$ in which to immerse the set $\mathscr{U}$ and an extension $\tilde{\Sigma}$ of the system Σ to χ, so that the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible.

We first construct a lattice $(\chi, \leqslant)$ and the extended system $\tilde{\Sigma}=\mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ such that $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible. We choose as χ the set of vectors in $\mathbb{N}^N$ with coordinates $x_i \in [1, N]$, that is, $\chi = \{x \in \mathbb{N}^N \mid x_i \in [1, N]\}$. For the elements in χ, we use the vector notation, that is $x = (x_1, \dots, x_N)$. The partial order that we choose on such a set is given by

$$\forall x, w \in \chi, \quad x \leqslant w \ \text{ if } x_i \leqslant w_i \ \ \forall i. \tag{9}$$

As a consequence, the join and the meet between any two elements $x, w \in \chi$ are given by $v = x \curlyvee w$ if $v_i = \max\{x_i, w_i\}$, and $v = x \curlywedge w$ if $v_i = \min\{x_i, w_i\}$. With this choice,we have $\bigvee \chi = (N, \dots, N)$ and $\bigwedge \chi = (1, \dots, 1)$. The pair $(\chi, \leqslant)$ with the order defined by (9) is clearly a lattice. The set $\mathscr{U}$ is the set of all permutations of N elements and it is a subset of χ. All of the elements in $\mathscr{U}$ form an anti-chain of the lattice, that is, any two elements of $\mathscr{U}$ are not related by the order in $(\chi, \leqslant)$. In the sequel, we will denote by w any variable in χ not specifying if it is in $\mathscr{U}$, and we will denote by α any variable in $\mathscr{U}$.

The function $h : \mathrm{perm}(N) \times \mathbb{R}^N \to \mathbb{R}^N$ can be naturally extended to χ as

$$z_i(k+1) = z_i(k) + \delta \quad \text{if } z_i(k) < x_{w_i(k)},$$

$$z_i(k+1) = z_i(k) - \delta \quad \text{if } z_i(k) > x_{w_i(k)} \tag{10}$$

for $w \in \chi$. The rules (10) specify $\tilde{h} : \chi \times \mathbb{R}^N \to \mathbb{R}^N$, and one can check that $\tilde{h}|_{\mathscr{U}\times\mathscr{Z}} = h$. In analogous way $f : \mathrm{perm}(N) \times \mathbb{R}^N \to \mathrm{perm}(N)$ is extended to χ as

$$(w_i(k+1), w_{i+1}(k+1)) = (w_{i+1}(k), w_i(k))$$
$$\text{if } x_{w_i(k)} \geqslant z_{i+1}(k) \wedge x_{w_{i+1}(k)} \leqslant z_{i+1}(k), \tag{11}$$

for $w \in \chi$. The rules (11) model the function $\tilde{f} : \chi \times \mathbb{R}^N \to \chi$, and one can check that $\tilde{f}|_{\mathscr{U}\times\mathscr{Z}} = f$. Therefore, the system $\tilde{\Sigma} = (\tilde{f}, \tilde{h}, \chi, \mathbb{R}^N)$ is the extended system of $\Sigma = (f, h, \mathrm{perm}(N), \mathbb{R}^N)$ (see Definition 4.2). One can show that the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible (the proof can be found in Del Vecchio, 2005).

The estimator $\hat{\Sigma} = (\chi \times \chi, \mathscr{Z} \times \mathscr{Z}, \chi \times \chi, (f_1, f_2), \mathrm{id})$ given in Theorem 4.1 can be constructed because the hypotheses of the theorem are satisfied. The estimator $\hat{\Sigma}$ can be specified by the following rules:

$$l_i(k+1) = i+1 \quad \text{if } z_i(k+1) = z_i(k) + \delta, \tag{12}$$

$$l_i(k+1) = 1 \quad \text{if } z_i(k+1) = z_i(k) - \delta, \tag{13}$$

$$L_{i,y}(k+1) = \max\{L_i(k), l_i(k+1)\}, \tag{14}$$

$$(L_i(k+1), L_{i+1}(k+1)) = (L_{i+1,y}(k+1), L_{i,y}(k+1))$$
$$\text{if } x_{L_{i,y}(k+1)} \geqslant z_{i+1}(k) \wedge x_{L_{i+1,y}(k+1)} \leqslant z_{i+1}(k), \tag{15}$$

$$u_i(k+1) = N \quad \text{if } z_i(k+1) = z_i(k) + \delta, \tag{16}$$

$$u_i(k+1) = i \quad \text{if } z_i(k+1) = z_i(k) - \delta, \tag{17}$$

$$U_{i,y}(k+1) = \min\{U_i(k), u_i(k+1)\}, \tag{18}$$

$$(U_i(k+1), U_{i+1}(k+1)) = (U_{i+1,y}(k+1), U_{i,y}(k+1))$$
$$\text{if } x_{U_{i,y}(k+1)} \geqslant z_{i+1}(k) \wedge x_{U_{i+1,y}(k+1)} \leqslant z_{i+1}(k) \tag{19}$$

initialized with $L(0)=\bigwedge \chi$ and $U(0)=\bigvee \chi$. Rules (12)–(13) and (16)–(17) take the output information z and set the lower and upper bound of $O_y(k)$, respectively. Rules (14) and (18) compute the lower and upper bound of the intersection $[L(k), U(k)] \cap O_y(k)$, respectively. Finally, rules (15) and (19) compute the lower and upper bound of the set $\tilde{f}([L(k), U(k)] \cap O_y(k))$, respectively.

