Submitted, Automatica
http://www.cds.caltech.edu/~murray/papers/2004l_dmk04-automatica.html

Discrete State Estimators for Systems on a Lattice

D. Del Vecchio*, R. M. Murray*, and E. Klavins+
(*) Control and Dynamical Systems
California Institute of Technology
Pasadena
(+) Electrical Engineering
University of Washington
Seattle

Abstract. We address the problem of estimating discrete variables in a class of deterministic transition systems where the continuous variables are available for measurement. This simplified scenario has practical interest, for example, in the case of decentralized multi-robot systems. In these systems, the continuous variables represent physical quantities such as the position and velocity of a robot, while discrete variables may represent the state of the logical system that is used for control and coordination. We propose a novel approach to the estimation of discrete variables using basic lattice theory that overcomes some of the severe complexity issues encountered in previous work. We show how to construct the proposed estimator for a multi-robot system performing a cooperative assignment task.

Keywords: discrete state estimation, lattice, multi-agent systems.

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 et al. [5] 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. [5] propose the notion of incremental observability for piecewise affine systems and propose a deadbeat observer that requires large amounts of computation. Balluchi et al. [3] combine a *location* observer with a Luenberger observer to design hybrid observers that identify the location in a finite number of steps 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 et al., sufficient conditions for a linear hybrid system to be final state determinable are given [4]. In Alessandri et al., Luenberger-like observers are proposed for hybrid systems where the system location is known [1], [2]. Vidal et al. [21] 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 [22], these notions are generalized to the case of continuous time jump linear systems. For jump Markov linear systems, Costa and do Val derive test for observability [8], and Cassandra et al. propose an approach to optimal control for partially observable Markov decision processes [7]. For continuous time hybrid systems, De Santis et al. proposes a definition of observability based on the possibility of reconstructing the system state and testable conditions for observability are provided [18].

In the discrete event literature, observability has been defined by Ramadge [17], for example, which derives conditions for current state observability. Oishi et al. [16] derive conditions 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 et al. [11] and Caines [6] propose discrete event observers based on the construction of the current-location observation tree that, as explored also in Del Vecchio and Klavins [20], 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 allow 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 $(\chi, \leq)$, 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 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 [9]. 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 fill 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. 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 $\alpha : \{1, ..., N\} \rightarrow \{1, ..., N\}$, 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 [14] or [13] 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

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

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

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

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

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 [13].

Equation (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 [3]). The number of possible locations is $N!$, which, for $N \geq 8$, renders prohibitive the application of location observers based on the current-location observation tree as described in [6] and used in [3], [11], or discrete state observers based on similar concepts as the one in [20]. 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 Figure 1 (left) where $N = 8$. We see the blue robots $1, 3, 5$ going right and the others going left. From equations (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 $\alpha_i \geq i + 1$ for $i \in \{1, 2, 3\}$ and $\alpha_i \leq i$ for $i \in \{2, 4, 6, 7, 8\}$. The size of this set is of the order of 40320. According to the current-location observation tree method, this set needs to be mapped forward through the dynamics

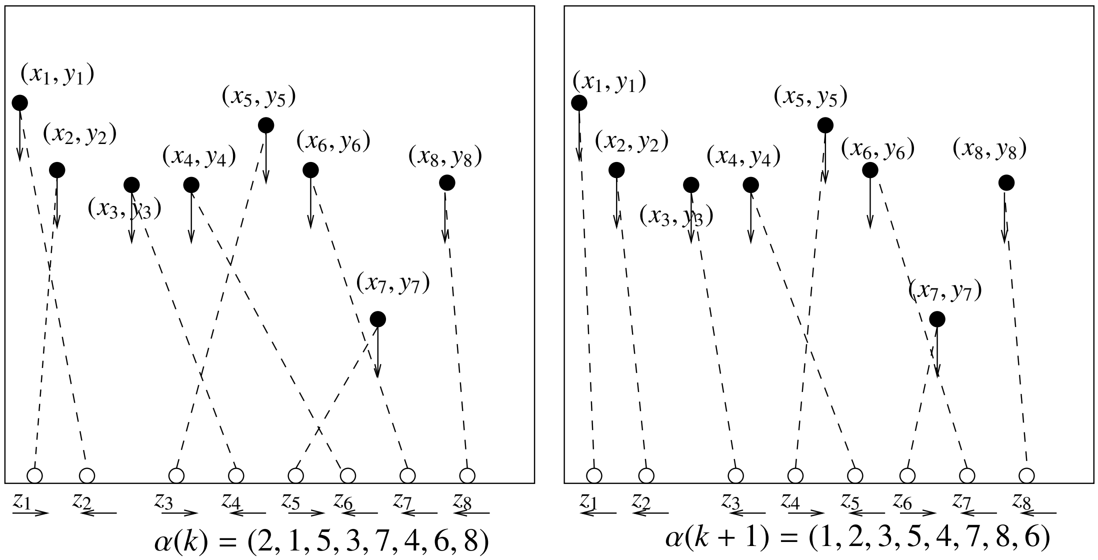

Figure 1: Example of the RoboFlag Drill with 8 robots per team.

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 order of 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 (2)-(3), denoted $O_y(k)$, can be represented as an interval with the order established component-wise, see the diagram in Figure 2. The function $\tilde{f}$ that maps such a set forward, specified by the equations (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 2, which can still be represented as an interval. When the new output measurement becomes available (Figure 1, right) we obtain the new set $O_y(k + 1)$ reported in Figure 2. The sets $\tilde{f}(O_y(k))$ and $O_y(k + 1)$ can be intersected by simply computing the maximum 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:

$$\begin{aligned} L(k+1) &= \tilde{f}(\max(L(k), \inf O_y(k))) \\ U(k+1) &= \tilde{f}(\min(U(k), \sup O_y(k))). \end{aligned} \tag{5}$$

As it will be shown in detail in the paper, the update laws in equations (5) have, among others, the property that $[L(k), U(k)] \cap \mathrm{perm}(N)$ tends to $\alpha(k)$. Letting $V(k) = |[L(k), U(k)] \cap \mathrm{perm}(N)|$,

$$\text{observation of } z \text{ motion at step } k \longrightarrow \underbrace{\left[\begin{pmatrix}2\\1\\4\\1\\6\\1\\1\\1\end{pmatrix},\begin{pmatrix}8\\2\\8\\4\\8\\6\\7\\8\end{pmatrix}\right]}_{O_y(k)} \xrightarrow[\tilde{f}]{} \underbrace{\left[\begin{pmatrix}1\\2\\1\\4\\1\\6\\1\\1\end{pmatrix},\begin{pmatrix}2\\8\\4\\8\\6\\8\\7\\8\end{pmatrix}\right]}_{\tilde{f}(O_y(k))} \cap \underbrace{\left[\begin{pmatrix}1\\1\\1\\5\\1\\7\\1\\1\end{pmatrix},\begin{pmatrix}1\\2\\3\\8\\5\\8\\7\\8\end{pmatrix}\right]}_{O_y(k+1)} = \left[\begin{pmatrix}1\\2\\1\\5\\1\\7\\1\\1\end{pmatrix},\begin{pmatrix}1\\2\\3\\8\\5\\8\\7\\8\end{pmatrix}\right]$$

Figure 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 *sublattice interval* 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 $k+1$.

Figure 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 $(1, ..., N)$.

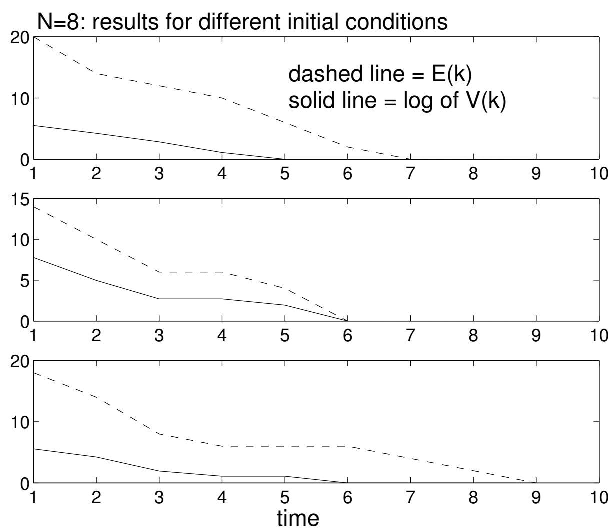

Figure 3: Convergence plots for the estimator (V(k)) compared to the convergence plot of the assignment protocol to its equilibrium (E(k)).