5.2. Complexity of the RoboFlag Drill estimator

The amount of computation required for updating L and U according to (12)–(19) 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 $\mathbb{N}$, in which N is the number of robots. Therefore, the complexity of the algorithm that generates the sequences $L(k)$ and $U(k)$ is proportional to N, i.e., it is the same as the complexity of the algorithm that generates the α trajectories. Also, note that the rules in (12)–(19) are obtained by "copying" the rules in (11) and correcting them by means of the output information, according to how the Kalman filter or the Luenberger observer are constructed for dynamical systems (see the seminal paper by Kalman, 1960, and by Luenberger, 1971).

As established by property (iii) of Problem 1, the function of k given by $|[L(k), U(k)] \cap \mathscr{U} - \alpha(k)|$ tends to zero. This function is useful for analysis purposes, but it is not necessary to compute it at any point in the algorithm proposed in Eqs. (12)–(19). However, since $L(k)$ does not converge to $U(k)$, once the algorithm has converged, i.e., when $|[L(k), U(k)] \cap \mathscr{U}| = 1$, we cannot find the value of $\alpha(k)$ from the values of $U(k)$ and $L(k)$ directly. Instead of computing directly $[L(k), U(k)] \cap \mathscr{U}$, we carry out a simple algorithm, that 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.

Refinement algorithm. Let $c_i = [L_i, U_i]$. Then the algorithm

$$(m_1, \dots, m_N) = \mathit{Refine}(c_1, \dots, c_N),$$

which takes assignment sets $c_1, \dots, c_N$ and produces assignment sets $m_1, \dots, m_N$, is such that if $m_i = \{k\}$ then $k \notin m_j$ for any $j \neq i$.

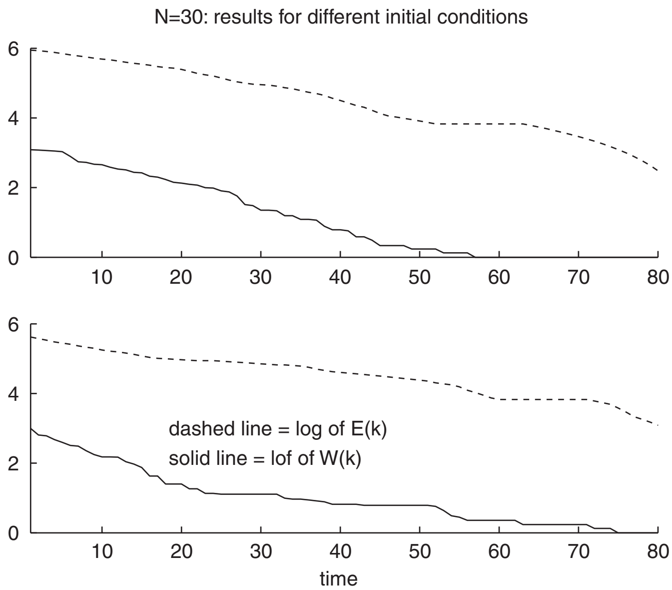

Fig. 8. Example with $N = 30$: note that the function $W(k)$ is always nonincreasing and its logarithm is converging to zero.

This algorithm takes as input the sets m_i and removes singletons occurring at one coordinate set from all of the other coordinate sets. 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 $[L, U] \cap \mathscr{U} = \alpha$. Second, the cardinality of the sets $m_i(k)$ is nonincreasing with the time step k. A formal proof of these properties can be found in Del Vecchio (2005).

The case in which the robots are in a space with dimension higher than one, for example $z_i \in \mathbb{R}^3$, is an interesting case to explore. In principle, nothing in the estimation algorithm structure would change provided that the direction of motion of a defender defines a set of possible attackers to which it is assigned to.

5.3. Simulation results

The RoboFlag Drill system represented in rules (2)–(4) has been implemented in MATLAB together with the estimator reported in the rules (12)–(19). Fig. 3 in Section 2 showed the behavior of the quantities $V(k) = |[L(k), U(k)] \cap \mathscr{U}|$ and $E(k) = (1/N)\sum_{i=1}^{N} |\alpha_i(k) - i|$. $V(k)$ represents the cardinality of the set of all possible assignments at each step. This quantity gives an idea of the convergence rate of the estimator. $E(k)$ is a function of α, and it is not increasing along the executions of the system $\Sigma = \mathscr{S}(\text{perm}(N), \mathbb{R}^N, f, h)$. This quantity is showing the rate of convergence of the α assignment to its equilibrium $(1, \ldots, N)$. In Fig. 8, we show the results for $N = 30$ robots per team. In particular, we report the log of $E(k)$ and the log of $W(k)$ defined as $W(k) = (1/N)\sum_{i=1}^{N} |m_i(k)|$, which is non increasing and converging to one, that is the sets $(m_1(k), \ldots, m_N(k))$ converge to $\alpha(k) = (\alpha_1(k), \ldots, \alpha_N(k))$. In the same figure, we notice that when $W(k)$ converges to one, $E(k)$ has not converged to zero yet. This suggests that the estimator is faster than the dynamics of the system under study. We cannot explain such a good performance formally yet, and the estimator speed issue will be addressed in future work.

In the previous sections, we proposed an estimator $\hat{\Sigma} = (\chi \times \chi, \mathscr{Z} \times \mathscr{Z}, \chi \times \chi, (f_1, f_2), \text{id})$ on a lattice $(\chi, \leqslant)$ for a DTS $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ with $\mathscr{U} \subseteq \chi$. Such an estimator can be constructed if the system Σ is observable and if the extended system $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ is such that the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible. In the next section, we investigate when the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible, and what are possible causes of the estimator complexity.