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 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 [10]. 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 "$\leq$", and we denote it by the pair $(\chi, \leq)$. We define the *join* "$\vee$" and the *meet* "$\wedge$" of two elements x and w in χ as

1. $x \vee w = \sup\{x, w\}$ and $x \wedge w = \inf\{x, w\}$;
2. if $S \subseteq \chi$, $\bigvee S = \sup\ S$ and $S \subseteq \chi$, $\bigwedge S = \inf\ S$;

where by $\sup\{x, w\}$ we mean the smallest element in χ that is bigger than both x and w, and we denote by $\inf\{x, w\}$ the biggest element in χ that is smaller than both x and w.

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 5 (left) we illustrate Hasse diagrams [10] 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.

Let $(\chi, \leq)$ be a partial order. Then $(\chi, \leq)$ 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. At the opposite extreme of a chain is an anti-chain. The partial order $(\chi, \leq)$ is an *anti-chain* if $x \leq y$ if and only if $x = y$.

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*. 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 only two elements. For example, the intervals of $(\mathbb{R}, \leq)$ are just the familiar closed intervals on the real line.

Let $(\chi, \leq)$ 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 $(\chi, \leq)$ 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 $\mathcal{A}(\mathcal{P}(\mathcal{U}), \subseteq) = \mathcal{U}$. An example is illustrated in Figure 4. Given a set P, we denote by $|P|$ its cardinality.

Definition 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 $x \leq w \implies f(x) \leq f(w)$;

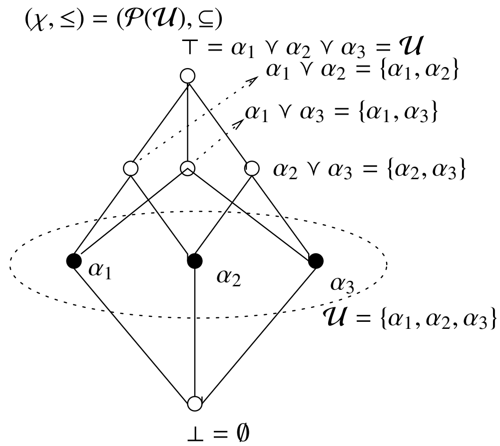

Figure 4: Power lattice $(\chi, \leq)$ of a set $\mathcal{U}$ composed by three elements.

(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.

Definition 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 3.1. *(see [10]) If $f : P \to Q$ is a bijective homomorphism, then it is an order isomorphism.*

Every order isomorphic map faithfully mirrors the structure of P onto Q. In Figure 5 (right) we show some examples. The notion of order preserving map can be generalized to the case in which the map is non deterministic, 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 non-standard definition.

Definition 3.3. Let $x, w \in \chi$, with $(\chi, \leq)$ a lattice, $x \leq w$, and $f : \chi \to \mathcal{P}(\chi)$. We say that f is *order preserving* if $\bigvee f(x) \leq \bigvee f(w)$ and $\bigwedge f(x) \leq \bigwedge f(w)$.

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

(i) S is a set of states with $s \in S$;

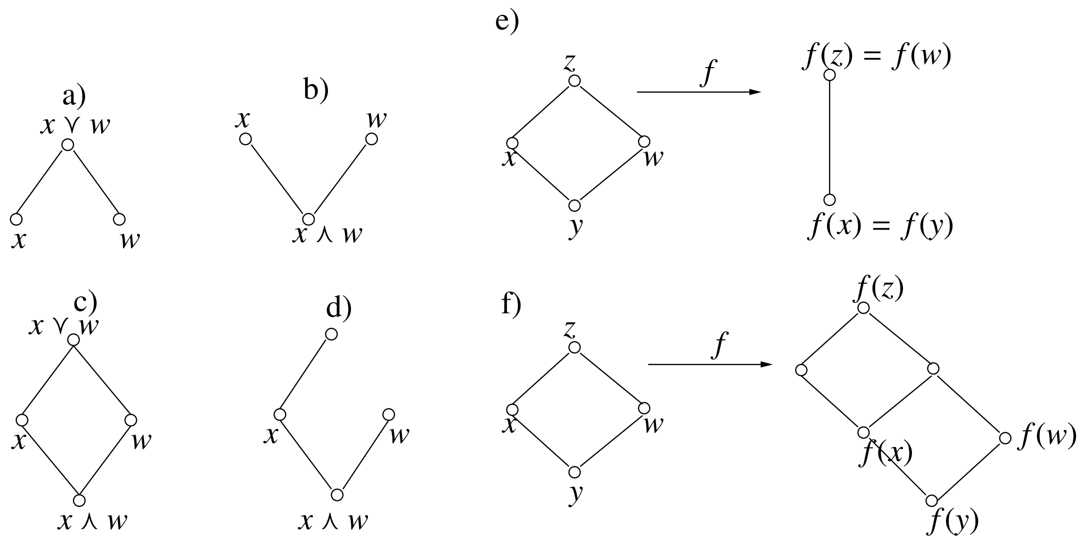

Figure 5: (Left) In diagram a) and b), x and w are not related, but they have a join and a meet respectively. In diagram c), we show a complete lattice. In diagram d), we show a partially ordered set that is not a lattice, since the elements x and w have a meet, but not a join. In diagram e), we show a map that is order preserving but not order embedding. In diagram f), we show an order embedding that is not an order isomorphism: 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).

(ii) $\mathcal{Y}$ is a set of outputs with $y \in \mathcal{Y}$;

(iii) $F : S \to S$ is the state transition function;

(iv) $g : S \to \mathcal{Y}$ 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 $\mathcal{E}(\Sigma)$.

Definition 3.5. Let $\Sigma = (S, \mathcal{Y}, F, g)$ 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 $\sigma \in \mathcal{E}(\Sigma)$, there exists k_σ such that $\sigma(k) \in \Omega$ for all $k \geq k_\sigma$.

Definition 3.6. Given a deterministic transition system $\Sigma = (S, \mathcal{Y}, F, g)$, two executions $\sigma_1, \sigma_2 \in \mathcal{E}(\Sigma)$ are *distinguishable* if there exists a k such that $g(\sigma_1(k)) \neq g(\sigma_2(k))$.

Definition 3.7. (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.

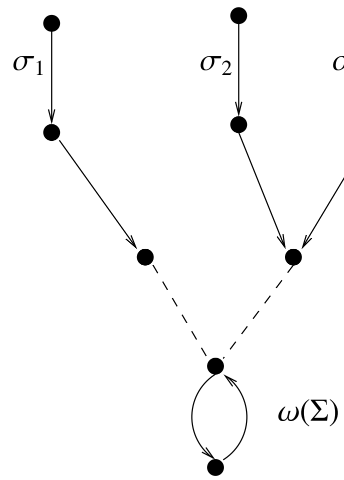

Figure 6: Executions σ_2 and σ_3 are weakly equivalent according to Definition 3.8 while σ_3 is not weakly equivalent to either σ_1 or σ_2.

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* [19] that accounts for this distinction.

Definition 3.8. Given a deterministic transition system $\Sigma = (S, \mathcal{Y}, F, g)$, two executions $\sigma_1, \sigma_2 \in \mathcal{E}(\Sigma)$ 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 $k \geq k^*$.

In Figure 6, we show examples of equivalent and not equivalent system executions.

Definition 3.9. (Weak Observability) A deterministic transition system $\Sigma = (S, \mathcal{Y}, F, g)$ 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 $\Sigma = (S, \mathcal{Y}, F, g)$ 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) $g(\alpha, z) := z$, where $\alpha \in \mathcal{U}$, $z \in \mathcal{Z}$, 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 2, $\mathcal{U} = \mathrm{perm}(N)$ and $\mathcal{Z} = \mathbb{R}^N$, the function f is represented by equations (4) and the function h is represented by equations (2-3). In the sequel, we will denote this class of DTS by $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ where we associate to the tuple $(\mathcal{U}, \mathcal{Z}, f, h)$, the equations:

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

where $\alpha \in \mathcal{U}$ and $z \in \mathcal{Z}$. An execution of the system Σ in equations (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 $(S, \mathcal{I}, \mathcal{Y}, F, g)$ in which

(i) S is a set of states;

(ii) $\mathcal{I}$ is a set of inputs;

(iii) $\mathcal{Y}$ is a set of outputs;

(iv) $F : S \times \mathcal{I} \to S$ is a transition function;

(v) $g : S \to \mathcal{Y}$ 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 $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$. First, note that the set $\mathcal{U}$ 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 [20], 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 $\mathcal{U}$ can be immersed in a larger set χ whose elements can be related by an order relation $\leq$, we could represent a subset of $(\chi, \leq)$ 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 $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$, find a deterministic transition system with input $\hat{\Sigma} = (\chi \times \chi, \mathcal{Z} \times \mathcal{Z}, \chi \times \chi, (f_1, f_2), \text{id})$, with $f_1 : \chi \times \mathcal{Z} \times \mathcal{Z} \to \chi$, $f_2 : \chi \times \mathcal{Z} \times \mathcal{Z} \to \chi$, $\mathcal{U} \subseteq \chi$, with $(\chi, \leq)$ a lattice, represented by the equations

$$\begin{array}{rcl} L(k+1) & = & f_1(L(k), y(k), y(k+1)) \\ U(k+1) & = & f_2(U(k), y(k), y(k+1)) \end{array}$$

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).