6. Extensions to basic results

In this section, we give a characterization of what observable means in terms of extensibility of a system into an extended system that is interval compatible with a lattice $(\chi, \leqslant)$. We show that if the system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ is observable, there always exists a lattice $(\chi, \leqslant)$ such that the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible. The worst case size of the lattice is computed, which gives a computational burden equivalent to the observer tree approach. The main advantage of this method from a computational standpoint is clear when a lattice with algebraic structure can be found, in which the sup and inf can be computed exploiting the algebra. Thus, we show a possible way of constructing the estimator on a chosen lattice by constructing a nondeterministic extension of Σ on χ. The previous section results are then generalized to nondeterministic systems.

6.1. Estimator existence

For the deterministic transition system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$, the Σ-transition sets and the Σ-transition classes are defined as for the extended system $\tilde{\Sigma} = \mathscr{S}(\mathscr{U}, \mathscr{Z}, \tilde{f}, \tilde{h})$ in Definitions 4.3 and in 4.4, respectively, by replacing $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ with $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$. Each Σ-transition set $T_{(z^1,z^2)}(\Sigma)$ contains all of α values in $\mathscr{U}$ that allow the transition from z^1 to z^2 through the function h. Note also that for any $z^1, z^2 \in \mathscr{Z}$ we have $T_{(z^1,z^2)}(\Sigma) \subseteq T_{(z^1,z^2)}(\tilde{\Sigma})$ because $\tilde{h}|_{\mathscr{U}\times\mathscr{Z}} = h$ and $\mathscr{U} \subseteq \chi$. This in turn implies that $\mathscr{T}_j(\Sigma) \subseteq \mathscr{T}_j(\tilde{\Sigma})$.

We also assume that all of the executions contained in the ω^+-limit set of Σ, $\omega(\Sigma)$, are distinguishable. More formally we have:

Assumption 6.1. The ω^+-limit set of $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$, $\omega(\Sigma)$, is such that for any two different executions σ_1, σ_2 with $\sigma_1(0), \sigma_2(0) \in \omega(\Sigma)$ there is $k \in \mathbb{N}$ such that $\sigma_1(k)(z) \neq \sigma_2(k)(z)$.

Lemma 6.1. *Consider the deterministic transition system* $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$. *Let* $\omega(\Sigma)$ *verify Assumption* 6.1. *Then* Σ *is observable if and only if* $f : (\mathscr{T}_j(\Sigma), z) \to f(\mathscr{T}_j(\Sigma), z)$ *is one to one for any* $j \in \{1, \ldots, M\}$ *and for any* $z \in \mathscr{Z}$.

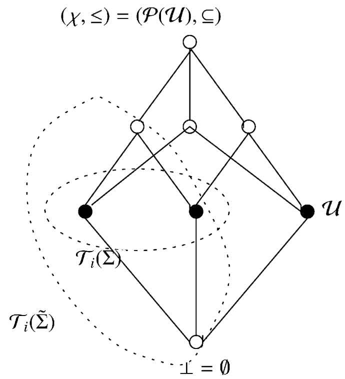

Fig. 9. Example of the Σ and $\tilde{\Sigma}$ transition classes with $\mathscr{U}$ (dark elements) composed of three elements.

The proof of this lemma can be found in Del Vecchio (2005). This lemma shows that observability can be determined by checking if the function f is one to one on the Σ-transition classes $\mathscr{T}_j(\Sigma)$, provided that the executions evolving in $\omega(\Sigma)$ are distinguishable. This lemma is used in the following theorem, which gives an alternative characterization of what observable means in terms of extensibility of the system Σ into a system $\tilde{\Sigma}$ that is interval compatible with a lattice $(\chi, \leqslant)$.

Theorem 6.1 (*Observability on bounded lattices*). *Consider the deterministic transition system* $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$. *Let* $\omega(\Sigma)$ *verify Assumption* 6.1. *Then the following are equivalent*:

(i) *System* Σ *is observable.*
(ii) *There exist a complete lattice* $(\chi, \leqslant)$ *with* $\mathscr{U} \subseteq \chi$, *such that the extension* $\tilde{\Sigma} = (\tilde{f}, \tilde{h}, \chi, \mathscr{Z})$ *of* Σ *on* χ *is such that* $(\tilde{\Sigma}, (\chi, \leqslant))$ *is interval compatible.*

Proof. ((i) $\Rightarrow$ (ii)) We show the existence of a lattice $(\chi, \leqslant)$ and of an extended system $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ with $(\tilde{\Sigma}, (\chi, \leqslant))$ an interval compatible pair by construction. Define $\chi := \mathscr{P}(\mathscr{U})$, and $(\chi, \leqslant) := (\mathscr{P}(\mathscr{U}), \subseteq)$.

To define $\tilde{h}$, we define the sublattices $(\mathscr{T}_i(\tilde{\Sigma}), \leqslant)$ of $(\chi, \leqslant)$ for $i \in \{1, \dots, M\}$, by $(\mathscr{T}_i(\tilde{\Sigma}), \leqslant) := (\mathscr{P}(\mathscr{T}_i(\Sigma)), \subseteq)$ as shown in Fig. 9. As a consequence, for any given $z^1, z^2 \in \mathscr{Z}$ such that $z^2 = h(\alpha, z^1)$ for $\alpha \in \mathscr{T}_i(\Sigma)$ for some i, we define $z^2 = \tilde{h}(w, z^1)$ for any $w \in \mathscr{T}_i(\tilde{\Sigma})$. Clearly, $\tilde{h}|_{\mathscr{U} \times \mathscr{Z}} = h$, and $\mathscr{T}_i(\tilde{\Sigma})$ for any i is an interval sublattice of the form $\mathscr{T}_i(\tilde{\Sigma}) = [\perp, \bigvee \mathscr{T}_i(\tilde{\Sigma})]$.

The function $\tilde{f}$ is defined in the following way. For any $x, w \in \chi$ and $\alpha \in \mathscr{U}$ we have

$$\begin{cases} \tilde{f}(x \curlyvee w) = \tilde{f}(x) \curlyvee \tilde{f}(w), \\ \tilde{f}(x \curlywedge w) = \tilde{f}(x) \curlywedge \tilde{f}(w), \\ \tilde{f}(\perp) = \perp, \\ \tilde{f}(\alpha) = f(\alpha), \end{cases} \tag{20}$$

where we have omitted the dependency on the z variables for simplifying notation. We prove first that $\tilde{f} : \mathscr{T}_i(\tilde{\Sigma}) \to [\perp, \tilde{f}(\bigvee \mathscr{T}_i(\tilde{\Sigma}))]$ is onto. We have to show that for any $w \neq \perp \in [\perp, \tilde{f}(\bigvee \mathscr{T}_i(\tilde{\Sigma}))]$ there is $x \in [\perp, \bigvee \mathscr{T}_i(\tilde{\Sigma})]$ such that $w = \tilde{f}(x)$. Since $\bigvee \mathscr{T}_i(\tilde{\Sigma}) = \alpha_1 \curlyvee \cdots \curlyvee \alpha_p$ for $\{\alpha_1, \dots, \alpha_p\} = \mathscr{T}_i(\Sigma)$, we have also that $\tilde{f}(\bigvee \mathscr{T}_i(\tilde{\Sigma})) = f(\alpha_1) \curlyvee \cdots \curlyvee f(\alpha_p)$ by virtue of Eqs. (20). Because $w \leqslant \tilde{f}(\bigvee \mathscr{T}_i(\tilde{\Sigma}))$, we have that $w = f(\alpha_{j_1}) \curlyvee \cdots \curlyvee f(\alpha_{j_m})$ for $j_k \in \{1, \dots, p\}$ and $m < p$. This in turn implies, by Eqs. (20), that $w = \tilde{f}(\alpha_{j_1} \curlyvee \cdots \curlyvee \alpha_{j_m})$. Since $x := \alpha_{j_1} \curlyvee \cdots \curlyvee \alpha_{j_m} < \bigvee \mathscr{T}_i(\tilde{\Sigma})$, we have proved that $w = \tilde{f}(x)$ for $x \in \mathscr{T}_i(\tilde{\Sigma})$. Second, we notice that $\tilde{f} : \mathscr{T}_i(\tilde{\Sigma}) \to [\perp, \tilde{f}(\bigvee \mathscr{T}_i(\tilde{\Sigma}))]$ is one to one because of Lemma 6.1. Thus, we have proved that $\tilde{f} : \mathscr{T}_i(\tilde{\Sigma}) \to [\perp, \tilde{f}(\bigvee \mathscr{T}_i(\tilde{\Sigma}))]$ is a bijection, and by Eqs. (20) it is also an homomorphism. We then apply Proposition 1 to obtain the result.

((ii) $\Rightarrow$ (i)). To show that (ii) implies that $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ is observable, we apply Lemma 6.1. In particular, $(\tilde{\Sigma}, (\chi, \leqslant))$ being interval compatible implies that $\tilde{f} : \mathscr{T}_i(\tilde{\Sigma}) \to [\tilde{f}(\bigwedge \mathscr{T}_i(\tilde{\Sigma})), \tilde{f}(\bigvee \mathscr{T}_i(\tilde{\Sigma}))]$ is one to one for any i. This, along with Assumption 6.1, by Lemma 6.1 imply that the system is observable. □

This result links the property of a pair $(\tilde{\Sigma}, (\chi, \leqslant))$ being interval compatible with the observability properties of the original system Σ.

Theorem 6.1 shows that an observable system admits a lattice and a system extension that satisfy interval compatibility by constructing them, in a similar way as one shows that a stable dynamical system has a Lyapunov function. However, the constructed lattice is impractical for the implementation of the estimator of Theorem 4.1 when the size of $\mathscr{U}$ is large because the size of the representation of the elements of χ is large as well. The worst case size of χ is computed by the following proposition.

Proposition 2. *Consider the system* $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$, *with* $f : \mathscr{U} \to \mathscr{U}$. *Assume that the sets* $\{\mathscr{T}_1(\Sigma), \dots, \mathscr{T}_m(\Sigma)\}$ *are all disjoint. Then* $|\chi| \leqslant 2|\mathscr{U}|^2$.

The proof of this proposition can be found in Del Vecchio (2005). 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 $\mathscr{P}(\mathscr{U})$ is 16 778 238, while the worst case size given in Proposition 2 is 576, and the size of the lattice χ proposed in Section 5.1 is $4^4 = 256$. Thus, the estimate given by Proposition 2 significantly reduces the size of χ given by $\mathscr{P}(\mathscr{U})$. Note that the size of the lattice proposed in Section 5.1 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).