In the example shown in Section 2, we had that

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

and

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

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 $\mathcal{U} = \text{perm}(N)$, and χ the set of vectors in $\mathbb{N}^N$ with components $x_i \in [1, N]$. The order is established componentwise, so that $(\chi, \leq)$ is a complete lattice. The function $\tilde{f}$ is defined on $(\chi, \leq)$, it coincides with f on $\mathcal{U}$, and it preserves the structure of the interval sublattices in $(\chi, \leq)$. With $\tilde{f}$, we extend the system defined on $\mathcal{U}$ 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 $(\chi, \leq)$ 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 $(\chi, \leq)$ such that $\mathcal{U} \subseteq \chi$ for some 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. 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 $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$, an *extension of* Σ *on* χ, with $\mathcal{U} \subseteq \chi$ and $(\chi, \leq)$ a complete lattice, is any system $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$, such that

(i) $\tilde{f} : \chi \times \mathcal{Z} \to \chi$ and $\tilde{f}|_{\mathcal{U}\times\mathcal{Z}} = f$;

(ii) $\tilde{h} : \chi \times \mathcal{Z} \to \mathcal{Z}$ and $\tilde{h}|_{\mathcal{U}\times\mathcal{Z}} = h$.

Definition 4.3. (Transition sets) Let $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ 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 $\tilde{h}$.

Definition 4.4. *(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 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 4.5. (Output set) Given the extension $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ of the deterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ 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 $i \in \{1, ..., M\}$. 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 ($\tilde{h}|_{\mathcal{U}\times\mathcal{Z}} = h$), 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 $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ of the system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ 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 $(\chi, \leq)$:

$$\mathcal{T}_i(\tilde{\Sigma}) = [\bigwedge \mathcal{T}_i(\tilde{\Sigma}), \bigvee \mathcal{T}_i(\tilde{\Sigma})]$$

(ii) $\tilde{f} : (\mathcal{T}_i(\tilde{\Sigma}), z) \rightarrow [\tilde{f}(\bigwedge \mathcal{T}_i(\tilde{\Sigma}), z), \tilde{f}(\bigvee \mathcal{T}_i(\tilde{\Sigma}), z)]$ 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 for Problem 1.

Theorem 4.1. *Assume that the deterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ 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*

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

solves Problem 1.

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

$$\begin{aligned} L(k+1) &= \tilde{f}(L(k) \vee \bigwedge O_y(k), y(k)) \\ U(k+1) &= \tilde{f}(U(k) \wedge \bigvee O_y(k), y(k)) \end{aligned} \tag{7}$$

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

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

Because we have that $L(k) \vee \bigwedge O_y(k) \in O_y(k)$, and $U(k) \wedge \bigvee O_y(k) \in O_y(k)$, and the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible, we can use the isomorphic property of $\tilde{f}$ (property (ii) of Definition 4.6), 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 (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 $w = \tilde{f}(z)$, and hence the number of elements in $[L(k+1), U(k+1)]$ is almost the same as the number of elements in $[L(k), U(k)]$. By equation (7), $w \in [L(k+1), U(k+1)]$ implies that

$$\tilde{f}(L(k) \vee \bigwedge O_y(k)) \leq w \leq \tilde{f}(U(k) \wedge \bigvee O_y(k)). \tag{8}$$

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 4.6), 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 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

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

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 $k \geq k_0$ 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. □

The following corollary is a consequence of 4.1 in the case in which the extended system $\tilde{\Sigma}$ is observable.

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 $L(k) \longrightarrow \alpha(k)$ and $U(k) \longrightarrow \alpha(k)$ as $k \longrightarrow \infty$.*

Proof. The proof proceeds by contradiction. Assume that for any $k_0 \geq 0$ there is $k \geq k_0$ 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 $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ that share the same output sequence. This contradicts the observability of the system $\tilde{\Sigma}$. □

5 Example: The RoboFlag Drill

The RoboFlag Drill has been described in Section 2. In this section, we revisit the example by showing first that it is observable with measurable variables z, and then by finding a lattice and a system extension that can be used for constructing the estimator proposed in Theorem 4.1.

5.1 System specification

For completeness, we report here the system specification. The red robot dynamics are described by the N rules

$$y_i(k+1) = y_i(k) - \delta \quad \text{if} \quad y_i(k) \geq \delta \tag{9}$$

for $i \in \{1, ..., N\}$. These state simply that the red robots move a distance δ toward the defensive zone at each step. The blue robot dynamics are described by the $2N$ rules

$$\begin{aligned} z_i(k+1) &= z_i(k) + \delta \quad \text{if} \quad z_i(k) < x_{\alpha_i(k)} \\ z_i(k+1) &= z_i(k) - \delta \quad \text{if} \quad z_i(k) > x_{\alpha_i(k)} \end{aligned} \tag{10}$$

for $i \in \{1, ..., N\}$. For the blue robots we assume that initially $z_i \in [z_{min}, z_{max}]$ and $z_i < z_{i+1}$ and that $x_i < z_i < x_{i+1}$ for all time. The assignment protocol dynamics is defined by

$$(\alpha_i(k+1), \alpha_{i+1}(k+1)) = (\alpha_{i+1}(k), \alpha_i(k)) \quad \text{if} \quad x_{\alpha_i(k)} \geq z_{i+1}(k) \wedge x_{\alpha_{i+1}(k)} \leq z_{i+1}(k), \tag{11}$$

which is a modification of the protocol presented in [13], since two adjacent robots switch assignments only if they are moving one toward the other. We define $x = (x_1, ..., x_N)$, $z =$

$(z_1, ..., z_N)$, $\alpha = (\alpha_1, ..., \alpha_N)$. The complete RoboFlag specification is then given by the program given in rules (9)-(10)-(11). In particular the rules in (10) model the function $h : \mathcal{U} \times \mathcal{Z} \to \mathcal{Z}$ that updates the continuous variables, and the rules in (11) 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 $\Sigma = \mathcal{S}(\text{perm}(N), \mathbb{R}^N, f, h)$, and the variables $z \in \mathbb{R}^N$ are measured.

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

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

Proposition 5.1. *The system $\Sigma = \mathcal{S}(perm(N), \mathbb{R}^N, f, h)$ represented by the rules (10) and (11) 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, also their direction of motion is 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 $\sigma_1(\bar{k})(\alpha) \neq \sigma_2(\bar{k})(\alpha)$. Here we have denoted by $\{\sigma(k)(\alpha)\}_{k\in\mathbb{N}}$ the α sequence corresponding to the execution σ as introduced in Section 4. As a consequence the system is observable if $g(\sigma_1(\bar{k})) \neq g(\sigma_2(\bar{k}))$. 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 $z, v \in \mathbb{R}^N$. $g(\alpha) = g(\beta)$ by (10) implies that (1) $z_i < x_{\alpha_i} \iff v_i < x_{\beta_i}$ and (2) $z_i \geq x_{\alpha_i} \Leftrightarrow v_i \geq x_{\beta_i}$. 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 $f(\alpha, z) \neq f(\beta, v)$. □

5.2 RoboFlag Drill Estimator

In the previous section, we have shown that the RoboFlag system $\Sigma = \mathcal{S}(\text{perm}(N), \mathbb{R}^N, f, h)$ represented by the rules (10) and (11) with measurable variables z is observable. 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 need to 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 first construct a lattice $(\chi, \leq)$ and the extended system $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ such that $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible. We choose as χ the set of vectors in $\mathbb{N}^N$ with coordinates $x_i \in [1, N]$, that is

$$\chi = \{x \in \mathbb{N}^N : x_i \in [1, N]\} . \tag{12}$$

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

$$\forall x, w \in \chi,\ x \leq w \text{ if } x_i \leq w_i\ \forall i\ . \tag{13}$$

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 (13) 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 in $(\chi, \leq)$. In the sequel, we will denote by w the variables in χ not specifying if it is in $\mathcal{U}$, and we will 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

$$\begin{aligned} z_i(k+1) &= z_i(k) + \delta \quad \text{if} \quad z_i(k) < x_{w_i(k)} \\ z_i(k+1) &= z_i(k) - \delta \quad \text{if} \quad z_i(k) > x_{w_i(k)} \end{aligned} \tag{14}$$

for $w \in \chi$. The rules (14) 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 analogous way $f : \text{perm}(N) \times \mathbb{R}^N \to \text{perm}(N)$ is extended to χ as

$$(w_i(k+1), w_{i+1}(k+1)) = (w_{i+1}(k), w_i(k)) \quad \text{if} \quad x_{w_i(k)} \geq z_{i+1}(k) \wedge x_{w_{i+1}(k)} \leq z_{i+1}(k), \tag{15}$$

for $w \in \chi$. The rules (15) 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 $\tilde{\Sigma} = (\tilde{f}, \tilde{h}, \chi, \mathbb{R}^N)$ is the extended system of $\Sigma = (f, h, \text{perm}(N), \mathbb{R}^N)$ (see Definition 4.2).

The following proposition shows that the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible.

Proposition 5.2. *The pair $(\tilde{\Sigma}, (\chi, \leq))$, where $\Sigma = \mathcal{S}(perm(N), \mathbb{R}^N, f, h)$ is represented by the rules (10-11), and $(\chi, \leq)$ is given by (12-13), is interval compatible.*

Proof. According to Definition 4.6, we need to show the following two properties

(i) $\mathcal{T}_i(\tilde{\Sigma}) = [\bigwedge \mathcal{T}_i(\tilde{\Sigma}), \bigvee \mathcal{T}_i(\tilde{\Sigma})]$,

(ii) $\tilde{f} : ([\bigwedge \mathcal{T}_i(\tilde{\Sigma}), \bigvee \mathcal{T}_i(\tilde{\Sigma})]) \to [\tilde{f}(\bigwedge \mathcal{T}_i(\tilde{\Sigma})), \tilde{f}(\bigvee \mathcal{T}_i(\tilde{\Sigma}))]$ is an order isomorphism.

To simplify notation, we neglected the dependence of $\tilde{f}$ on its second argument.

Proof of (i): By (14) 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

$$T_{(z^1,z^2)}(\tilde{\Sigma}) = \begin{cases} \{w \mid x_{w_i} > z_i^1, \}, & \text{if } z_i^2 = z_i^1 + \delta \\ \{w \mid x_{w_i} < z_i^1, \}, & \text{if } z_i^2 = z_i^1 - \delta \\ \{w \mid x_{w_i} = z_i^1, \}, & \text{if } z_i^2 = z_i^1. \end{cases} \tag{16}$$

Because we assumed that $x_i < z_i < x_{i+1}$, we have that

$$x_{w_i} > z_i \quad \text{if and only if} \quad w_i > i$$
$$x_{w_i} < z_i \quad \text{if and only if} \quad w_i < i.$$

This, along with relations (16) and Definition 4.4, imply (i).

Proof of (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; 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 (15) 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 to $\mathcal{T}_{i,j}$. 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 $\hat{\Sigma} = (\chi \times \chi, \mathcal{Z} \times \mathcal{Z}, \chi \times \chi, (f_1, f_2), \mathrm{id})$ given in Theorem 4.1 can be constructed because the hypotheses of the theorem are satisfied by virtue of Proposition 5.1 and Proposition 5.2. The estimator $\hat{\Sigma}$ can be specified by the following rules

$$l_i(k+1) = i+1 \quad \text{if} \quad z_i(k+1) = z_i(k) + \delta \tag{17}$$
$$l_i(k+1) = 1 \quad \text{if} \quad z_i(k+1) = z_i(k) - \delta \tag{18}$$

$$L_{i,y}(k+1) = \max\{L_i(k), l_i(k+1)\} \tag{19}$$

$$\begin{aligned}(L_i(k+1), L_{i+1}(k+1)) &= (L_{i+1,y}(k+1), L_{i,y}(k+1)) \\ &\text{if } x_{L_{i,y}(k+1)} \geq z_{i+1}(k) \wedge x_{L_{i+1,y}(k+1)} \leq z_{i+1}(k)\end{aligned} \tag{20}$$

$$u_i(k+1) = N \quad \text{if} \quad z_i(k+1) = z_i(k) + \delta \tag{21}$$
$$u_i(k+1) = i \quad \text{if} \quad z_i(k+1) = z_i(k) - \delta \tag{22}$$

$$U_{i,y}(k+1) = \min\{U_i(k), U_i(k+1)\} \tag{23}$$

$$\begin{aligned}(U_i(k+1), U_{i+1}(k+1)) &= (U_{i+1,y}(k+1), U_{i,y}(k+1)) \\ &\text{if } x_{U_{i,y}(k+1)} \geq z_{i+1}(k) \wedge x_{U_{i+1,y}(k+1)} \leq z_{i+1}(k)\end{aligned} \tag{24}$$

initialized with $L(0) = \bigwedge\chi$ and $U(0) = \bigvee\chi$. Rules (17-18) and (21-22) take the output information z and set the lower and upper bound of $O_y(k)$ respectively. Rules (19) and (23) compute the lower and upper bound of the intersection $[L(k), U(k)] \cap O_y(k)$ respectively. Finally, rules (20) and (24) compute the lower and upper bound of the set $\tilde{f}([L(k), U(k)] \cap O_y(k))$ respectively.

5.3 Complexity of the RoboFlag Drill Estimator

The amount of computation required for updating L and U according to (17-24)) 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 we can roughly say that the complexity of the algorithm that generates the sequences $L(k)$ and $U(k)$ is about the same as the complexity of the algorithm that generates the α trajectories. Also note that the rules in (17-24) are obtained by "copying" the rules in (15) 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 [12], [15]).

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 algorithm proposed in equation (17-24). 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, 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.

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 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 $[L, U] \cap \mathcal{U} = \alpha$. Second, the cardinality of the sets $m_i(k)$ is non-increasing with the time step k. These properties are proved formally in the following propositions.

Proposition 5.3. *If* $[L, U] \cap \mathcal{U} = \alpha$ *with* $L, U \in \chi$, *and* $c_i = [L_i, U_i]$, *then* $Refine(c_1, ..., c_N) = \alpha$.

Proof. Let c_i denote the sets $[L_i, U_i]$. Also, let $\mathcal{U}_i$ denote the set of permutation of i elements. If $[L, U] \cap \mathcal{U} = \alpha$, we note that among the sets $[L_i, U_i]$ there is at least one i for which $L_i = U_i$, and therefore we have at least one singleton to take out from all of the other coordinate sets. Without loss in generality we assume that $i = N$ (if not we can reduce to this case by performing a permutation of the coordinate sets and keeping track of the used permutation). We are left to show that the process of taking out one singleton always creates a new singleton that then needs to be removed from the other coordinate sets. Then, we remove that singleton from all of the other sets c_j for $j < N$ to obtain new sets c_j^1 whose elements take values in a set of possible $N-1$ natural numbers. Still, there is only one $\beta \in \mathcal{U}_{N-1}$ such that $\beta \in (c_1^1, ..., c_{N-1}^1)$. Again, for this to be true there must exist j such that c_j^1, for $j \in [1, N-1]$, is a singleton. Assume $j = N-1$. We thus remove this singleton from all of the other sets c_j^1 for $j < N-1$ to obtain new sets c_j^2 whose elements take values in a set of possible $N-2$ natural numbers. Proceeding iteratively, we finally obtain $m_1 = c_1^{N-1}$,...,$m_{N-1} = c_{N-1}^1$, $m_N = c_N$, which implies that the m_i are singletons. Since $\alpha_i \in m_i$ by construction, we have proved what we wanted. □

Proposition 5.4. *Let $c_i(k) = [L_i(k), U_i(k)]$, and denote by $m_i(k)$ the sets obtained with the refinement algorithm. Then*

$$\sum_{i=1}^{N} |m_i(k+1)| \leq \sum_{i=1}^{N} |m_i(k)|$$

Proof. Let us denote the variables at step $k+1$ with primed variables and the variables at step k with unprimed variables. The proof proceeds by showing that for each j there exist a k such that $m'_j \subseteq m_k$. By equations(17-24) we deduce that we can have one of the following cases for each i: (a) $c'_i \subseteq c_{i+1} \ \wedge \ c'_{i+1} \subseteq c_i$, (b) $c'_i \subseteq c_i$, (c) $c'_i \subseteq c_{i-1} \ \wedge \ c'_{i-1} \subseteq c_i$. Let us consider case (a), the other cases can be treated in analogous way. Let c_j be a singleton. In the refinement process it is deleted from any other set, so that we have $c_i = m_i + c_j$ for all i. Assume that in the first singleton removal process no new singletons are created. We have one of the following situations: $c'_j \subseteq c_{j+1} \ \wedge \ c_{j+1} \subseteq c_j$, $c'_j \subseteq c_j$, $c'_j \subseteq c_{j-1} \ \wedge \ c'_{j-1} \subseteq c_j$. This implies that one of the c'_k is equal to the singleton c_j. The sets m'_i are created removing such singleton for all the other sets, so that we obtain $m'_i + c_j = c'_i \subseteq c_{i+1} = m_{i+1} + c_j$ and $m'_{i+1} + c_j = c'_{i+1} \subseteq c_i = m_i + c_j$. This in turn implies that $m'_i \subseteq m_{i+1}$ and $m'_{i+1} \subseteq m_i$. Because this holds for any i, we have that $\sum_{i=1}^{N} |m'_i| \leq \sum_{i=1}^{N} |m_i|$. This reasoning can be generalized to the case where a singleton removal process creates new singletons. □