This proposition shows that the worst case computation needed for implementing our estimator is the same as the one needed in Caines et al. (1991), where the observer tree method is proposed. The main advantage of this method is clear when the space of discrete variables can be immersed in a lattice whose order relations can be computed algebraically ($(\chi, \leqslant)$ does not need to be stored). We then consider the case in which there is a preferred lattice structure $(\chi, \leqslant)$ in which the order relations can be computed algebraically, but there is no system extension $\tilde{\Sigma}$ such that the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is interval compatible. We thus look for an over-approximation of the system Σ that might be interval compatible with the desired lattice $(\chi, \leqslant)$. Such an over-approximation is called a weakly equivalent generalization and is defined the following way.

Definition 6.1. Consider the deterministic transition system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$. We define $\Sigma_{\geqslant} = \mathscr{S}(\mathscr{U}_{\geqslant}, \mathscr{Z}, f_{\geqslant}, h)$ to be a *Σ-weakly equivalent generalization* of Σ on $\mathscr{U}_{\geqslant}$ with $\mathscr{U} \subseteq \mathscr{U}_{\geqslant}$ if

(i) $\mathscr{E}(\Sigma) \subseteq \mathscr{E}(\Sigma_{\geqslant})$.
(ii) Any $\sigma_{\Sigma_{\geqslant}} \in \mathscr{E}(\Sigma_{\geqslant})$ such that $\{\sigma_{\Sigma_{\geqslant}}(k)(z)\}_{k\in\mathbb{N}} = \{\sigma_{\Sigma}(k)(z)\}_{k\in\mathbb{N}}$, for some execution $\sigma_{\Sigma} \in \mathscr{E}(\Sigma)$, is such that $\sigma_{\Sigma_{\geqslant}} \sim \sigma_{\Sigma}$.

Item (i) establishes that $\Sigma_{\geqslant}$ is a generalization of Σ, denoted $\Sigma \subseteq \Sigma_{\geqslant}$. Moreover, (ii) establishes that those executions of $\Sigma_{\geqslant}$ that have the same output sequence as one of the executions, σ_{Σ}, of Σ are equivalent to σ_{Σ}. As a consequence, if the system Σ is observable (or weakly observable), its Σ-weakly equivalent generalization $\Sigma_{\geqslant}$ is weakly observable on the set of executions of Σ. For weakly observable systems, Theorem 4.1 can be applied by substituting the assumption of the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ being interval compatible with a weaker assumption that we call *weak interval compatibility* defined as follows.

Definition 6.2 (*Weak interval compatibility*). Consider the extended system $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ of $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ on $(\chi, \leqslant)$. The pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is said to be *weakly interval compatible* if

(i) Each $\tilde{\Sigma}$-transition class, $\mathscr{T}_i(\tilde{\Sigma}) \in \mathscr{T}(\tilde{\Sigma})$, is an interval sublattice of $(\chi, \leqslant)$, i.e., $\mathscr{T}_i(\tilde{\Sigma}) = [\bigwedge \mathscr{T}_i(\tilde{\Sigma}), \bigvee \mathscr{T}_i(\tilde{\Sigma})]$.
(ii) $\tilde{f} : ([L, U], z) \longrightarrow [\tilde{f}(L, z), \tilde{f}(U, z)]$ is order preserving for any $[L, U] \subseteq \mathscr{T}_i(\tilde{\Sigma})$, and any $z \in \mathscr{Z}$ and for any $i \in \{1, \ldots, M\}$.
(iii) $\tilde{f} : ([L, U], z) \longrightarrow [\tilde{f}(L, z), \tilde{f}(U, z)]$ is onto for any $[L, U] \subseteq \mathscr{T}_i(\tilde{\Sigma})$ for any $z \in \mathscr{Z}$ and for any $i \in \{1, \ldots, M\}$.

We have a difference between observable systems and weakly observable systems because in a weakly observable system, two executions sharing the same output can collapse one onto the other, thus there cannot be any extension $\tilde{f}$ that is a bijection between the output lattice and the set it is mapped to. Thus, we can restate Theorem 4.1 for weakly observable systems in the following way.

Theorem 6.2. *Assume that the deterministic transition system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ is weakly observable. If there is a lattice $(\chi, \leqslant)$, such that the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is weakly interval compatible, then the deterministic transition system with input $\hat{\Sigma} = (\chi \times \chi, \mathscr{Z} \times \mathscr{Z}, \chi \times \chi, (f_1, f_2), \mathrm{id})$ with*

$$f_1(L(k), y(k), y(k+1)) = \tilde{f}\left(L(k) \curlyvee \bigwedge O_y(k), y(k)\right),$$

$$f_2(U(k), y(k), y(k+1)) = \tilde{f}\left(U(k) \curlywedge \bigvee O_y(k), y(k)\right)$$

solves Problem 1.

If we can find a Σ-weakly equivalent generalization $\Sigma_{\geqslant}$ for Σ that is weakly interval compatible with the desired lattice χ, we can construct the estimator for the system Σ by using $\Sigma_{\geqslant}$. This is formally stated in the following proposition.

Proposition 3. *If the system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ is observable (or weakly observable) and its Σ-weakly equivalent generalization $\Sigma_{\geqslant} = \mathscr{S}(\mathscr{U}_{\geqslant}, \mathscr{Z}, f_{\geqslant}, h)$ is such that the pair $(\tilde{\Sigma}_{\geqslant}, (\chi, \leqslant))$ is weakly interval compatible for a given $(\chi, \leqslant)$ and $\mathscr{U}_{\geqslant} \subseteq \chi$, then Theorem 6.2 can be applied to $\Sigma_{\geqslant}$ with $\alpha(k) = \sigma_{\Sigma}(k)(\alpha)$ and $z(k) = \sigma_{\Sigma}(k)(z)$.*

This way, we construct the estimator using $f_{\geqslant}$, but we estimate the value of α corresponding to the execution of Σ whose output z we are measuring. The proof of this proposition can be carried out easily by using directly (i) and (ii) of Definition 6.1. The counterpart is that if the Σ-weakly equivalent generalization is a too rough over-approximation of Σ, the convergence speed can be low.