5.4 Simulation Results

The RoboFlag Drill system represented in the rules (10) and (11) has been implemented in MATLAB together with the estimator reported in the rules (17-24). Figure 7 (left) shows the behavior of the quantity

$$V(k) = |[L(k), U(k)] \cap \mathcal{U}|$$

and

$$E(k) = \frac{1}{N} \sum_{i=1}^{N} |\alpha_i(k) - i|.$$

$V(k)$ 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 $\Sigma = \mathcal{S}(\text{perm}(N), \mathbb{R}^N, f, h)$. This quantity is showing the rate of convergence of the α assignment to its equilibrium $(1, ..., N)$. In Figure 7 (right) 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

$$W(k) = \frac{1}{N} \sum_{i=1}^{N} |m_i(k)|,$$

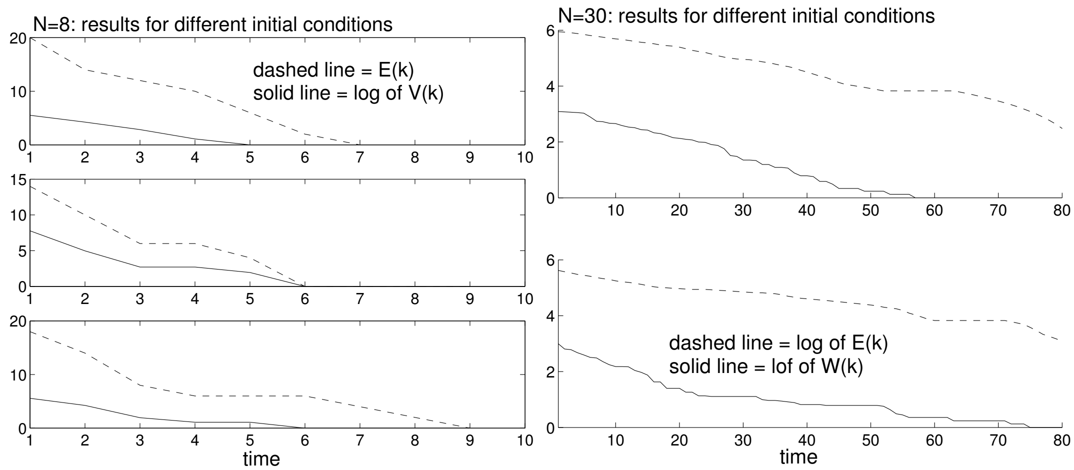

Figure 7: (Left) Example with N=8: note that the function $V(k)$ is always non-increasing because the set $\chi - \mathcal{U}$ is invariant under $\tilde{f}$. (Right) Example with N=30: note that the function $W(k)$ is always non-increasing and its logarithm is converging to zero.

which by virtue of Proposition 5.3 and Proposition 5.4 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 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 $\hat{\Sigma} = (\chi\times\chi, \mathcal{Z}\times\mathcal{Z}, \chi\times\chi, (f_1, f_2), \text{id})$ on a lattice $(\chi, \leq)$ for a DTS $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ with $\mathcal{U} \subseteq \chi$. Such an estimator can be constructed if the system Σ is observable and if the extended system $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ is such that the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible. In the next section, we investigate when the pair $(\tilde{\Sigma}, (\chi, \leq))$ 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 $(\chi, \leq)$. We show that if the system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ 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 possible cause of complexity, and a worst case size is computed. For systems where $\mathcal{U}$ can be naturally 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. However, the pair $(\tilde{\Sigma}, (\chi, \leq))$ is not necessarily interval compatible for any $(\chi, \leq)$. We propose a way of constructing the estimator on a chosen lattice by constructing a nondeterministic extension of Σ on χ. The previous section results are thus generalized for non-deterministic systems.

6.1 Estimator Existence

For the deterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$, the Σ-transition sets and the Σ-transition classes are defined as for the extended system $\tilde{\Sigma} = \mathcal{S}(\mathcal{U}, \mathcal{Z}, \tilde{f}, \tilde{h})$ in Definition 4.3 and Definition 4.4 respectively, by replacing $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ with $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$. 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 $\mathcal{T}_i(\Sigma) \subseteq \mathcal{T}_i(\tilde{\Sigma})$.

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 $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$, $\omega(\Sigma)$, 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 $\sigma_1(k)(z) \neq \sigma_2(k)(z)$.

In the case where ω^+-limit set is just one fixed point, this assumption is always trivially verified. In case where ω^+-limit set is made up by fixed points and limit cycles, the assumption requires that any two different fixed points have different output values, otherwise two different executions starting in the two fixed points will not be distinguishable.

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

This lemma shows that observability can be determined by checking if the function f is one to one on the Σ-transition classes $\mathcal{T}_j(\Sigma)$, 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 $(\chi, \leq)$.

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

(i) System Σ is observable;

(ii) There exist a complete lattice $(\chi, \leq)$ 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. ((i) $\Rightarrow$ (ii))We show the existence of a lattice $(\chi, \leq)$ and of an extended system $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ 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 $\tilde{h}$, we 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 8. 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 $\mathcal{T}_i(\tilde{\Sigma}) = [\perp, \bigvee\mathcal{T}_i(\tilde{\Sigma})]$.

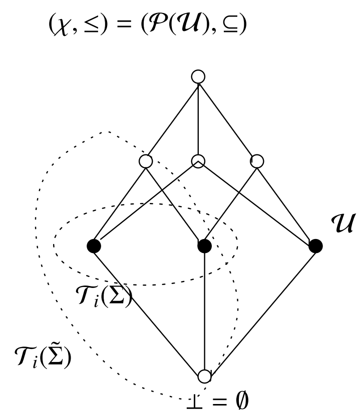

Figure 8: Example of the Σ and $\tilde{\Sigma}$ transition classes with $\mathcal{U}$ (dark elements) composed by three elements.

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

$$\begin{cases} \tilde{f}(x \vee w) & = \tilde{f}(x) \vee \tilde{f}(w) \\ \tilde{f}(x \wedge w) & = \tilde{f}(x) \wedge \tilde{f}(w) \\ \tilde{f}(\perp) & = \perp \\ \tilde{f}(\alpha) & = f(\alpha), \end{cases} \tag{25}$$

where we have omitted the dependency on the z variables for simplifying notation. We prove first that $\tilde{f} : \mathcal{T}_i(\tilde{\Sigma}) \rightarrow [\perp, \tilde{f}(\bigvee \mathcal{T}_i(\tilde{\Sigma}))]$ is onto. We have to show that for any $w \neq \perp \in [\perp, \tilde{f}(\bigvee \mathcal{T}_i(\tilde{\Sigma}))]$ there is $x \in [\perp, \bigvee \mathcal{T}_i(\tilde{\Sigma})]$ such hat $w = \tilde{f}(x)$. 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 (25). 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 (25), 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 $\tilde{f} : \mathcal{T}_i(\tilde{\Sigma}) \rightarrow [\perp, \tilde{f}(\bigvee \mathcal{T}_i(\tilde{\Sigma}))]$ is one to one because of Lemma 6.1. Thus, we have proved that $\tilde{f} : \mathcal{T}_i(\tilde{\Sigma}) \rightarrow [\perp, \tilde{f}(\bigvee \mathcal{T}_i(\tilde{\Sigma}))]$ is a bijection, and by equations (25) it is also an homomorphism. We then apply Proposition 3.1 to obtain the result.

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

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

6.2 Complexity Considerations

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 $\mathcal{U}$ is large because the size of the representation of the elements of χ is large as well. In such a case, one needs to find a better lattice, if it exists, considering its size, the representation of its elements, and the complexity of computing joins and meets. In the RoboFlag Drill, for example, such a better lattice exists. Even if the size of $\mathcal{U}$ is $N!$ (which is huge if N is large) the lattice $(\chi, \leq)$ is such that its elements can be represented by a set of N natural numbers plus joins and meets, and $\tilde{f}$ can be computed by using the algebra naturally associated with $\mathbb{N}^N$. Thus, some systems have a preferred lattice structure that drastically reduces complexity. For these systems however, the extended system and such a preferred lattice structure are not always interval compatible. In such a case, we propose a way in Proposition 6.4 to construct an estimator on the desired lattice even if the interval compatibility condition is not satisfied. The counter part is that the convergence speed of the estimator can be lower.

Instead, in the case a system does not have a preferred lattice structure, a lattice $(\chi, \leq)$ for which there is an extension $\tilde{\Sigma}$ such that $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible is the power lattice of $\mathcal{U}$ proposed in the proof of Theorem 6.1 or a lattice isomorphic to it. Thus, the size of χ is the primary source of complexity. 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, that is, the elements in the $\tilde{\Sigma}$-transition classes. With this consideration, the following proposition computes the worst case size of χ.

Definition 6.1. Consider the deterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$. We define the Σ-*overlap factor* by

$$m_i := \max_{j \in \{1,\ldots,M\},\, z \in \mathbb{R}^N} \Big(|f(\mathcal{T}_i(\Sigma), z) \cap \mathcal{T}_j(\Sigma)|, |f(\mathcal{T}_j(\Sigma), z) \cap \mathcal{T}_i(\Sigma)| \Big). \tag{26}$$

Basically, the Σ-overlap factor gives an idea of how many values of α are such that α is in $O_y(k)$ and $f(\alpha, y(k))$ is in $O_y(k+1)$ for any k. Ideally, we would like this number to be the smallest as possible, so that with few output measurement we would single out the value of α.

Proposition 6.2. *Consider the deterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$. There exists a lattice $(\chi, \leq)$ such that $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible, with*

$$|\chi| \leq \sum_{i=1}^{M} \left(\sum_{j=1}^{m_i} \binom{K_i}{j} + 1 \right) + 2 \tag{27}$$

where $K_i = |\mathcal{T}_j(\Sigma)|$ for $i \in \{1, ..., M\}$.

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 6.2 is 16370, and the size of the lattice χ proposed in Section 5.2 is $4^4 = 256$. Thus the estimate given by Proposition 6.2 significantly reduces the size of χ given by $\mathcal{P}(\mathcal{U})$. Note that the size of the lattice proposed in Section 5.2 is smaller than 16370, 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 $(4, 2, 4, 3)$.

We next consider the case in which there is a preferred lattice structure $(\chi, \leq)$, but there is no system extension $\tilde{\Sigma}$ such that the pair $(\tilde{\Sigma}, (\chi, \leq))$ is interval compatible. We thus look for an over-approximation of the system Σ that might be interval compatible with the desired lattice $(\chi, \leq)$. Such an over-approximation is called a weakly equivalent generalization and is defined the following way.

Definition 6.2. Consider the deterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$. We define $\Sigma_{\geq} = \mathcal{S}(\mathcal{U}_{\geq}, \mathcal{Z}, f_{\geq}, h)$ to be a Σ-*weakly equivalent generalization* of Σ on $\mathcal{U}_{\geq}$ with $\mathcal{U} \subseteq \mathcal{U}_{\geq}$ if

(i) $\mathcal{E}(\Sigma) \subseteq \mathcal{E}(\Sigma_{\geq})$;

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

Item (i) establishes that $\Sigma_{\geq}$ is a generalization of Σ, denoted $\Sigma \subseteq \Sigma_{\geq}$. Moreover, (ii) establishes that those executions of $\Sigma_{\geq}$ 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 $\Sigma_{\geq}$ 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 $(\tilde{\Sigma}, (\chi, \leq))$ being interval compatible with a weaker assumption that we call *weak interval compatibility* defined as follows.

Definition 6.3. (Weak interval compatibility) Consider the extended system $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ of $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ on $(\chi, \leq)$. The pair $(\tilde{\Sigma}, (\chi, \leq))$ is said to be *weakly interval compatible* if

(i) Each $\tilde{\Sigma}$-transition class, $\mathcal{T}_i(\tilde{\Sigma}) \in \mathcal{T}(\tilde{\Sigma})$, is an interval sublattice of $(\chi, \leq)$:

$$\mathcal{T}_i(\tilde{\Sigma}) = [\bigwedge \mathcal{T}_i(\tilde{\Sigma}), \bigvee \mathcal{T}_i(\tilde{\Sigma})];$$

(ii) $\tilde{f} : ([L, U], z) \longrightarrow [\tilde{f}(L, z), \tilde{f}(U, z)]$ is order preserving for any $[L, U] \subseteq \mathcal{T}_i(\tilde{\Sigma})$, and any $z \in \mathcal{Z}$ and for any $i \in \{1, ..., M\}$;

(iii) $\tilde{f} : ([L, U], z) \longrightarrow [\tilde{f}(L, z), \tilde{f}(U, z)]$ is onto for any $[L, U] \subseteq \mathcal{T}_i(\tilde{\Sigma})$ for any $z \in \mathcal{Z}$ and for any $i \in \{1, ..., M\}$;

We have this 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 6.3. *Assume that the deterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ is weakly observable. If there is a lattice $(\chi, \leq)$, such that the pair $(\tilde{\Sigma}, (\chi, \leq))$ is weakly 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*

$$\begin{aligned} f_1(L(k), y(k), y(k+1)) &= \tilde{f}(L(k) \vee \bigwedge O_y(k), y(k)) \\ f_2(U(k), y(k), y(k+1)) &= \tilde{f}(U(k) \wedge \bigvee O_y(k), y(k)) \end{aligned}$$

solves Problem 1.

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

Proposition 6.4. *If the system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ is observable (or weakly observable) and its Σ-weakly equivalent generalization $\Sigma_{\geq} = \mathcal{S}(\mathcal{U}_{\geq}, \mathcal{Z}, f_{\geq}, h)$ is such that the pair $(\tilde{\Sigma}_{\geq}, (\chi, \leq))$ is weakly interval compatible for a given $(\chi, \leq)$ and $\mathcal{U}_{\geq} \subseteq \chi$, then Theorem 6.3 can be applied to $\Sigma_{\geq}$ with $\alpha(k) = \sigma_{\Sigma}(k)(\alpha)$ and $z(k) = \sigma_{\Sigma}(k)(z)$.*

This way, we construct the estimator using $f_{\geq}$, 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 6.2. 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 $f_{\geq} : \mathcal{U} \times \mathcal{Z} \to \mathcal{P}(\mathcal{U})$ such that if $\alpha(k) = \sigma_{\Sigma}(k)(\alpha)$ and $z(k) = \sigma_{\Sigma}(k)(z)$, then $\alpha(k+1) \in f_{\geq}(\alpha(k), z(k))$. $f_{\geq}$ maps an element to a set of possible values in $\mathcal{U}$, and $\mathcal{U}_{\geq} = \mathcal{U}$. 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.

6.3 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.

Definition 6.4. (Nondeterministic transition systems) A *nondeterministic transition system* (NTS) is the tuple $\Sigma = (S, \mathcal{Y}, F, g)$, where

(i) S is a set of states with $s \in S$;

(ii) $\mathcal{Y}$ is a set of outputs with $y \in \mathcal{Y}$;

(iii) $F : S \to \mathcal{P}(S)$ is the state transition set-valued function;

(iv) $g : S \to \mathcal{Y}$ 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 $s(k+1) \in F(s(k))$ 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 3.5, 3.8, and 3.9, which are related with the weak observability property, can be rewritten the same way for NTSs by replacing "deterministic trasition 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) $S = \mathcal{U} \times \mathcal{Z}$ with $\mathcal{U}$ a finite set and $\mathcal{Z}$ a finite dimensional space;

(ii) $F = (f, h)$, where $f : \mathcal{U} \times \mathcal{Z} \to \mathcal{P}(\mathcal{U})$ and $h : \mathcal{U} \times \mathcal{Z} \to \mathcal{Z}$;

(iii) $g(\alpha, z) := z$, where $\alpha \in \mathcal{U}$, $z \in \mathcal{Z}$, and $\mathcal{Y} = \mathcal{Z}$.

We denote this class of nondeterministic transition systems by $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$, and we associate to the tuple $(\mathcal{U}, \mathcal{Z}, f, h)$ the equations

$$\begin{array}{rcl} \alpha(k+1) & \in & f(\alpha(k), z(k)) \\ z(k+1) & = & h(\alpha(k), z(k)) \\ y(k) & = & z(k), \end{array} \tag{28}$$

if f is a set-valued map. Given a lattice $(\chi, \leq)$ with $\mathcal{U} \subset \chi$, the extension $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ 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 $\mathcal{U}$ is allowed to be not invariant under $\tilde{f}$.