A way for constructing a Σ-weakly equivalent generalization of Σ is to find a nondeterministic function $f_{\geqslant} : \mathscr{U} \times \mathscr{Z} \to \mathscr{P}(\mathscr{U})$ such that if $\alpha(k) = \sigma_{\Sigma}(k)(\alpha)$ and $z(k) = \sigma_{\Sigma}(k)(z)$, then $\alpha(k+1) \in f_{\geqslant}(\alpha(k), z(k))$. $f_{\geqslant}$ maps an element to a set of possible values in $\mathscr{U}$, and $\mathscr{U}_{\geqslant} = \mathscr{U}$. We show in the following section how the notion of interval compatible pair generalizes to nondeterministic systems, and how the result given in Theorem 4.1 modifies.

6.2. Nondeterministic transition systems

In this section, we outline the basic ideas that allow us to generalize the results of Section 4 to nondeterministic transition systems. The main difference with the deterministic case is that the function f maps one element to a set. In general, this feature may be due to model uncertainty, to noise on the dynamics, or to uncertainty on the sequence of events in a concurrent system. As a consequence, the extension $\tilde{f}$ cannot be a bijection. It maps the lower and upper bounds L and U to two sets. This way, it is not clear any more how to establish update rules of the kind of the ones in Theorem 4.1. In this section, we propose a solution to this problem.

Definition 6.3 (*Nondeterministic transition systems*). A *nondeterministic transition system* (NTS) is the tuple $\Sigma = (S, \mathscr{Y}, F, g)$, where (i) S is a set of states with $s \in S$; (ii) $\mathscr{Y}$ is a set of outputs with $y \in \mathscr{Y}$; (iii) $F : S \to \mathscr{P}(S)$ is the state transition set-valued function; (iv) $g : S \to \mathscr{Y}$ 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 $s(k+1) \in F(s(k))$ for all $k \in \mathbb{N}$. As opposed to a DTS, in an NTS F maps an element to a set, and thus it is a set-valued function. The Definitions 3.2, 3.5, and 3.6, which are related with the weak observability property, can be rewritten the same way for NTSs by replacing "deterministic transition system" with "nondeterministic transition system", and by taking that F is a set-valued map into account. As done for deterministic transition systems, we consider nondeterministic transition systems with the special structure (i) $S = \mathscr{U} \times \mathscr{Z}$ with $\mathscr{U}$ a finite set and $\mathscr{Z}$ a finite dimensional space; (ii) $F = (f, h)$, where $f : \mathscr{U} \times \mathscr{Z} \to \mathscr{P}(\mathscr{U})$ and $h : \mathscr{U} \times \mathscr{Z} \to \mathscr{Z}$; (iii) $g(\alpha, z) := z$, where $\alpha \in \mathscr{U}$, $z \in \mathscr{Z}$, and $\mathscr{Y} = \mathscr{Z}$. We denote this class of nondeterministic transition systems by $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$, and we associate to the tuple $(\mathscr{U}, \mathscr{Z}, f, h)$ the equations

$$\begin{aligned}\alpha(k+1) &\in f(\alpha(k), z(k)),\\ z(k+1) &= h(\alpha(k), z(k)),\\ y(k) &= z(k), \end{aligned} \tag{21}$$

if f is a set-valued map. Given a lattice $(\chi, \leqslant)$ with $\mathscr{U} \subset \chi$, the extension $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ of Σ is defined in a way similar to the way it is defined for deterministic transition systems (see Definition 4.2), but in this case $\tilde{\Sigma}$ is nondeterministic itself and $\mathscr{U}$ is allowed to be not invariant under $\tilde{f}$.

Definition 6.4. Given the nondeterministic transition system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$, a *N-extension of Σ on χ*, with $\mathscr{U} \subseteq \chi$ and $(\chi, \leqslant)$ a complete lattice, is any system $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$, such that (i) $\tilde{f} : \chi \times \mathscr{Z} \to \mathscr{P}(\chi)$ and $\tilde{f}|_{\mathscr{U}\times\mathscr{Z}} \cap \mathscr{P}(\mathscr{U}) = f$; (ii) $\tilde{h} : \chi \times \mathscr{Z} \to \mathscr{Z}$ and $\tilde{h}|_{\mathscr{U}\times\mathscr{Z}} = h$.

The definition of interval compatible pair changes to the following definition.