Definition 6.5. Given the nondeterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$, a *N-extension of* Σ *on* χ, with $\mathcal{U} \subseteq \chi$ and $(\chi, \leq)$ a complete lattice, is any system $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$, such that

(i) $\tilde{f} : \chi \times \mathcal{Z} \to \mathcal{P}(\chi)$ and $\tilde{f}|_{\mathcal{U}\times\mathcal{Z}} \cap \mathcal{P}(\mathcal{U}) = f$;

(ii) $\tilde{h} : \chi \times \mathcal{Z} \to \mathcal{Z}$ and $\tilde{h}|_{\mathcal{U}\times\mathcal{Z}} = h$.

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

Definition 6.6. Consider the N-extension $\tilde{\Sigma} = \mathcal{S}(\chi, \mathcal{Z}, \tilde{f}, \tilde{h})$ of the nondeterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ on $(\chi, \leq)$. The pair $(\tilde{\Sigma}, (\chi, \leq))$ is said to be *N-interval compatible* if

(i) Each $\tilde{\Sigma}$-transition class, $\mathcal{T}_i(\tilde{\Sigma}) \in \mathcal{T}(\tilde{\Sigma})$, is an interval sublattice of $(\chi, \leq)$:

$$\mathcal{T}_i(\tilde{\Sigma}) = [\bigwedge \mathcal{T}_i(\tilde{\Sigma}), \bigvee \mathcal{T}_i(\tilde{\Sigma})];$$

(ii) $\tilde{f} : ([L, U], z) \longrightarrow [\bigwedge \tilde{f}(L, z), \bigvee \tilde{f}(U, z)]$ is order preserving for any $[L, U] \subseteq \mathcal{T}_i(\tilde{\Sigma})$, and any $z \in \mathcal{Z}$ and for any $i \in \{1, ..., M\}$;

(iii) $\tilde{f} : ([L, U] \cap \mathcal{U}, z) \longrightarrow [\bigwedge \tilde{f}(L, z), \bigvee \tilde{f}(U, z))] \cap \mathcal{U}$ is onto for any $[L, U] \subseteq \mathcal{T}_i(\tilde{\Sigma})$ for any $z \in \mathcal{Z}$ and for any $i \in \{1, ..., M\}$;

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 $b \in f(a)$. Theorem 4.1 transforms to the following.

Theorem 6.5. *Assume that the nondeterministic transition system* $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ *is weakly observable. If there is a lattice* $(\chi, \leq)$*, such that the pair* $(\tilde{\Sigma}, (\chi, \leq))$ *is N-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*

$$\begin{array}{rcl} f_1(L(k), y(k), y(k+1)) & = & \bigwedge \tilde{f}(L(k) \vee \bigwedge O_y(k), y(k)) \\ f_2(U(k), y(k), y(k+1)) & = & \bigvee \tilde{f}(U(k) \wedge \bigwedge O_y(k), y(k)) \end{array}$$

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

In Theorem 6.5, 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.

Note that an equivalent of Proposition 6.4 holds if Σ is nondeterministic and weakly observable. For completeness we reformulate such proposition.

Proposition 6.6. *If the nondeterministic transition system $\Sigma = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f, h)$ is weakly observable and its Σ-weakly equivalent generalization $\Sigma_{\geq} = \mathcal{S}(\mathcal{U}_{\geq}, \mathcal{Z}, f_{\geq}, h)$ is such that the pair $(\tilde{\Sigma}_{\geq}, (\chi, \leq))$ is N-interval compatible for a given $(\chi, \leq)$, then Theorem 6.5 can be applied to $\Sigma_{\geq}$ with $\alpha(k) = \sigma_{\Sigma}(k)(\alpha)$ and $z(k) = \sigma_{\Sigma}(k)(z)$.*

In the following example we show how to apply this proposition to a nondeterministic version of the RoboFlag system in order to construct an estimator.

6.4 Nondeterministic example

In this section, we propose a non-deterministic version of the RoboFlag system and we show how to construct an estimator. The system is analogous to the one introduced in Section 2 and analyzed in Section 5 except for the way the assignment is updated. In fact, we assume that at each step only one pair of robots among the pairs with conflicting assignments swap the assignment, the pair that switches being randomly chosen. The blue robot dynamics are described by the rules

$$z_i(k+1) = z_i(k) + \delta \quad \text{if} \quad z_i(k) < x_{\alpha_i(k)}, \tag{29}$$

$$z_i(k+1) = z_i(k) - \delta \quad \text{if} \quad z_i(k) > x_{\alpha_i(k)}, \tag{30}$$

$$(\alpha_i(k+1), \alpha_{i+1}(k+1)) = (\alpha_{i+1}(k), \alpha_i(k)) \quad \text{if} \quad switch_{(i,i+1)}(k), \tag{31}$$

for $i \in \{1, ..., N\}$, where $switch_{(i,j)}(k)$ is such that

$$switch_{(i,i+1)}(k) \Rightarrow x_{\alpha_i(k)} \geq x_{\alpha_{i+1}(k)} \tag{32}$$

$$switch_{(i,i+1)}(k) \wedge switch_{(j,j+1)}(k) = false, \; i \neq j \tag{33}$$

$$\begin{aligned}&((x_{\alpha_1(k)} \geq x_{\alpha_2(k)}) \vee .... \vee (x_{\alpha_{N-1}(k)} \geq x_{\alpha_N(k)})) \Rightarrow \\ &\quad (switch_{(1,2)}(k) \vee ... \vee switch_{(N-1,N)}(k) = true.\end{aligned} \tag{34}$$

Rules (31) establish that two close robots will exchange their assignments if *switch* is true. In particular, (32) implies that *switch* can be true only for two robots with conflicting assignments, (34) establishes that one pair of close robots will exchange assignments provided there is at least one pair of robots with conflicting assignments, (33) implies that only one pair of robots will exchange assignments. Therefore (32), (33), (34), along with (31), guarantee that, if there are some close robots with conflicting assignments, there is one and only one pair of robots among them that will switch the assignments. This renders the assignment protocol in commands (31) nondeterministic, as at each step we do not know which pair of robots switches assignments. It is possible to show that the assignment protocol converges to the equilibrium value $(1, ..., N)$. For this we defer the reader to [14]. For the blue robots we assume that initially $z_i \in [z_{min}, z_{max}]$, $z_i < z_{i+1}$, and that $x_i < z_i < x_{i+1}$ for all time. With this assumption, one can check that system Σ is weakly observable. The proof is similar the one given in Proposition 5.1. We define $x = (x_1, ..., x_N)$, $z = (z_1, ..., z_N)$, $\alpha = (\alpha_1, ..., \alpha_N)$.

The rules reported in (31) determine the function $f : \mathcal{U} \times \mathbb{R}^N \to \mathcal{P}(\mathcal{U})$ that updates the discrete variables α, while the rules in (30) and (29) determine the function $h : \mathcal{U} \times \mathbb{R}^N \to \mathbb{R}^N$. Therefore the blue robot system is defined by $\Sigma = \mathcal{S}(\text{perm}(N), \mathbb{R}^N, f, h)$.

For the purpose of constructing the estimator, we consider the order $(\chi, \leq)$ described in Section 5.2. One can verify that there is no extension of Σ on χ that is N-interval compatible with $(\chi, \leq)$. As a consequence, we apply Proposition 6.6. We look for a Σ-weakly equivalent generalization of the NTS Σ that admits an extension $\tilde{\Sigma}_{\geq}$ on χ that is N-interval compatible with $(\chi, \leq)$. We define the system $\Sigma_{\geq} = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f_{\geq}, h)$ by defining $f_{\geq}$ in the following way. Let $v(k) = z(k+1) - z(k)$ denote the velocity, then at step k we have for $\beta \in \mathcal{U}$

$$f_{\geq}(\beta, z) := f(\beta, z) \text{ if } v(k) \neq v(k-1) \tag{35}$$

$$f_{\geq}(\beta, z) := [\bigwedge O_y(k), \bigvee O_y(k)] \cap \mathcal{U} \text{ otherwise.} \tag{36}$$

It is easy to verify (i)-(ii) of Definition 6.2, so that $\Sigma_{\geq} = \mathcal{S}(\mathcal{U}, \mathcal{Z}, f_{\geq}, h)$ is a Σ-weakly equivalent generalization of $\Sigma = \mathcal{S}(\text{perm}(N), \mathbb{R}^N, f, h)$. Property (i) is trivially verified. To verify (ii) it is enough to notice that the switch before the stabilization time k_σ of the sequence $\{\sigma_\Sigma(k)(\alpha)\}_{k\in\mathbb{N}}$ is observable. Let $\sigma_{\Sigma_{\geq}}$ denote an execution of $\Sigma_{\geq}$ and $\{\sigma_{\Sigma_{\geq}}(k)(\alpha)\}_{k\in\mathbb{N}}$ the corresponding α sequence, we have that $f_{\geq}(\sigma_{\Sigma_{\geq}}(k_\sigma - 1)(\alpha), z(k_\sigma - 1)) = f(\sigma_{\Sigma_{\geq}}(k_\sigma - 1)(\alpha), z(k_\sigma - 1)) = (1, ..., N)$. This in turn implies that $\sigma_{\Sigma_{\geq}}(k_\sigma - 1)(\alpha) = \sigma_\Sigma(k_\sigma - 1)(\alpha)$ for some execution σ_Σ of Σ.