Definition 6.5. Consider the N-extension $\tilde{\Sigma} = \mathscr{S}(\chi, \mathscr{Z}, \tilde{f}, \tilde{h})$ of the nondeterministic transition system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ on $(\chi, \leqslant)$. The pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is said to be *N-interval compsatible* if

(i) Each $\tilde{\Sigma}$-transition class, $\mathscr{T}_i(\tilde{\Sigma}) \in \mathscr{T}(\tilde{\Sigma})$, is an interval sublattice of $(\chi, \leqslant)$, i.e., $\mathscr{T}_i(\tilde{\Sigma}) = [\bigwedge \mathscr{T}_i(\tilde{\Sigma}), \bigvee \mathscr{T}_i(\tilde{\Sigma})]$.
(ii) $\tilde{f} : ([L, U], z) \longrightarrow [\bigwedge \tilde{f}(L, z), \bigvee \tilde{f}(U, z)]$ is order preserving for any $[L, U] \subseteq \mathscr{T}_i(\tilde{\Sigma})$, and any $z \in \mathscr{Z}$ and for any $i \in \{1, \ldots, M\}$.
(iii) $\tilde{f} : ([L, U] \cap \mathscr{U}, z) \longrightarrow [\bigwedge \tilde{f}(L, z), \bigvee \tilde{f}(U, z))] \cap \mathscr{U}$ is onto for any $[L, U] \subseteq \mathscr{T}_i(\tilde{\Sigma})$ for any $z \in \mathscr{Z}$ and for any $i \in \{1, \ldots, M\}$.

Note that for a set-valued function f, we have that $f : A \to B$ is onto if for any element $b \in B$ there is an element $a \in A$ such that $b \in f(a)$. The notions of observability and of weak observability remain the same as the ones in the deterministic case. Thus, Theorem 4.1 transforms to the following.

Theorem 6.3. *Assume that the nondeterministic transition system $\Sigma = \mathscr{S}(\mathscr{U}, \mathscr{Z}, f, h)$ is weakly observable. If there is a lattice $(\chi, \leqslant)$, such that the pair $(\tilde{\Sigma}, (\chi, \leqslant))$ is N-interval compatible, then the deterministic transition system with input* $\hat{\Sigma} = (\chi \times \chi, \mathscr{Z} \times \mathscr{Z}, \chi \times \chi, (f_1, f_2), \mathrm{id})$ *with*

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

solves (i) *and* (iii) *of Problem* 1.

In Theorem 6.3, we assume that the system is weakly observable as opposed to observable as assumed in Theorem 4.1, and the functions f_1 and f_2 are modified by taking that $f(\cdot)$ is a set into account. Also, (ii) of Problem 1 cannot be guaranteed because $\tilde{f}$ maps an element to a set. The proof of this theorem proceeds the same way as the proof of Theorem 4.1. The basic idea of Theorem 6.3 is the same as the one of the deterministic counterpart, except that in the present case $L(k)$ and $U(k)$ are mapped to sets. As a consequence, the update laws in Theorem 6.3 take track of the lower and upper bounds of the sets to which $L(k)$ and $U(k)$ are respectively updated.

7. Conclusions and future work

In this paper, we have presented a novel approach to the estimation of discrete variables in systems where the continuous variables are available for measurement. Using lattice theory, we developed a discrete state estimator that updates two variables at each step, the upper and the lower bound of the set of all possible discrete states compatible with the output sequence. This way, we were able to overcome some of the severe complexity issues that arise in discrete state estimation methods based on the current observation tree such as is found in Caines et al. (1991), Balluchi et al. (2002), and Özveren and Willsky (1990), or in similar methods such as in Del Vecchio and Klavins (2003). In fact, these methods update the set of all possible discrete states compatible with the output sequence by updating each of the elements of the set; therefore, the computation need is prohibitive for systems in which the set of discrete states is large. We were able to overcome this problem by representing a set by its lower and upper bounds in some lattice, and by determining the updated set by the updates of its lower and upper bounds.

It was also shown that the proposed estimation approach is general as it applies to any observable system. The main advantage from a computational standpoint of using this approach is clear when a lattice with algebraic structure can be found, in which min and max can be computed exploiting the algebra. The existence of such a "good" lattice is often related to the way the system is described. For example, it can be shown that all discrete event systems described as Petri nets evolve on a partial order that is preserved by the dynamics of the system, and for any system the causal order relation is always preserved along its trajectories. These statements need to be formalized in order to give a clear idea of the set of systems in which the proposed approach successfully applies to reduce complexity. This will be investigated in our future work.

Many more aspects need to be still improved upon. More work needs to be done in order to formally identify the types of lattices that allow efficient computation and representation of joins and meets. In the case of a finite state machine, we would like to determine how to find the minimal lattice and what is the complexity of its computation. An other major challenge for our future work is to extend these results to the case in which also the continuous variables need to be estimated. In the case of the proposed multi-robot example, this would correspond to having more interesting evolutions of the robots continuous dynamics. The possibility of constructing a joint continuous-discrete variable lattice will be explored. Initial results in this direction can be found in Del Vecchio and Murray (2005). Finally, research needs to be done in order to establish how to close the loop for control on a lattice.

Acknowledgements

This work was supported in part by AFOSR under Grants F49620-01-1-0460 and FA9550-04-1-0169.

References

Alessandri, A., & Coletta, P. (2001). Design of Luenberger observer for a class of hybrid linear systems. In M. D. Di Benedetto, & A. Sangiovanni-Vincentelli (Eds.), *Lecture notes in computer science* (Vol. 2034) (pp. 7–18). Berlin: Springer.

Alessandri, A., & Coletta, P. (2003). Design of observer for switched discrete-time linear systems. In *American control conference* (pp. 2785–2790).

Balluchi, A., Benvenuti, L., Di Benedetto, M. D., & Sangiovanni-Vincentelli, A. (2002). Design of observers for hybrid systems. In C. J. Tomlin, & M. R. Greensreet (Eds.), *Lecture notes in computer science* (Vol. 2289) (pp. 76–89). Berlin: Springer.

Balluchi, A., Benvenuti, L., Di Benedetto, M. D., & Sangiovanni-Vincentelli, A. (2003). Observability of hybrid systems. In *Conference on Decision and Control* (pp. 1159–1164).

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

Caines, P. E., Greiner, R., & Wang, S. (1991). Classical and logic-based dynamic observers for finite automata. *IMA Journal of Mathematical Control and Information* (pp. 45–80).

Caines P. E., & Wang, S. (1995). Cocolog: A conditional observer and controller logic for finite machines. *SIAM Journal on Control and Optimization* (pp. 1687–1715).

Cassandra, A. R., Kaelbling, L. P., & Littman, M. L. (1994). Acting optimally in partially observable stochastic domains. In *Proceedings of the 12th conference on artificial intelligence* (pp. 1023–1028), Seattle, WA.

Costa, E. F., & do Val, J. B. R. (2002). On the observability and detectability of continuous-time jump linear systems. *SIAM Journal on Control and Optimization*, *41*, 1295–1314.

D'Andrea, R., Murray, R. M., Adams, J. A., Hayes, A. T., Campbell, M., & Chaudry, A. (2003). The RoboFlag Game. In *American control conference* (pp. 661–666).

Davey, B. A., & Priesteley, H. A. (2002). *Introduction to lattices and order*. Cambridge: Cambridge University Press.

Del Vecchio, D. (2005). *State estimation in multi-agent decision and control systems*. Ph.D. thesis, California Institute of Technology. <http://www.cds.caltech.edu/~ddomitilla/papers/Thesis.pdf>.

Del Vecchio, D., & Klavins, E. (2003). Observation of guarded command programs. In *Conference on decision and control* (pp. 3353–3359), Hawaii.

Del Vecchio, D., & Murray, R. M. (2004). Discrete state estimators for a class of hybrid systems on a lattice. In R. Alur, & G. Pappas (Eds.), *Lecture notes in computer science* (Vol. 2993) (pp. 311–325). Philadelphia. Berlin: Springer.

Del Vecchio, D., & Murray, R. M. (2005). Existence of cascade discrete-continuous state estimators on a partial order. In M. Morari, L. Thiele, & F. Rossi (Eds.), *Lecture notes in computer science* (Vol. 3414) (pp. 226–241). Zurich. Berlin: Springer.

De Santis, E., Di Benedetto, M. D., & Pola, G. (2003). On observability and detectability of continuous-time linear switching systems. In *Conference on decision and control* (pp. 5777–5782).

Kalman, R. E. (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. In *Conference on decision and control* (pp. 4133–4139), Hawaii.

Klavins, E., & Murray, R. M. (2004). Distributed algorithms for cooperative control. *Pervasive Computing*, *3*, 56–65.

Luenberger, D. G. (1971). An introduction to observers. *IEEE Transactions on Automatic Control*, *AC-16*(6), 596–602.

Oishi, M., Hwang, I., & Tomlin, C. (2003). Immediate observability of discrete event systems with application to user-interface design. In *Conference on decision and control* (pp. 2665–2672), Hawaii.

Özveren, C. M., & Willsky, A. S. (1990). Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control*, *35*(7), 797–806.

Ramadge, P. J. (1986). Observability of discrete event systems. In *Conference on decision and control* (pp. 1108–1112). Athens, Greece.

Sontag, E. D. (1998). *Mathematical control theory*. Berlin: Springer.

Vidal, R., Chiuso, A., & Soatto, S. (2002). Observability and identifiability of jump linear systems. In *Conference on decision and control* (pp. 3614–3619), Las Vegas.

Vidal, R., Chiuso, A., Soatto, S., & Sastry, S. (2003). Observability of linear hybrid systems. In O. Maler, & A. Pnueli (Eds.), *Lecture notes in computer science* (Vol. 2623) (pp. 526–539). Berlin: Springer.

Domitilla Del Vecchio received the Laurea degree in Electronic Engineering from the University of Rome at Tor Vergata in July 1999 and the Ph.D. degree in Control and Dynamical Systems from the California Institute of Technology in March 2005. She is currently an Assistant Professor of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. Her research interests include analysis, estimation, and control of hybrid systems, with application to multi-agent systems such as automotive and transportation systems, and mobile sensor networks. She is also interested in the application of control and dynamical systems theory to interdisciplinary research areas such as the analysis and design of gene regulatory networks.

Richard M. Murray received the B.S. degree in Electrical Engineering from California Institute of Technology in 1985 and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley, in 1988 and 1991, respectively. He is currently a Professor of Mechanical Engineering and the Chair of the Division of Engineering and Applied Science at the California Institute of Technology, Pasadena. In 1998–1999, Professor Murray was the Director of Mechatronic Systems at the United Technologies Research Center. His research interests include nonlinear control of mechanical systems with applications to aerospace vehicles and robotic locomotion, active control of fluids with applications to propulsion systems, and nonlinear dynamical systems theory.

Eric Klavins is an assistant professor of Electrical Engineering at the University of Washington in Seattle. He received a B.S. in Computer Science in 1996 from San Francisco State University and M.S. and Ph.D. degrees in Computer Science and Engineering in 1999 and 2001 from the University of Michigan in Ann Arbor, MI. From 2001 to 2003 he was a Postdoctoral Scholar in the Control and Dynamical Systems Department at the California Institute of Technology. As a graduate student, he was supported by a Charles DeVlieg Fellowship for manufacturing. In 2001, he received an NSF CAREER award entitled "Programmable Robotic Self-Assembly". His research interests include cooperative control, robotics, distributed systems, concurrency, self-organizing systems and nanotechnology.