To find an extension $\tilde{\Sigma}_{\geq}$ that is N-interval compatible with $(\chi, \leq)$, consider the following extension of $f_{\geq}$ on χ at step k for any $q \in \chi$

$$\tilde{f}_{\geq}(q, z) = w, \ (w_i, w_{i+1}) := (q_{i+1}, q_i), \text{ if } v_i(k) \neq v_i(k-1) \tag{37}$$

$$\tilde{f}_{\geq}(q, z) := [\bigwedge O_y(k), \bigvee O_y(k)] \text{ otherwise.} \tag{38}$$

Expression (37) defines an order isomorphism between $[L, U]$ and $[\tilde{f}_{\geq}(L, z), \tilde{f}_{\geq}(U, z)]$ for any $L, U \in \chi$. From expression (38), we deduce that $\tilde{f}_{\geq}$ is trivially order preserving according to the Definition 3.3. Moreover $\tilde{f}_{\geq} : ([L, U] \cap \mathcal{U}, z) \to [\bigwedge \tilde{f}_{\geq}(L, z), \bigvee \tilde{f}_{\geq}(U, z)] \cap \mathcal{U}$ is clearly onto by construction for any $[L, U] \subseteq O_y(k)$, and $\tilde{f}_{\geq}|_{\mathcal{U}} \cap \mathcal{P}(\mathcal{U})$ coincides with $f_{\geq}$ by construction as well. As a consequence the system $\tilde{\Sigma}_{\geq} = \mathcal{S}(\mathcal{P}(\chi), \mathcal{Z}, \tilde{f}_{\geq}, \tilde{h})$ with $\tilde{h}$ as defined in Section 5.2 is N-interval compatible with $(\chi, \leq)$.

We then apply Proposition 6.6 for constructing the estimator. Such an estimator can be written as a set of rules as already done for the example in Section 5.2. In Figure 9 we report

$$W(k) = \frac{1}{N} \sum_{i=1}^{N} |m_i(k)|,$$

which converges to 1 when the value of α has been locked, and

$$E(k) = \frac{1}{N} \sum_{i=1}^{N} |\alpha_i(k) - i|,$$

which gives an idea of the speed of convergence of the assignment to the equilibrium value $(1, ..., N)$. $|[L(k), U(k)] \cap \mathcal{U}|$ converges to 1, but $|[L(k), U(k)]|$ is not a monotonic function of k as it was in the deterministic case. This is due to the nondeterministic nature of the transition functions, as one element can be mapped to many. The choice of $f_{\geq}$ has a considerable impact on the convergence speed of the estimator. The map $f_{\geq}$ we chose is rough and does not take other information that we have on the system into account. For example it does not model the fact that even if there is an unobservable switch, a subset of the robots, depending on their assignment estimates, undergoes particular switches. The most information we can model with $f_{\geq}$ the fastest is the convergence rate.

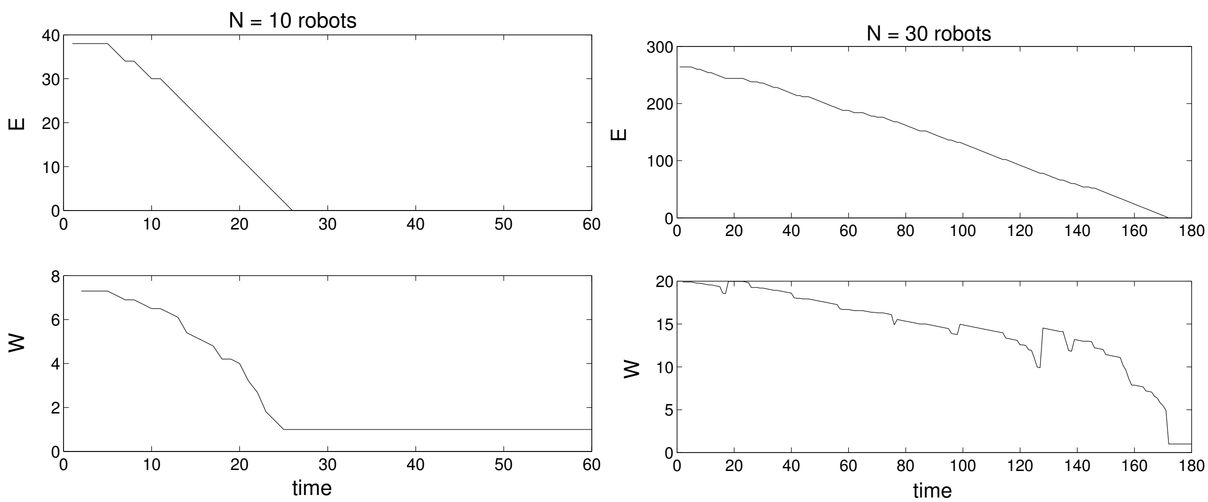

Figure 9: Example with N=10(left) and N=30(right): upper plot shows the stabilization function of the α assignment (E), while lower plot shows the function W for the estimator.

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 three such as is found in Caines [6], Balluchi et al. [3], and Özveren et al. [11], or in similar methods such as in Del Vecchio and Klavins [20]. 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.

This drastic complexity reduction can be allowed if the system under consideration can be extended to a lattice in such a way that the extension is interval compatible with the lattice itself. The proposed methodology has revealed to be a powerful tool to construct scalable estimators. This was confirmed by the simulation results obtained for a multi-robot system composed by 30 robots and with a number of discrete states equal to 30!. Some extensions to the basic results have been provided as well.

Many aspects need to be still improved upon. The complexity considerations carried out in Section 6 are not formal enough. They just give a rough idea of what causes of complexity might be when designing a discrete state estimator. More work needs to be done in this direction in order to formally identify the types of lattices that allow efficient computation and representation of joins and meets. 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. The possibility of constructing a joint continuous-discrete variable lattice will be explored. Another future research direction will consider the possibility of finding an automated way to look for a system extension that is interval compatible with a given lattice, if it exists.

8 Acknowledgments

This work was supported in part by the NSF Center for Neuromorphic Systems Engineering at Caltech, and partially by AFOSR under grants F49620-01-1-0460 and FA9550-04-1-0169.

References

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

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

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

[4] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, and A. Sangiovanni-Vincentelli. Observability of hybrid systems. In *Conf. on Decision and Control*, pages 1159–1164, 2003.

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

[6] P. E. Caines. Classical and logic-based dynamic observers for finite automata. *IMA J. of Mathematical Control and Information*, pages 45–80, 1991.

[7] A. R. Cassandra, L. P. Kaelbling, and M. L Littman. Acting optimally in partially observable stochastic domains. In *Proc. 12th Conference on Artificial Intelligence*, pages 1023–1028, Seattle, WA, 1994.

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

[9] R. D'Andrea, R. M. Murray, J. A. Adams, A. T. Hayes, M. Campbell, and A. Chaudry. The RoboFlag Game. In *American Control Conference*, pages 661–666, 2003.

[10] B. A. Davey and H. A. Priesteley. *Introduction to Lattices and Order*. Cambridge University Press, 2002.

[11] C. M. ¨Ozveren and A. S. Willsky. Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control*, 35(7):797–806, 1990.

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

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

[14] E. Klavins and R. M. Murray. Distributed algorithms for cooperative control. *Pervasive Computing*, 3:56–65, 2004.

[15] David G. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, AC-16:6:596–602, 1971.

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

[17] P. J. Ramadge. Observability of discrete event systems. In *Conf. on Decision and Control*, pages 1108–1112, Athens, Greece, 1986.

[18] E. De Santis, M. D. Di Benedetto, and G. Pola. On observability and detectability of continuous-time linear switching systems. In *Conf. on Decision and Control*, pages 5777–5782, 2003.

[19] E. D. Sontag. *Mathematical Control Theory*. Springer, 1998.

[20] D. Del Vecchio and E. Klavins. Observation of guarded command programs. In *Conf. on Decision and Control*, pages 3353–3359, Hawaii, 2003.

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

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